MICROPROCESSOR-CONTROLLED INVERTER-FED SYNCHRONOUS MOTOR

by

Nicholas Martin Mallinson, B.Sc.(Eng), A.C.G.I.

Volume One of Two Volumes

October 1983

A thesis submitted for the degree of
Doctor of Philosophy of the University of London
and for the Diploma of Membership of the
Imperial College of Science and Technology

Department of Electrical Engineering
Imperial College of Science and Technology
London SW7 2AZ

ABSTRACT

The thesis is concerned with a class of variable-speed motor drive system, in which a synchronous motor is operated in the autopiloted mode from a voltage-source inverter. A system comprising a permanent magnet square-wave synchronous motor, a transistor inverter, a microprocessor controller and an optical fibre shaft-position detector was designed, built and tested.

A simple sinusoidal analysis to predict the performance of a synchronous motor operated in voltage-forced mode is given. The experimental test equipment used to confirm the general nature of the predictions is described and the results presented. The logic circuits that the microprocessor later replaced are explained.

The motor design is unusual. The stator has seven phases and generates a quasi-square back-emf to match the inverter voltage waveform. A designed rotor speed of 30000 rpm results in a high power-to-weight ratio. The square-wave rotor flux is generated by polymer bonded samarium cobalt magnets encased in a carbon fibre cylinder.

The inverter uses MOSFET transistors and the seven phases are independently controllable. The phase rating is ±60 volts at 7 amperes giving a total continuous rating of 3kW at frequencies from d.c. to at least 1.5kHz.

The controller uses the 16 bit Texas 9900 microprocessor to keep the motor synchronised. Software was developed to implement two synchronising strategies using either interrupts or a counter sampled by the microprocessor. Control commands are entered via a purpose built keypad. A digital tachometer was developed to provide speed data to the controller.

The system was tested under steady state conditions and the performance characteristics and waveforms are illustrated by experimental curves and photographs, and are compared with predictions.

It is concluded that most of the system's design features have potential for use on autopiloted synchronous motor drives of various rated powers, and in particular the performance of the square-wave motor is sufficiently promising to warrant further investigation. It is also concluded that a microprocessor can successfully replace hardwired logic in inverter controllers.

ACKNOWLEDGEMENTS

The author would like to express his gratitude to his supervisor, Dr. H.R. Bolton, for providing valuable guidance whilst not interfering excessively during the course of the work described in this thesis. Without doubt, the author would not have contemplated working for a Ph.D had Dr. Bolton not provided the opportunity to do so. The author was given great freedom to direct the work and Dr. Bolton deserves sincere thanks for this.

Thanks for financial support are due to the S.E.R.C. (formerly the S.R.C.) and to Dr. Bolton.

Several friends have been particularly helpful. Those who deserve special mention are: Malcolm Shute; Nigel McQuin; John Tasker; Dr. Clive Bryant; Dr. Somchai Chatratana; Dr. Robert Palmer. The author feels that without the assistance given by the above mentioned people, the project may not have reached such a successful conclusion.

The author would also like to acknowledge the manufacturing and technical assistance given to him by Mr. R.B. Owen, Mr. R. Moore, Dr. B. Williams, Dr. S. Williamson and Dr. D.A.G. Peddar during the project.

Finally, it must be stated that none of the work would have been possible without the continuous support, financial and otherwise, provided by the author's wife, Jennie. The low standard of rented accomodation in London is but one of the many sacrifices that she has endured during the course of the work. This thesis is a splendid demonstration of her newly acquired skill of typing.

Thank you Jennie.

LIST OF CONTENTS

Volume One

Volume Two

CHAPTER 1

INTRODUCTION

An extensive variety of electric motor drives are used in industrial, commercial, and domestic applications throughout the world. A simple way of classifying the many types of drives is to separate them into either a "controlled drives" category or a "fixed speed drives" category.

The "controlled drives" category covers many forms of drive. Drives within this category can regulate one or more variables such as speed, torque, or output power. The most common controlled drive is based on the d.c. motor with variable armature volts. Unfortunately, the fact that d.c. motors operate by virtue of the electro-mechanical commutator is a major source of unreliability. In practice d.c. motors require regular maintenance to ensure trouble free commutation. In addition, the sparking associated with the action of a commutator makes a d.c. motor drive unsuitable for use in certain industrial locations, since the sparks can cause radio interference or ignite flammable gases.

The requirement from industry for reliable, low-maintenance, controlled drives has therefore led to considerable work in recent years on various "brushless" controlled drive systems. These controlled drives can be classified into two main types. The first type achieves the desired controlled performance by virtue of the machine design. Drives of this type include the spherical motor, pole-changing induction and synchronous motors, and induction motors fed from a fixed frequency variable voltage supply.

The second type of controlled drive achieves the desired performance, by a combination of a machine and (nowadays) power electronics. Typical examples of such systems include: slip energy recovery on induction motors; fixed frequency, variable voltage thyristor/triac supplies for induction motors; open loop connection of cycloconverters to induction or synchronous motors; open loop connection of d.c. link inverters to induction or synchronous motors; switched reluctance/stepper motors fed from unipolar switching inverters; brushless d.c. autopiloted synchronous motors (that is, the closed loop combination of an inverter with a synchronous motor). All of the example systems listed above are employed to varying degrees in present day industrial and consumer drive products.

The fixed-speed drives category is dominated by the standard mains-fed induction motor but d.c., hysteresis, reluctance, and synchronous motors are used (and a.c. commutator motors to some extent). A brushless inverter-fed system can be used to advantage, when the fixed speed required is greater than that which an induction motor can provide when supplied from the mains. For example, the textile industry and special applications including aerospace projects or centrifuges require high, fixed operating speeds.

This thesis is concerned with work on controlled drives of the brushless d.c. autopiloted synchronous motor type. In essence, a brushless d.c. drive comprises four units: a motor; a power supply; a power converter (such as a voltage source inverter); and a rotor position sensor. The motor is of a synchronous type with the "armature" winding carried on the stator. A permanent magnet or wound field coil is mounted on the rotor to provide the working flux. The stator windings are energised by the power converter, which is itself controlled by the rotor position sensor. The motor torque arises from the misalignment of the stator and rotor magnetic axes, and the rotor position sensor ensures

that the torque angle remains essentially constant as the rotor turns. Hence continuous unidirectional torque is generated by the synchronous motor, no matter what speed the system is running at.

Typical papers which have dealt with "small" (low power) brushless d.c. drives are those by Williamson et al (1.1), Russell et al (1.2), Persson (1.3, 1.4), and Ashen et al (1.5). The same principles are employed with larger power brushless d.c. drives, and typical papers for such drives are those by Feltbower et al (1.6) and Schauder et al (1.7). There are numerous publications in the technical press, covering work performed in Canada, the U.S.A., France, Germany, and Japan, on converter fed synchronous motor drives with position sensor control. Excellent sources for information on recent work are conference proceedings such as: 1981 IEEE Industry Applications (1.8); 1979 IEE Electrical Variable-Speed Drives (1.9); 1980 Athens Conference on Electrical Machines (1.10); and the 1980 Conumel Conference (1.11). The main topics covered by the papers presented in these proceedings are typically:

(a) the circuit analysis of motor/converter systems, with many papers concentrating on thyristor converters;

(b) the analysis of the operating modes of brushless d.c. systems using phasor diagrams, loci, and operating charts, to predict the performance for variable frequency operation with, for example, fixed load angle;

(c) the determination of the effect of motor design on performance (for example, rotor dampers, quadrature axis field excitation, and induced field excitation), and the operating limits of the motor taking into account possibilities such as flux boosting;

(d) accounts of drives developed for special purposes (for example, high torque, low speed servo drives);

(e) unusual converter designs with descriptions of commutation techniques, natural commutation limits, use of transistors, and details of inverter control strategies (for example, pulse width modulation);

(f) the use of microprocessors in drive systems.

The object of the project described in this thesis was to design, build, test and assess an inverter fed synchronous motor system, capable of high speed operation with a power output of between one to five kilowatts. A synchronous motor was chosen in preference to an induction motor for several reasons including:

(a) smaller stator I^2R loss due to absence of magnetising current;

(b) no magnetising VA requirement from the feed system;

(c) in the case of motors with permanent magnet excitation, no rotor excitation losses/heating;

(d) unity or leading power factor operation possible, allowing reduced feed VA per output power;

(e) natural commutation conditions can occur over parts of the operating range;

(f) if desired, 90° torque angle operation can be maintained with autopiloted synchronous motor drives. This enables a larger torque per stator current and total flux to be obtained than with induction motor drives, where the torque angle

for maximum torque per stator-amp^2 is approximately 45^o.

In small machines the per-unit magnetising current levels can be appreciable, and so items a, b, and c listed above are of considerable significance. Against the advantages of synchronous motors must be set the synchronous motor's lower level of robustness, greater cost, and, for permanent magnet synchronous motors, reduced operating flexibility caused by the lack of field control capability. However, these disadvantages were not of sufficient consequence to prevent the use of a synchronous motor in the project work.

A high maximum operating speed of 30000 rpm was envisaged for the project synchronous motor in an attempt to achieve a good power to weight ratio for the system. The large number of components making up a typical brushless d.c. motor system tend to make it relatively expensive, and so an improved power to weight ratio is an important benefit which can in some applications be sufficient to justify the system cost. A motor speed of 30000 rpm is not directly usable in many applications, but advances in gearbox lifetimes and efficiencies in recent years have made it practicable and worthwhile to gear down the motor speed to the required magnitude. The most common example of the use of high speed machines to achieve good power to weight ratios is in the aircraft industry, where 400Hz generators and motors are used (2 pole 400Hz machines rotating at 24000 rpm). As the cost, reliability, and noise of high speed drives improve, it is likely that high speed versions of load equipment (such as pumps, fans, and drills) will emerge into the market. The choice of a high motor speed does increase several problems that are not generally of great concern at standard motor speeds. Any machine rotating at high speed is a potential noise source, and the noise produced at a motor speed of 30000 rpm could be unacceptable in some applications. Nevertheless, noise can

be tolerated if the power to weight or speed objectives are paramount, whereas the mechanical problems arising at speeds of the order of 30000 rpm have to be completely overcome to guarantee reliable and safe motor operation. Therefore, the rotor construction and the type of bearings used are of great significance. Bearings capable of the desired speed are becoming readily available; many have adequately long working lifetimes. The rotor is subjected to very high centrifugal forces, but a "strength-ring" feature can be incorporated into the rotor design, and centrifugal forces can be minimised by choosing a small rotor diameter.

A permanent magnet rotor design was chosen for the synchronous motor in preference to a wound field design. A wound field rotor does allow the field excitation to be adjusted, and so enables changes in the motor's operating characteristic to be made, but there are significant problems in transmitting power onto a rotor spinning at 30000 rpm. These problems are avoided by the choice of a permanent magnet rotor. In addition, the location of the permanent magnets on the rotor pole faces presents a large effective airgap to the stator winding, with beneficial reductions in armature reaction and phase current risetimes.

The design of the drive system is unusual owing to a desire to assess a number of "new technology" components and sub-systems, and to investigate an unconventional motor construction. The features incorporated into the system include briefly:

(a) a quasi-square wave flux distribution around the rotor periphery which reacts with a similar quasi-square distribution of current around the stator periphery. These unusual distributions raise the shaft output torque per motor size and dissipation. More important though, from a system design viewpoint, is the fact that quasi-square back-emf phase voltages are obtained. This permits good matching to a simple quasi-square voltage source inverter,

without the need for pulse width modulation techniques such as are necessary with sinusoidal systems. The quasi-square distributions were implemented via a stator winding of larger than normal phase number (7), and a rotor construction employing radially orientated permanent magnets near to the airgap. The high harmonic content of the quasi-square waveforms is a potential source of large losses, but the method was judged to be worth investigation so that the good inverter-to-motor match could be evaluated;

(b) the use of power MOSFET transistors as the switching elements in the inverter. Little published information was available on the use of these devices and so the successful implementation of a MOSFET inverter is felt to be of significance at this time;

(c) the use of a microprocessor (Texas Instruments TMS9900) to perform the real time autopiloting tasks of the drive system, and in addition permit changes in machine load angle to be selected;

(d) a keypad control unit which permits full control over the motor to be exercised via a simple set of keys;

(e) a carbon fibre strength ring to retain the permanent magnets on the rotor pole faces at the top design speed of 30000 rpm;

(f) the use of optical fibres in the optical rotor-position sensor unit;

(g) an accurate and fast response digital tachometer, providing a 16 bit speed reading, with a range from 16 rpm to 65535 rpm in 1 rpm steps.

These features and the reasons for their choice are described in detail in later chapters of this thesis.

It should be stated at this point that the work by McLean et al (1.12), on the performance and design of induction motors with square-wave excitation, and by Enslin et al (1.13), on the flux and current distributions in a 9 phase inverter driven induction motor, was important in the decision to tackle a quasi-square wave synchronous motor. The good system match that is possible with "square wave" systems was stressed by McLean et al:

> "Designing the induction motor to match the simpler square-wave-output inverter is thought to be preferred on grounds of cost and reliability to designing a more complex inverter with an output to match the requirements of a standard induction machine."

One of the main conclusions of McLean et al is that a 3 phase square-wave system using a conventional motor produces a low efficiency drive, but that increasing the number of phases and using fully pitched coils improves the electrical efficiency. A 9 phase square-wave induction motor was found to be virtually as efficient as a 3 phase motor fed from a sinusoidal supply.

It is interesting to note that the relative complexity of a "square-wave" synchronous motor, when compared to a "square-wave" induction motor, is compensated by the fact that the square-wave distributions are more easily maintainable in a pole-face mounted permanent-magnet-rotor synchronous machine than in an induction motor.

The work reported in this thesis is believed to be novel and relevant to the future special-drive requirements of industry. The project is thought to be one of the first to achieve real time control of a high speed autopiloted drive system with an interrupt rate in excess of 20000 per second. Critics may argue that the use of a microprocessor to perform the simple logic operations associated with autopiloting cannot be justified, but in general such an approach is attractive because:

(a) the power of the software can enable significant reductions in peripheral logic to be made;

(b) the microprocessor can be easily reprogrammed, thus enabling changes to the system operation to be readily implemented.

The idea behind the microprocessor part of the project was to investigate these potential benefits. A major concern at the start of the project was the speed limitation that the microprocessor might impose on the autopiloting operation, but suitable system design and careful choice of software instructions can minimise this problem as explained later in this thesis.

The main "cost" of a microprocessor based system is usually the development man-hours needed, and this is certainly true for the project described in this thesis. However, the hardware and software evolved during the project is flexible, and has potential for use (with little modification) in a great many autopiloted synchronous machine systems, no matter what switching sequence, number of phases, power rating, outer control loop configuration, etcetera, may be involved.

To gain experience with an autopiloted synchronous motor drive prior to building the rather advanced "square wave" system, some work was performed on a small 3 phase motor system (based on a magslip) and much of the microprocessor software and interfacing was tested first on the 3 phase system. This approach minimised the number of new pieces of technology that had to be "commissioned" at any one time.

The work described in this thesis is arranged on a topic-per-chapter basis wherever possible.

Chapter 2 deals with the sinusoidal analysis of a

voltage forced synchronous motor operated on a variable frequency supply, and demonstrates the effect that the load angle setting has on the torque/speed characteristic of such a motor when it is autopiloted.

Chapter 3 covers the basic operating principle of autopiloted synchronous motor systems. It goes on to discuss various position detector methods, and the ways in which an adjustable load angle can be implemented. The 3 phase motor system is described, and the test results obtained from it are presented and compared with the sinusoidal predictions obtained using the equations developed in Chapter 2. Chapter 3 concludes with a description of the electronic load angle adjustment circuits that were implemented on the 3 phase motor system.

Chapter 4 deals with the microprocessor system that was developed to control either the 3 phase motor or the 7 phase square-wave motor. Comprehensive details of the software are given and performance problems are discussed.

Chapters 5 and 6 respectively describe the design and construction of the 7 phase MOSFET inverter and the 7 phase "square-wave" synchronous motor. Chapter 6 also describes the optical-fibre-based rotor position detector unit which was designed and constructed for the 7 phase motor.

Chapter 7 discusses the problems of communicating with a microprocessor when it is executing a real time program. The development of the keypad unit, which enables control instructions to be communicated to the microprocessor during motor operation, is covered in detail.

Chapter 8 briefly explains several standard methods of obtaining digital speed readings, and the design and construction of a high-accuracy digital tachometer is then described.

Finally, Chapters 9 and 10 respectively discuss the test results obtained from the "square-wave" motor, and the conclusions that can be drawn from the work. Some ideas for future work are suggested in Chapter 10.

CHAPTER 2

SINEWAVE ANALYSIS OF A VARIABLE FREQUENCY VOLTAGE FORCED AUTOPILOTED SYNCHRONOUS MOTOR

2.1 Introduction

In autopiloted synchronous motor systems, a synchronous motor, fed from a variable frequency semiconductor inverter, is kept in synchronism at whatever speed it rotates at by a shaft position sensor controlling the inverter. One method of controlling speed is achieved by varying the d.c. supply voltage to the inverter. Therefore in such a system the motor supply voltage and frequency are actually variable and the load angle is fixed by the position sensor; whereas in a synchronous motor system fed from a conventional busbar the exact opposite is true, since the voltage and frequency are fixed and the load angle varies to suit the load conditions. The difference in operating mode requires an extension to "standard" synchronous machine analysis in order that the torque/speed characteristics can be established.

2.2 Steady State Performance Characteristics of Voltage Forced Autopiloted Synchronous Motors

2.2.1 Survey of Simple Analysis Methods

The analysis of multiphase synchronous motors operating on fixed frequency busbar supplies is dealt with in many texts, including those by Say (2.1) and Matsch (2.2).

The methods employed depend to some extent on the conventions adopted by the authors. The generalised theory of electrical machines (2.3, 2.4), in which all quantities are resolved along two axes, is a well known and powerful method by which synchronous motors can be analysed. However, little analysis was done on the variable frequency operation of synchronous motors prior to the development of semiconductor inverters capable of producing variable frequency voltage supplies. In addition, analyses for standard fixed frequency busbar operation often ignore stator resistance (2.1), as it is negligible in comparison with the stator reactances for the large machines conveniently considered.

The increasing use of brushless d.c. motor drives, and more particularly the use of both large and small synchronous motors in such systems, has led to new analyses being published. More importantly, the low power capability of the early semiconductor inverters meant that only relatively small synchronous motors could be driven, and since small motors do not have negligible stator resistance, recent analyses have tended to include it. Work has been published on the subject of sinusoidal synchronous motors fed from variable frequency sinusoidal voltage supplies by several authors including Ashen (2.5), Woodbury (2.6), Seshanna (2.7), Slemon (2.8), Chalmers (2.9) and Shepherd (2.10). The case of a sinusoidal motor fed from a quasi-square voltage waveform has been considered by Woodbury (2.11). These analyses employ steady state methods and are often highly idealised. For example, Woodbury (2.6), Slemon (2.8), Chalmers (2.9) and Shepherd (2.10) assume that the voltage and current waveforms are sinusoidal even though the systems they consider use quasi-square wave voltage source inverters. In addition, Woodbury (2.11) assumes that motor winding inductance is negligible since the analysis is conducted for "low motor speeds".

In order to take account of the actual inverter vol-

tage or current waveforms and the full winding parameters, it is necessary to extend the analyses. The state of the voltages and currents can be analysed by a numerical time-step method in which the instantaneous voltages and currents are calculated at suitable time intervals. The method allows irregular supply and back-emf waveforms to be readily incorporated. Alternatively, a Fourier series method can be used to determine the stator currents resulting from non sinusoidal applied stator voltages. Work has been published on this method by Gumaste (2.12).

Neither of the synchronous motors discussed in this thesis are really suitable for analysis by simple sinusoidal methods. The "Magslip motor" generates a sinusoidal back-emf, and is driven by a quasi-square voltage waveform, but it has a significant phase inductance and so can not be analysed by the method used by Woodbury (2.11), since in the upper part of the speed range reactance effects are important. The seven phase square wave motor possesses a very low stator inductance but the square wave back-emf is a major obstacle to any analysis based on sinusoidal assumptions. However, sinusoidal analysis of the Magslip system gives useful insight into the typical torque/speed curves that can be expected from an autopiloted synchronous motor. Whilst the predictions are not necessarily quantitatively accurate, they do give a qualitative idea of the optimum operating conditions. The equations and predictions of this chapter form the basis of the practical Magslip work described in Chapter 3.

2.2.2 Relationships Between Presented Analysis and Previously Published Analyses

Much of the work in this chapter on autopiloted voltage forced synchronous motors has been dealt with to some extent in the papers referenced in section 2.2.1 (2.5 to 2.12 inclusive). For example, the torque/speed trends of an autopiloted synchronous motor with its load angle set

at values of 0° and 90° are discussed both by Ashen (2.5) and Chalmers (2.9). However, some aspects of the work are not covered explicitly in any published work that the author is aware of. For example, the torque equation 2.15 for a salient pole machine derived in section 2.3.1 is referred to by Ashen (2.5) but he does not quote it. The derivation of the cylindrical rotor torque equation 2.22 from the salient pole torque equation 2.15 has not been found by the author in any known texts. Equation 2.22 is very useful as it includes the effect of stator winding resistance. Finally, the derivation of the optimum load angle to maximise torque per volt in both a cylindrical and salient pole machine (sections 2.5 and 2.7 respectively) is believed to be original, although Say (2.13) does consider the rate of change of torque with respect to load angle, in order to derive an expression for the synchronising torque in conventionally operated machines. Slemon (2.8) derives a formula for the required load angle for maximum torque in a cylindrical rotor motor, but it is more complicated than the one derived in section 2.5 of this chapter. Chalmers (2.9) presents curves of torque versus torque angle for a salient pole machine, as well as a tabulated summary of the range over which the load angle can be varied to achieve various operating conditions. However a predicted torque/speed curve of the form shown in section 2.7 is not given. Therefore, it is felt that there are a sufficient number of interesting points contained in the following sections of this chapter to justify the inclusion of basically standard analysis.

2.3 The Synchronous Motor Torque Equation

2.3.1 The Derivation of the Torque Equation for a N Phase Salient Pole Synchronous Motor Fed From a Voltage Source Inverter

If leakage inductance, magnetic circuit saturation, and losses such as eddy currents and hysteresis are neglected, the phase winding of a synchronous motor can be modelled as an inductor L_{ph} in series with a resistance R_{ph} and a "back-emf" voltage source E_{ph}, as shown in fig. 2.1.

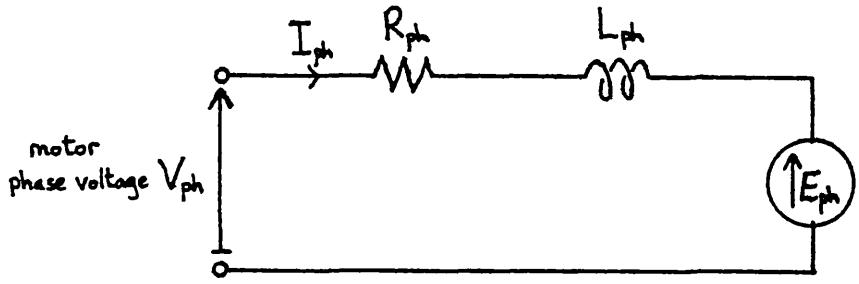


Fig. 2.1 Simple Model for One Phase of a Synchronous Motor

If the terminals of the phase winding are connected to a sinusoidal voltage supply, whose phase is fixed relative to E_{ph} by a position sensor, the input power P_{in} to the phase winding is given by

$$P_{in} = V_{ph} \cdot I_{ph} \cdot \cos \phi \quad \text{watts} \qquad 2.1$$

where V_{ph} is the rms phase voltage (volts); I_{ph} is the rms phase current (amps); and $\cos \phi$ is the power factor.

The mechanical output power per phase, P_{out}, is given by

$$P_{out} = T_{ph} \cdot \omega \quad \text{watts} \qquad 2.2$$

where T_{ph} is the output torque per phase (Nm) and ω is the

angular velocity of the motor shaft (rad/s).

The resistive power loss per phase, P_{loss}, is given by

$$P_{loss} = I_{ph}^2 . R_{ph} \quad \text{watts} \qquad 2.3$$

Therefore, if the losses due to eddy currents, hysteresis, and skin effect are neglected, the gross power per phase, P_{out}, available at the motor shaft is given by

$$P_{out} = P_{in} - P_{loss} \qquad 2.4$$

or substituting for the various terms

$$T_{ph} . \omega = V_{ph} . I_{ph} . \cos \phi - I_{ph}^2 . R_{ph}$$

dividing through by ω gives the torque equation for a single phase of the motor:

$$T_{ph} = \frac{1}{\omega} \left[V_{ph} . I_{ph} . \cos \phi - I_{ph}^2 . R_{ph} \right] \qquad 2.5$$

The current, I_{ph}, and power factor, $\cos \phi$, can be eliminated from equation 2.5 as follows. The phasor diagram for the motor phase winding is shown in fig. 2.2. The angle δ between V_{ph} and E_{ph} is the motor load angle, and the angle ϕ between V_{ph} and I_{ph} is the power factor angle. The phase current, I_{ph}, is shown resolved into its two component parts, the direct axis current I_d and the quadrature axis current I_q, which are related by the equation

$$I_{ph}^2 = I_d^2 + I_q^2 \qquad 2.6$$

The phase current I_{ph} may be resolved along the axis of the V_{ph} phasor in terms of I_d and I_q:

$$I . \cos \phi = I_q . \cos \delta - I_d . \sin \delta \qquad 2.7$$

Substitution of 2.7 into 2.5 yields

$$T_{ph} = \frac{1}{\omega}\left[V\left(I_q.\cos\delta - I_d.\sin\delta\right) - \left(I_d^2 + I_q^2\right)R_{ph}\right] \qquad 2.8$$

Having eliminated I_{ph} and cos ϕ, the analysis can be continued to remove I_d and I_q from the equation. Resolving the voltage phasor V_{ph} along the direct and quadrature axes gives, respectively:

$$V.\sin\delta = \omega.L_q.I_q - R_{ph}.I_d \qquad 2.9$$

where L_q is the phase winding quadrature axis inductance; and

$$V.\cos\delta = E_{ph} + R_{ph}.I_q + \omega.L_d.I_d \qquad 2.10$$

where L_d is the phase winding direct axis inductance.

The phase back-emf, E_{ph}, is given by

$$E_{ph} = K_b.\omega \qquad 2.11$$

where K_b is the back-emf constant. Hence equation 2.10 becomes

$$V.\cos\delta = K_b.\omega + R_{ph}.I_q + \omega.L_d.I_d \qquad 2.12$$

Solving equations 2.9 and 2.12 simultaneously leads to the expressions for I_d and I_q given below.

$$I_d = \frac{V_{ph}.\omega.L_q.\cos\delta - R_{ph}.V_{ph}.\sin\delta - K_b.\omega^2.L_q}{\left(R_{ph}^2 + \omega^2.L_d.L_q\right)} \qquad 2.13$$

and

$$I_q = \frac{V_{ph}\left(\omega.L_d.\sin\delta + R_{ph}.\cos\delta\right) - K_b.\omega.R_{ph}}{\left(R_{ph}^2 + \omega^2.L_d.L_q\right)} \qquad 2.14$$

Substitution of 2.13 and 2.14 into 2.8 then gives the final torque equation for a salient pole synchronous machine:

$$T_{ph} = \frac{1}{\omega}\left\{\frac{V_{ph}.\cos\delta\left(V_{ph}\left[\omega.L_d.\sin\delta + R_{ph}.\cos\delta\right] - R_{ph}.K_b.\omega\right)}{\left(R_{ph}^2 + \omega^2.L_d.L_q\right)}\right.$$

$$- \frac{V_{ph}.\sin\delta\left(V_{ph}\left[\omega.L_q.\cos\delta - R_{ph}.\sin\delta\right] - K_b.\omega^2.L_q\right)}{\left(R_{ph}^2 + \omega^2.L_d.L_q\right)}$$

$$- \left[\frac{V_{ph}\left(\omega.L_q.\cos\delta - R_{ph}.\sin\delta\right) - K_b.\omega^2.L_q}{\left(R_{ph}^2 + \omega^2.L_d.L_q\right)}\right]^2.R_{ph}$$

$$\left. - \left[\frac{V_{ph}\left(\omega.L_d.\sin\delta + R_{ph}.\cos\delta\right) - R_{ph}.K_b.\omega}{\left(R_{ph}^2 + \omega^2.L_d.L_q\right)}\right]^2.R_{ph}\right\} \qquad 2.15$$

The total torque T for an n phase machine is then

$$T = n.T_{ph} \quad \text{Nm} \qquad 2.16$$

The standard salient pole torque equation can be directly obtained from equation 2.15, since at reasonable operating speeds, ω is much greater than zero and hence the winding resistance is small compared to the winding inductance in all but the very smallest of motors. For $R_{ph} \rightarrow 0$, equation 2.15 becomes:

$$T_{ph} = \frac{1}{\omega^2}\left\{\left(\frac{V_{ph}.E_{ph}}{L_d}\right)\sin\delta + \frac{1}{2}\left[V_{ph}^2\left(\frac{1}{L_q} - \frac{1}{L_d}\right)\sin 2\delta\right]\right\} \qquad 2.17$$

which is the well known salient pole torque equation.

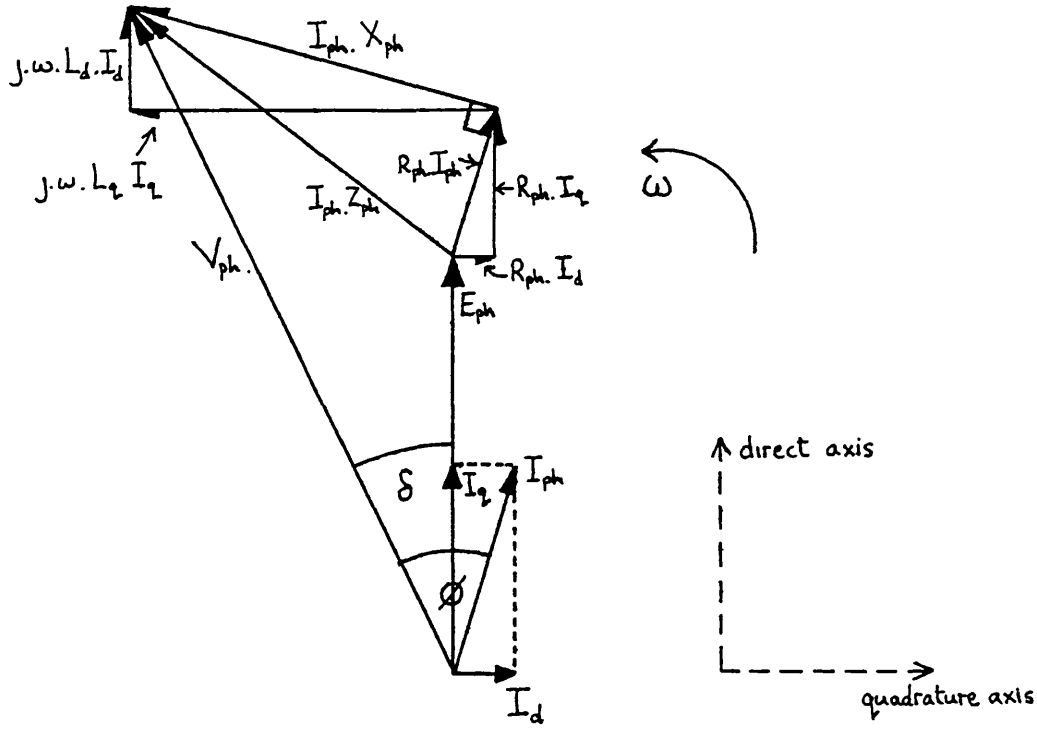


Fig. 2.2 Phasor Diagram for a Synchronous Motor

2.3.2 Simplification of the Salient Pole Torque Equation to the Cylindrical Rotor Form

In a cylindrical rotor machine, the direct and quadrature axis inductances are equal.

i.e. $$L_d = L_q = L_{ph} \qquad 2.18$$

Substitution of equation 2.18 into the torque expression 2.15 gives the torque equation for a cylindrical rotor machine. The equation is long and complicated but can be greatly reduced by substitution of equation 2.11 and the following:

$$X_{ph} = \omega . L_{ph} \qquad 2.19$$

and $$Z_{ph}^2 = R_{ph}^2 + \omega^2 . L_{ph}^2 \qquad 2.20$$

where X_{ph} is the phase winding reactance and Z_{ph} is the phase winding impedance. After simplifying the algebra,

the cylindrical rotor torque equation becomes

$$T_{ph} = \frac{E_{ph}}{\omega . Z_{ph}^2}\left[V_{ph}.X_{ph}.\sin\delta + V_{ph}.R_{ph}.\cos\delta - E_{ph}.R_{ph}\right] \quad 2.21$$

If equation 2.11 is used to replace the E_{ph} term outside the brackets of equation 2.21, and 2.21 is substituted into 2.16, the total torque for a cylindrical rotor synchronous motor is given by

$$T = \frac{n.K_b}{Z_{ph}^2}\left[V_{ph}.X_{ph}.\sin\delta + V_{ph}.R_{ph}.\cos\delta - E_{ph}.R_{ph}\right] \quad 2.22$$

For cylindrical rotor motors in which R_{ph} is negligible at reasonable values of ω, equation 2.22 reduces to:

$$T = \frac{n.K_b}{X_{ph}^2}\left[V_{ph}.X_{ph}\right] \sin\delta$$

$$\text{i.e. } T = \frac{1}{\omega}\left[\frac{n.E_{ph}.V_{ph}}{X_{ph}}\right] \sin\delta \quad 2.23$$

which is the standard cylindrical rotor synchronous motor torque equation quoted in many text books.

Equation 2.22 can be derived directly from the phasor diagram of fig. 2.2 by manipulating equation 2.5 to eliminate the current and power factor. The derivation given in this section shows that a cylindrical rotor machine is merely a special case of a salient pole rotor machine in which the direct and quadrature axes' inductances are equal. Equation 2.22 is an expression linking the motors' phase winding constants to the applied terminal volts V_{ph}, the back-emf voltage E_{ph}, and the load angle between the two voltages, δ.

For certain conditions of load angle and operating frequency the equation simplifies further. In particular, for $\delta = 0^\circ$

$$T = \frac{n.K_b.R_{ph}}{Z_{ph}^2}\left[V_{ph} - E_{ph}\right] \qquad 2.24$$

At low speeds, as $\omega \longrightarrow 0$, $Z_{ph} \longrightarrow R_{ph}$ and so equation 2.22 becomes

$$T = \frac{n.K_b}{R_{ph}}\left[V_{ph} - E_{ph}\right] \qquad 2.25$$

or

$$T = \frac{n.K_b}{R_{ph}}\left[V_{ph} - K_b.\omega\right] \qquad 2.26$$

i.e. at low speeds with $\delta = 0^\circ$, the torque characteristic is that of a d.c. shunt motor.

For $\delta = 90^\circ$, equation 2.22 becomes

$$T = \frac{n.K_b.R_{ph}}{Z_{ph}^2}\left[V_{ph}.X_{ph} - E_{ph}.R_{ph}\right] \qquad 2.27$$

i.e.

$$T = \frac{n.K_b.R_{ph}.\omega}{Z_{ph}^2}\left[V_{ph}.L_{ph} - K_b.R_{ph}\right] \qquad 2.28$$

Then for low speeds as $\omega \longrightarrow 0$, $T \longrightarrow 0$. Thus at low operating speeds the optimum load angle to maximise motor torque, is zero degrees.

At high speeds where ω is very large and the phase resistance is negligible compared to the phase reactance X_{ph}, equation 2.23 can be used to predict the motor characteristics and for $\delta = 90^\circ$ gives

$$T = \frac{n}{\omega}\left[\frac{K_b \cdot V_{ph}}{L_{ph}}\right] \qquad 2.29$$

i.e. the torque is proportional to applied phase voltage and inversely proportional to the speed. This is a series d.c. motor type characteristic suitable for traction applications.

A cylindrical rotor synchronous motor operated with position feedback can therefore be started at low frequencies with $\delta = 0^\circ$, and as the speed rises the load angle can be increased up to $\delta = 90^\circ$ giving a series motor characteristic.

2.4 Predicted Characteristics for the Magslip Machine

The small synchronous machine used for much of the development work in the project was a Muirhead Magslip Control Transmitter. The machine has a three phase star connected stator, which is sinusoidally distributed and a two pole wound field cylindrical rotor. The machine was in no way intended for use as a synchronous motor when designed but it can be used as a high quality low power motor. The data sheet is included for reference in Appendix 2A .

In order to be able to predict the performance of the magslip when fed from a variable frequency sinusoidal supply, it is necessary to determine suitable values of phase resistance R_{ph}, phase inductance L_{ph}, and back-emf constant K_b, for use in equation 2.22. The magslip was driven at a speed of 3000 rpm by a d.c. motor and the open and short circuit characteristics shown in fig. 2.3 were obtained. A reference field current value of 0.7 amps was selected from fig. 2.3 within the unsaturated region of operation.

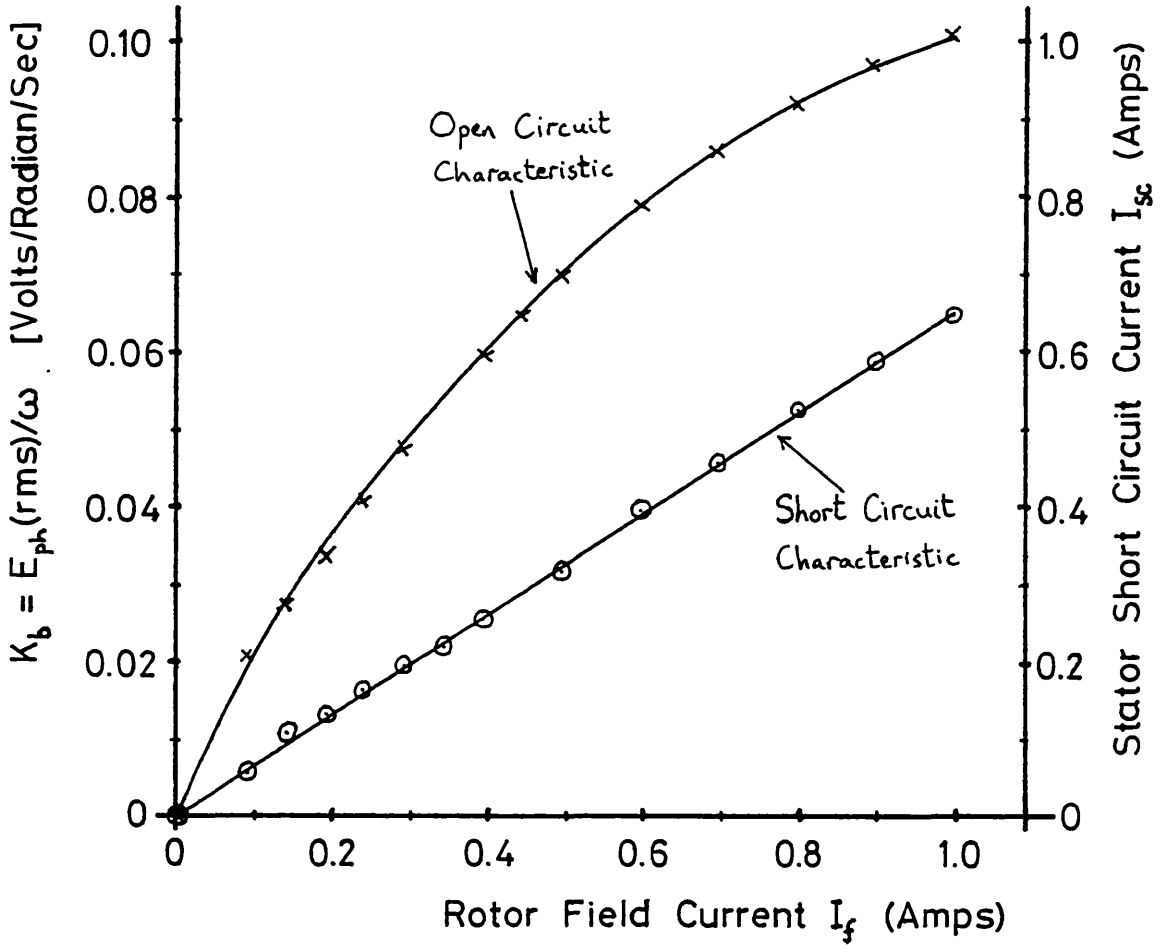


Fig. 2.3 The Magslip Open and Short Circuit Characteristics

At the chosen field current, the back-emf constant K_b is obtainable directly from fig. 2.3 and is found to be

$$K_b = 0.085 \text{ volt seconds per radian} \qquad 2.30$$

In addition, the short circuit current I_{sc}, obtainable from fig. 2.3 is

$$I_{sc} = 0.45 \text{ amps} \qquad 2.31$$

The phase winding impedance Z_{ph} is the ratio of the open circuit phase voltage divided by the short circuit phase current, which at an operating frequency of ω radians per second can be written as

$$Z_{ph} = \frac{K_b . \omega}{I_{sc}} \qquad 2.32$$

The test frequency at 3000 rpm is 100π rad/s and so the

phase impedance for a field excitation current of 0.7 amps is:

$$Z_{ph} = \left[\frac{0.085 \times 100\pi}{0.45}\right] \quad \Omega$$

$$Z_{ph} = 59.34 \ \Omega \qquad 2.33$$

The magslip phase resistance R_{ph} is given as 9.5Ω on the manufacturers data sheet, and by rearranging equation 2.20 the phase synchronous reactance X_{ph} can be determined for the test frequency.

i.e. $$X_{ph} = \sqrt{Z_{ph}^2 - R_{ph}^2} \quad \Omega \qquad 2.34$$

$$X_{ph} = \sqrt{59.34^2 - 9.5^2} \quad \Omega$$

$$X_{ph} = 58.57 \ \Omega \qquad 2.35$$

Equation 2.19 can then be used to determine the phase inductance L_{ph}:

$$L_{ph} = \frac{X_{ph}}{\omega} \qquad 2.36$$

$$L_{ph} = \left[\frac{58.57}{100\pi}\right] \quad \text{Henries}$$

$$L_{ph} = 0.186 \quad \text{Henries} \qquad 2.37$$

A Fortran computer program was written to calculate various motor quantities, including torque and phase current, over a wide speed range for particular combinations of fixed phase voltage and load angle. The program flow chart is shown in fig. 2.4.

The equations used to calculate the results can be

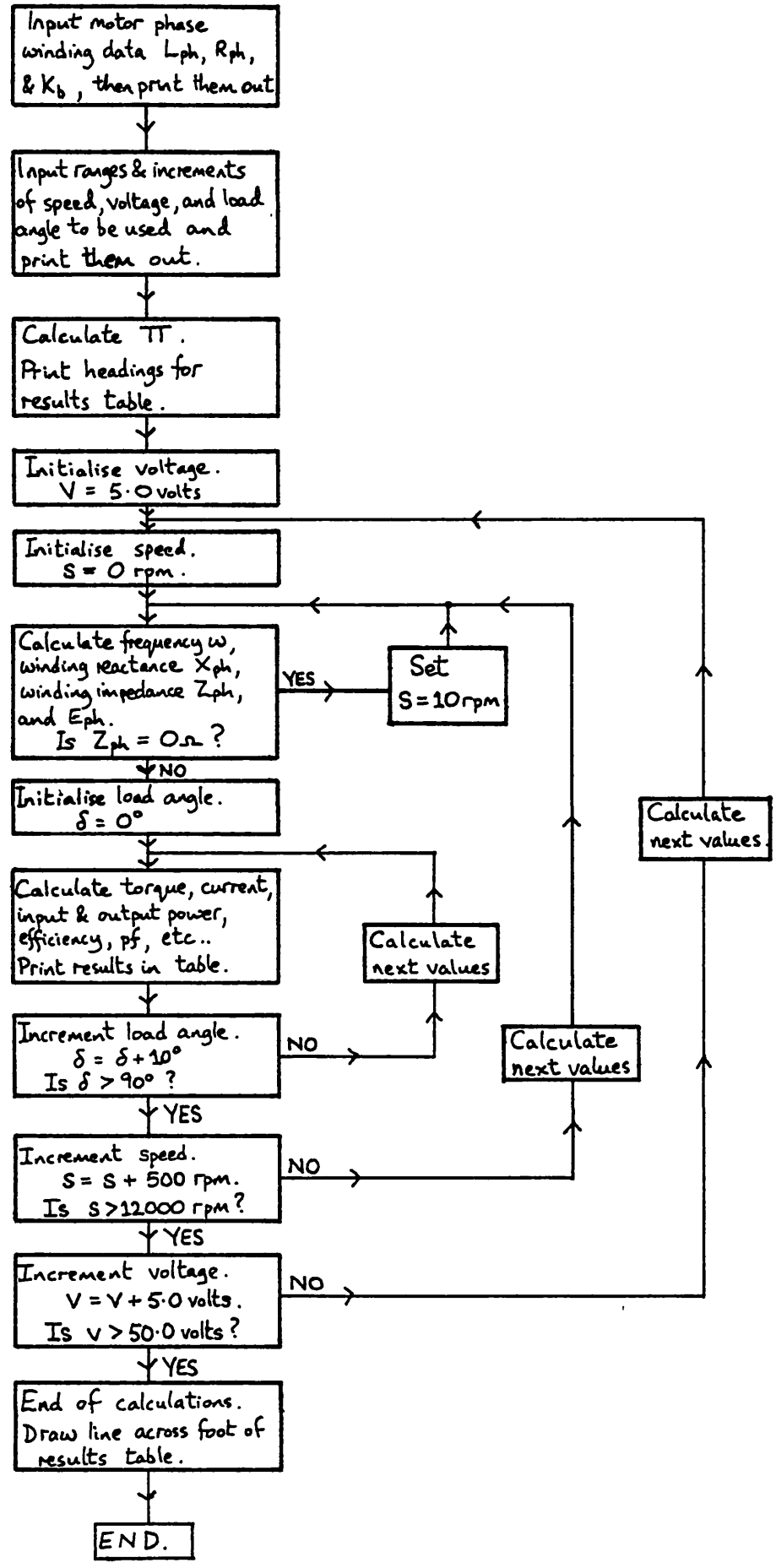


Fig. 2.4. Flow Chart For The Magslip Performance Program

summarised as follows. The angular velocity ω of the system is related to the speed of the motor, M, by the expression

$$\omega = \frac{2\pi.M}{60} \quad \text{radians/second} \qquad 2.38$$

where M is measured in revolutions per minute (rpm). The motor torque is calculated using equations 2.16 and 2.15 with the phase number n set to a value of 3.

The motor phase current is obtained from the phasor diagram of fig. 2.2. The voltage drop across the winding impedance Z_{ph} can be calculated using the cosine rule:

$$\left[I_{ph}.Z_{ph}\right]^2 = V_{ph}^2 + E_{ph}^2 - 2.V_{ph}.E_{ph}.\cos\delta \qquad 2.39$$

from which

$$I_{ph} = \sqrt{\left[\frac{V_{ph}^2 + E_{ph}^2 - 2.V_{ph}.E_{ph}.\cos\delta}{Z_{ph}^2}\right]} \qquad 2.40$$

The total motor output power P_{out} is calculated from

$$P_{out} = T.\omega \qquad 2.41$$

The power factor is the most awkward quantity to calculate, and the angles involved in its calculation are shown on the phasor diagram of fig. 2.5. The power factor p is given by

$$p = \cos\phi \qquad 2.42$$

where

$$\phi = \left[\frac{\pi}{2} - (\alpha + \gamma)\right] \quad \text{radians} \qquad 2.43$$

It is not possible to calculate γ reliably by simply using

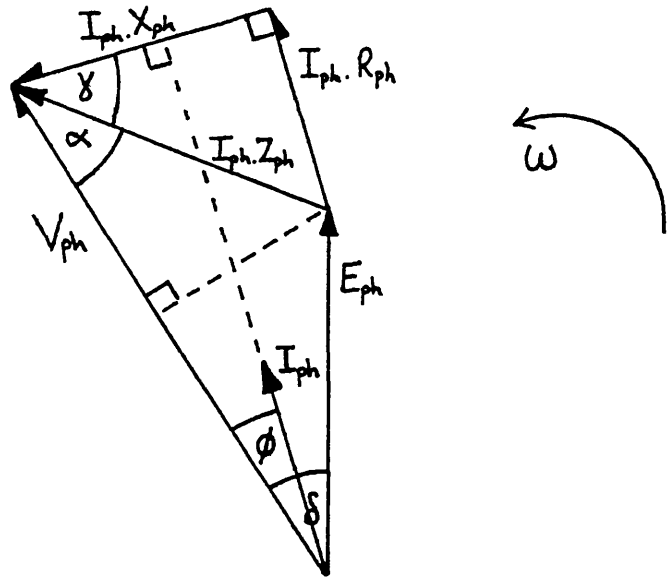


Fig. 2.5 Synchronous Motor Phasor Diagram Showing the Angles Used in the Calculation of the Power Factor

γ = arctan (R_{ph}/L_{ph}) since when the frequency is zero, X_{ph} is also zero, causing an indefinite result for the division. Therefore the angle α is calculated as shown below.

$$\gamma = \left[\frac{\pi}{2} - \arctan\left(\frac{X_{ph}}{R_{ph}}\right)\right] \text{ radians} \qquad 2.44$$

The angle α is found by resolving the $I_{ph}.Z_{ph}$ phasor and the E_{ph} phasor along the V_{ph} phasor.

$$\text{i.e. } V_{ph} = \left(I_{ph}.Z_{ph}.\cos\alpha\right) + \left(E_{ph}.\cos\delta\right) \qquad 2.45$$

This gives

$$\alpha = \arccos\left[\frac{\left(V_{ph} - E_{ph}.\cos\delta\right)}{I_{ph}.Z_{ph}}\right] \qquad 2.46$$

Computer rounding errors cause the argument of the arccos function to fractionally exceed +1 or -1 for values of α of 0 and π radians respectively. To overcome this, it is necessary to truncate the calculated value of the argument, if necessary, prior to using the arccos function.

The total input power to the motor, P_{in}, is given by

$$P_{in} = 3.V_{ph}.I_{ph}.p \qquad 2.47$$

The motor efficiency η is then given by

$$\eta = \frac{P_{out}}{P_{in}} \qquad 2.48$$

This efficiency neglects any iron losses within the motor and does not include the power consumed by the field winding. However, since the field flux is often supplied by permanent magnets in small machines, the second omission is not serious.

Finally the motor torque angle ψ, shown in fig. 2.6, can be calculated by using

$$\psi = \left[\frac{\pi}{2} + \delta - \phi\right] \quad \text{radians} \qquad 2.49$$

The torque angle in a cylindrical rotor synchronous machine is the angle between the axes of the stator and field magnetomotive forces (mmf's).

The program was run for a range of phase voltages from 5 volts to 45 volts, load angles from 0° to 90°, and speeds from 0 rpm to 12000 rpm. The results were produced in tabulated form. The torque/speed characteristics for load angles of 0°, 30°, 60°, and 90° at a phase voltage of 20 volts are shown plotted in fig. 2.7.

It can be seen that a load angle of 0° results in a torque/speed characteristic with high starting torque, but the torque falls rapidly to zero as the speed increases. A load angle of 90° results in zero starting torque, but there is much more torque produced at high speeds. This

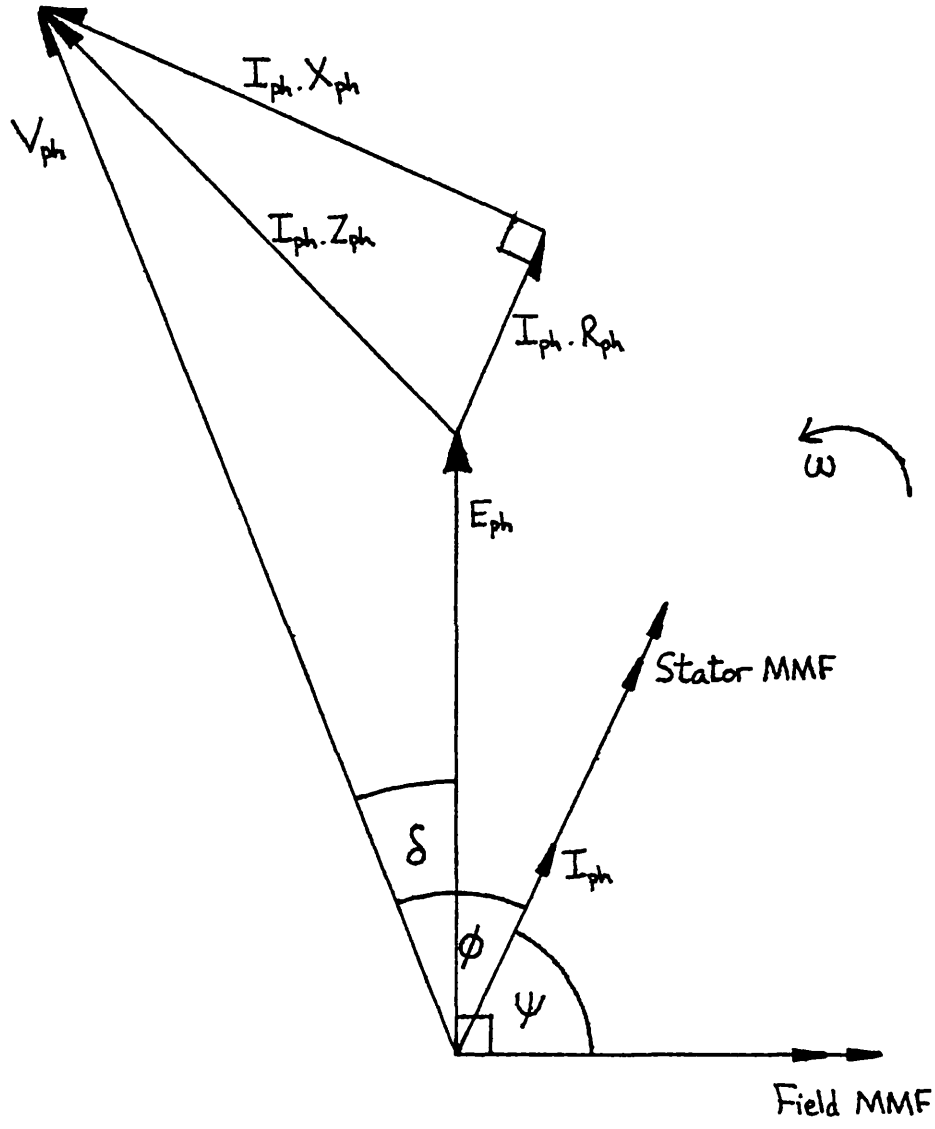


Fig. 2.6. Phasor Diagram Showing The Torque Angle ψ Of A Synchronous Motor.

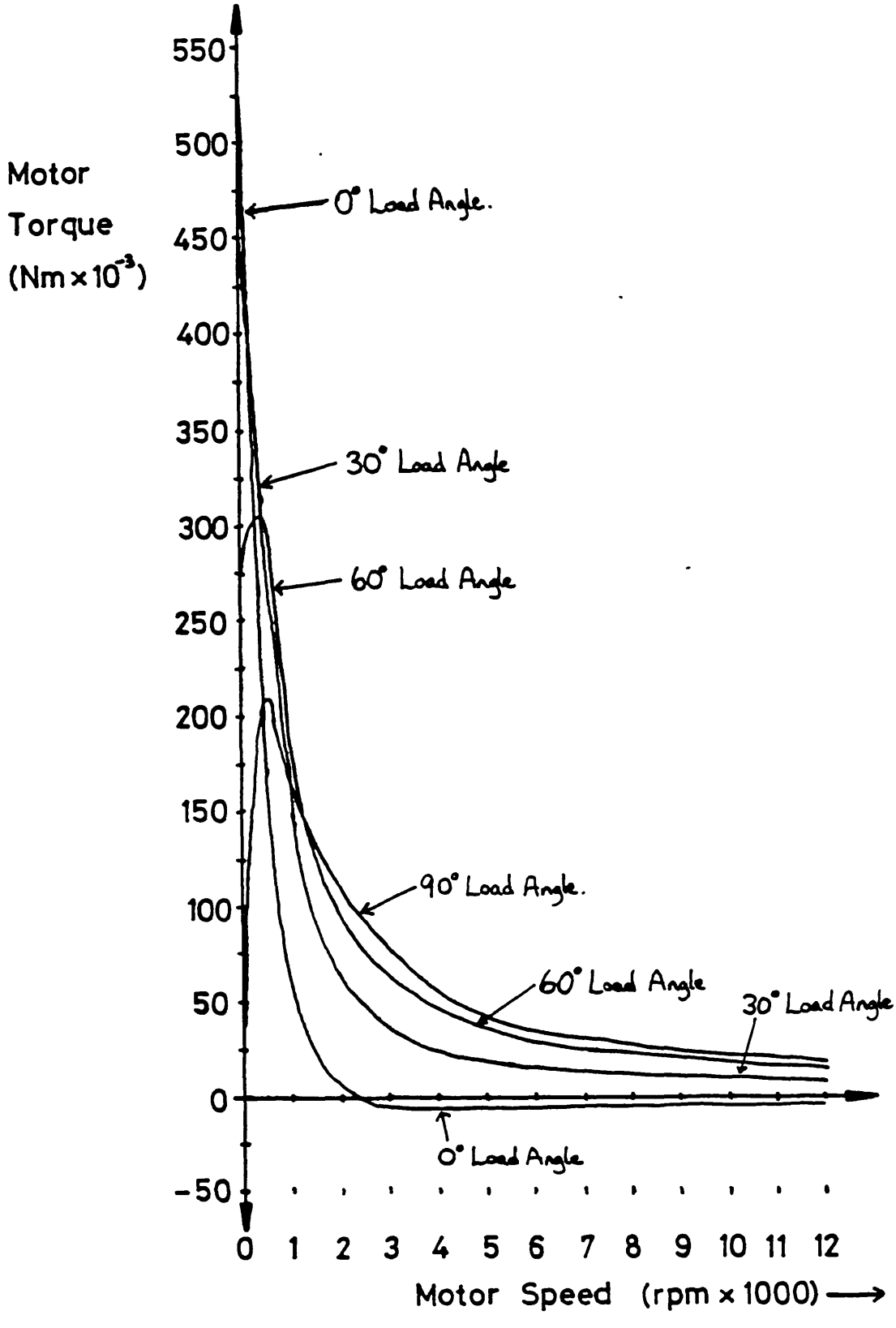


Fig. 2.7. Predicted Torque/Speed Characteristics For Various Load Angles And V_{ph} = 20 Volts.

enables the motor to run faster on the same phase volts than it can with a 0° load angle. Load angles between 0° and 90° result in torque/speed characteristics that come within the bounds set by the 0° and 90° characteristics. The way in which the four curves in fig. 2.7 intersect suggests that there is an optimum load angle that can be selected at a given speed in order to maximise the available torque, and to achieve high motor speeds the load angle must be set to 90°. This confirms the results published by Chalmers et al (2.9).

The various other motor quantities such as phase current and efficiency have not been plotted here since the main point to be made at this stage is the importance of load angle control in an autopiloted motor, if it is to operate effectively over a wide speed range. Some of the results are presented in Chapter 3, where they are plotted for comparison with the measured test results, obtained when the magslip was driven by a quasi-square wave 3 phase voltage-source inverter.

The computer program was modified to enable the torque/speed results to be plotted by the computer graphics system. The program was arranged to produce a set of curves for a fixed load angle at phase voltages of 5, 15, 25, 35, and 45 volts. Figures 2.8, 2.9, 2.10, and 2.11 show the families of curves obtained for load angles of 0°, 30°, 60°, and 90° respectively. These results show that the phase voltage affects the magnitude of the torque/speed curve, but it does not change the shape of the curve significantly. The load angle is therefore the important parameter that fixes the torque/speed characteristic of a given motor. Curves were also produced for load angles in the range -180° to 0° and 90° to 180°. However, these load angles do not produce useful motoring torque/speed characteristics and are not reproduced here.

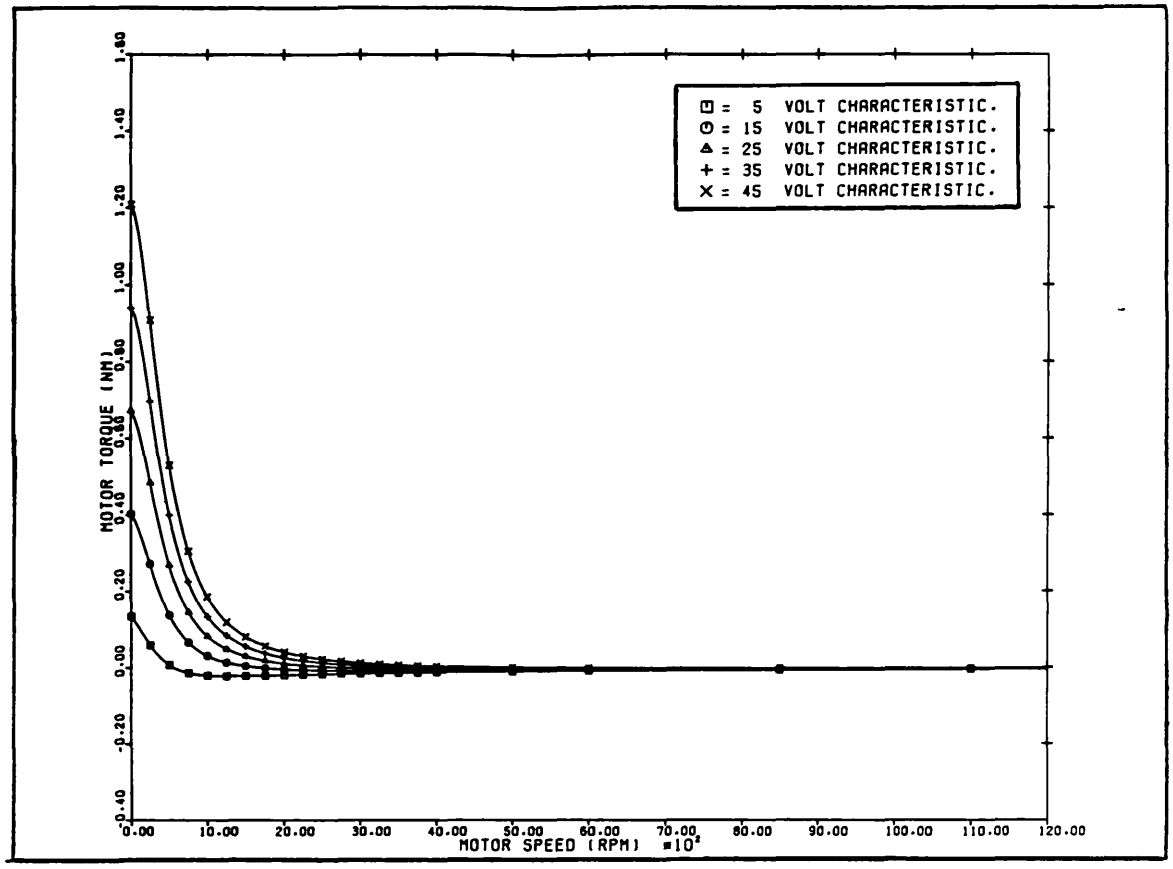


Fig.2.8. Predicted Torque/Speed Curves For 0° Load Angle.

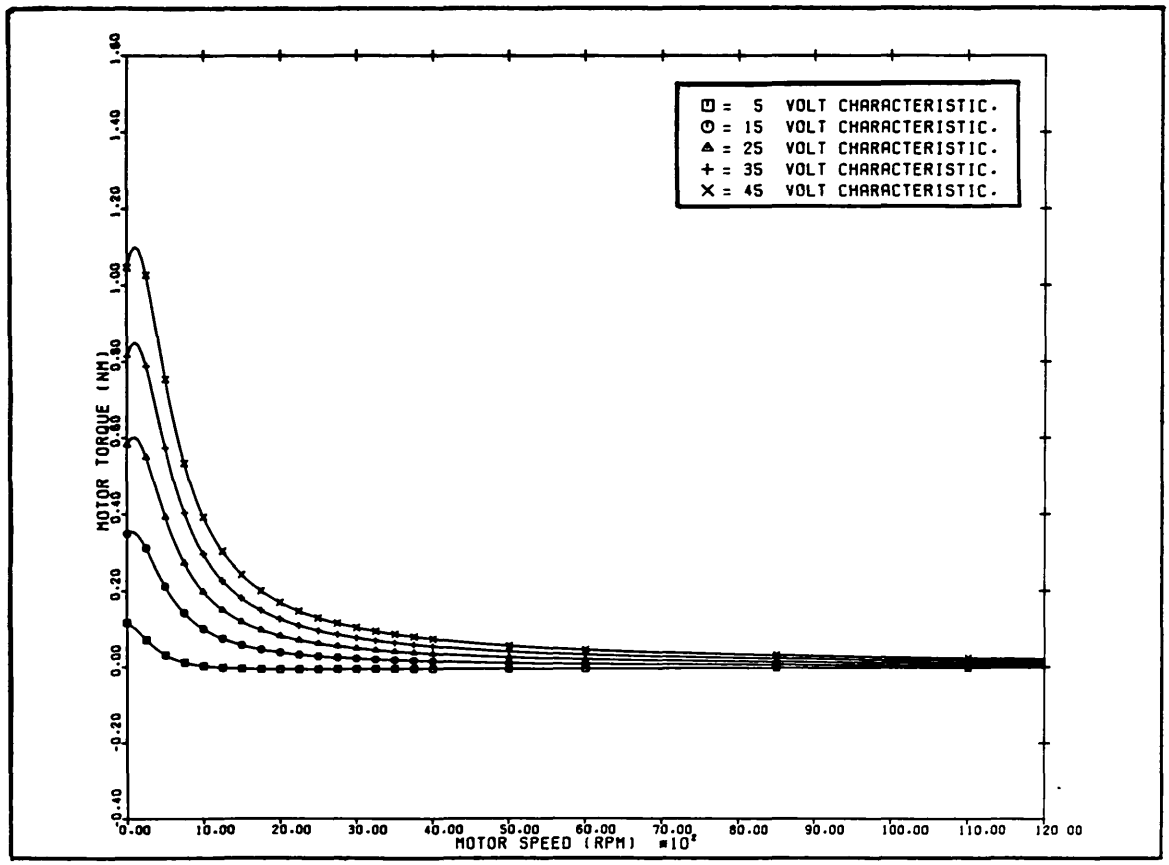


Fig.2.9. Predicted Torque/Speed Curves For 30° Load Angle.

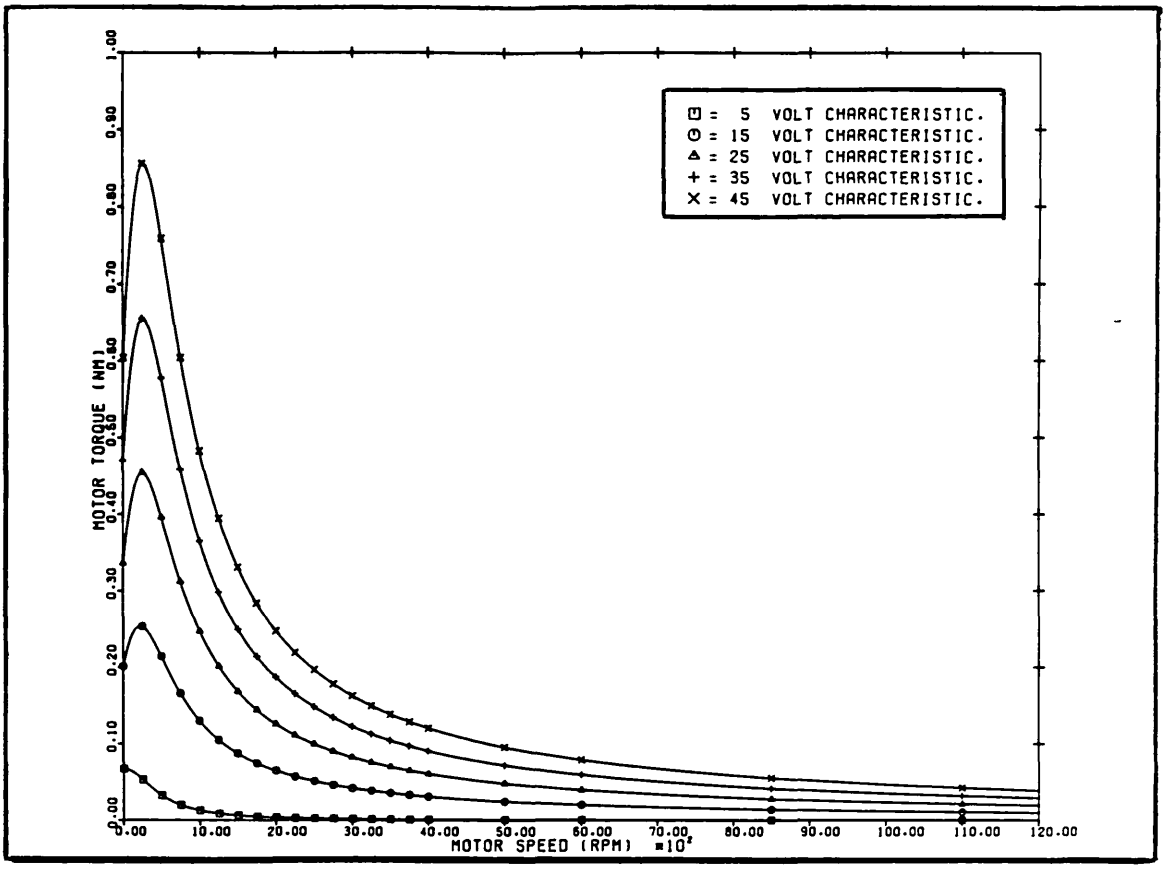


Fig.2.10. Predicted Torque/Speed Curves For 60° Load Angle.

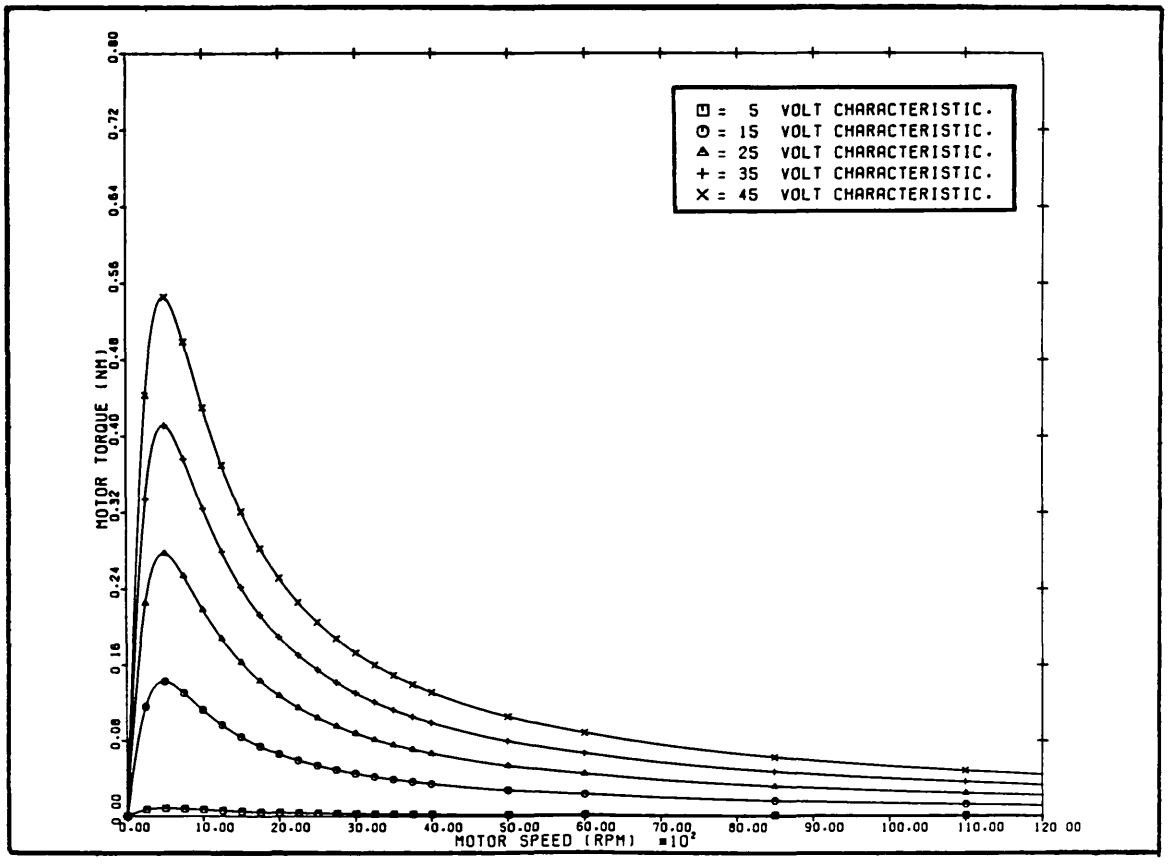


Fig.2.11. Predicted Torque/Speed Curves For 90° Load Angle.

2.5 The Optimum Load Angle for Maximum Torque at a Given Speed

The cylindrical rotor torque equation 2.22 has three variables V_{ph}, δ, and ω that can vary during operation. (This assumes K_b is a fixed parameter.) Under constant speed conditions (ω = constant) with the applied voltage V_{ph} also held constant, the torque T is then solely a function of the load angle δ. At any particular speed there is a load angle which results in the maximum torque being developed for the given ω and V_{ph}. This load angle can be determined by differentiation of equation 2.22 with respect to the load angle δ .

$$\text{i.e.} \quad \frac{dT}{d\delta} = n\left(\frac{K_b}{Z_{ph}^2}\right)\left[V_{ph}.X_{ph}.\cos\delta - V_{ph}.R_{ph}.\sin\delta\right] \qquad 2.50$$

The maximum torque occurs when $dT/d\delta = 0$. Equating equation 2.50 to zero gives

$$\tan\delta = \frac{X_{ph}}{R_{ph}} \qquad 2.51$$

$$\text{i.e.} \quad \delta = \arctan\left[\frac{\omega.L_{ph}}{R_{ph}}\right] \qquad 2.52$$

i.e. the maximum torque at a given speed occurs when the load angle δ is made equal to the arctangent of the machine phase winding reactance divided by the phase winding resistance. Then V_{ph} leads E_{ph} by the same angle as the current I_{ph} lags the impedance voltage drop $I_{ph}.Z_{ph}$. The motor speed M can be substituted into equation 2.52 by using equation 2.38 to give

$$\delta = \arctan\left[\frac{2\pi}{60}\left(\frac{M.L_{ph}}{R_{ph}}\right)\right] \qquad 2.53$$

where M is in rpm. At zero speed, the optimum load angle is therefore zero degrees as deduced in section 2.3.2 from the simplified equations 2.26 and 2.28. As the speed inceases towards infinity, the optimum load angle settles at a maximum value of 90°. The motor then has a series d.c. characteristic as shown earlier by equation 2.29.

It is interesting to note that in the conventional fixed frequency analysis of synchronous machines, the derivative $dP/d\delta$ $(\equiv dT/d\delta)$ is used to obtain a relationship between the available synchronising power and the load angle, as shown by Say (2.13). Therefore, for variable frequency operation, the effect of optimising the load angle δ by use of the $dT/d\delta$ derivative is to actually ensure that the machine operates at its peak synchronising torque at all times. The torque characteristic plotting program was modified so that the value of load angle used at each speed in the calculations was the optimum value given by equation 2.53.

The torque/speed curves for phase voltages of 5, 15, 25, 35, and 45 volts are shown in fig. 2.12. These are the optimum torque/speed curves for the magslip operated on a sinusoidal voltage supply. In practice it is possible that the load angle will not be continuously variable, but it will increase in discrete increments. It was found that with increments of 5° there was little change from the optimum load angle characteristics. The step size chosen obviously depends on a number of factors, such as the required specification, the position sensor resolution and the desired cost of the system.

Nevertheless, the curves in this section illustrate the fact that in a sinusoidal voltage-fed synchronous motor, the load angle must be carefully adjusted to optimise the performance at a given speed.

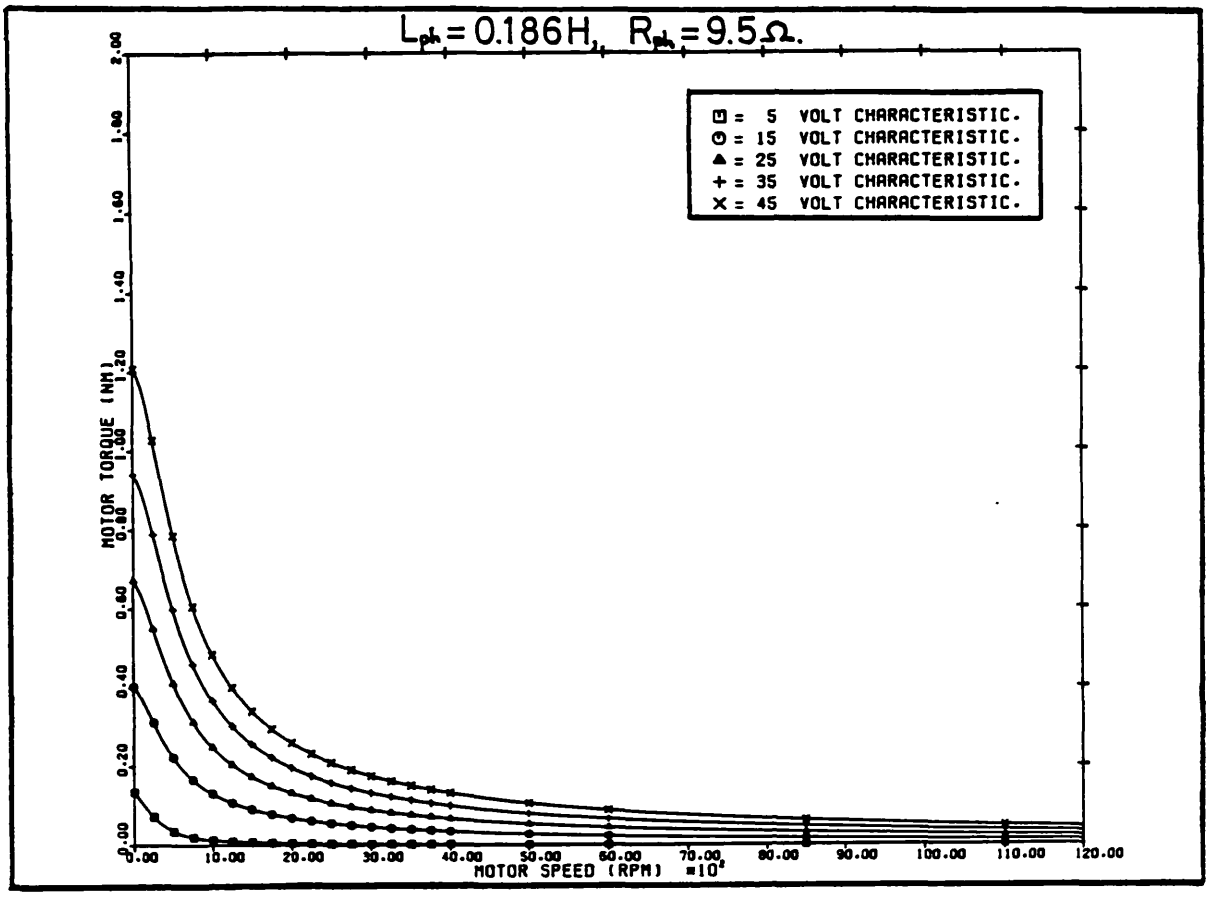


Fig.2.12. Predicted Torque Versus Speed With Load Angle Related To Speed By The Equation: $\delta = \tan^{-1}(X_{ph}/R_{ph})$.

2.6 Derivation of a Speed Versus Applied Stator Voltage Characteristic

It is interesting to determine the relationship between motor speed M and the applied stator voltage V_{ph}, for an autopiloted synchronous motor. The analysis can be easily performed if the load torque T_l is assumed constant with speed. The load angle δ is held fixed at a desired value by the position feedback. Since T_l must equal the motor output torque T, equation 2.21 becomes:

$$T_l = \left(\frac{n.K_b}{Z_{ph}^2}\right)\left[V_{ph}.X_{ph}.\sin\delta + V_{ph}.R_{ph}.\cos\delta - E_{ph}.R_{ph}\right] \qquad 2.54$$

Substituting equations 2.11, 2.19, and 2.20 into 2.54 yields after rearrangement:

$$\left(T_l.L_{ph}^2\right)\omega^2 + n.K_b\left(K_b.R_{ph} - V_{ph}.L_{ph}.\sin\delta\right)\omega$$

$$+ \left(T_l.R_{ph}^2 - n.K_b.V_{ph}.R_{ph}.\cos\delta\right) = 0 \qquad 2.55$$

This is a quadratic equation in ω of the form

$$a\omega^2 + b\omega + c = 0 \qquad 2.56$$

where $$a = T_l.L_{ph}^2 \qquad 2.57$$

$$b = n.K_b\left(K_b.R_{ph} - V_{ph}.L_{ph}.\sin\delta\right) \qquad 2.58$$

and $$c = \left(T_l.R_{ph}^2 - n.K_b.V_{ph}.R_{ph}.\cos\delta\right) \qquad 2.59$$

The value of angular velocity ω is given by the standard quadratic solution:

$$\omega = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \qquad 2.60$$

and two values of ω are produced. The value of ω at which the motor will usually run (with no speed control loop) is the one for which $(dT/d\omega)$ is negative, since that is a stable condition for a motor to drive a load. A positive $(dT/d\omega)$ means that, with most loads, the motor has a tendency to accelerate at the value of ω that the gradient is evaluated for, because the output torque is increasing as the speed increases. If both values of ω give negative slopes, the motor will accelerate up to the lowest positive value of ω given by equation 2.40. It will not be able to reach the higher value of ω unless it is already at or above that value when the particular phase voltage is applied, or if it is assisted up to that angular velocity by an external torque.

Therefore, in general, the motor will usually run at the lowest positive value of angular velocity consistent with a negative torque/speed slope.

A simple Fortran program was written to calculate and plot the speed/voltage characteristic of the magslip. The program flow chart is shown in fig. 2.13 and the resulting curves for various load angles of 0°, 30°, 60°, 90° and load torques of 0.017, 0.1, 0.3, and 0.6 newton metres, are shown in figures 2.14, 2.15, 2.16, and 2.17 respectively. The characteristics are reasonably linear but the speed is in no way directly proportional to the applied voltage. This can be confirmed by differentiating the speed/voltage equation 2.60 with respect to the voltage. The result obtained is

$$\frac{d\omega}{dV_{ph}} = U\left[V + \frac{W}{2\sqrt{Y}}\right] \qquad 2.61$$

where $$U = 1/(2.T_l.L_{ph}^{\,2}) \qquad 2.62$$

$$V = 3.K_b.L_{ph}.\sin\delta \qquad 2.63$$

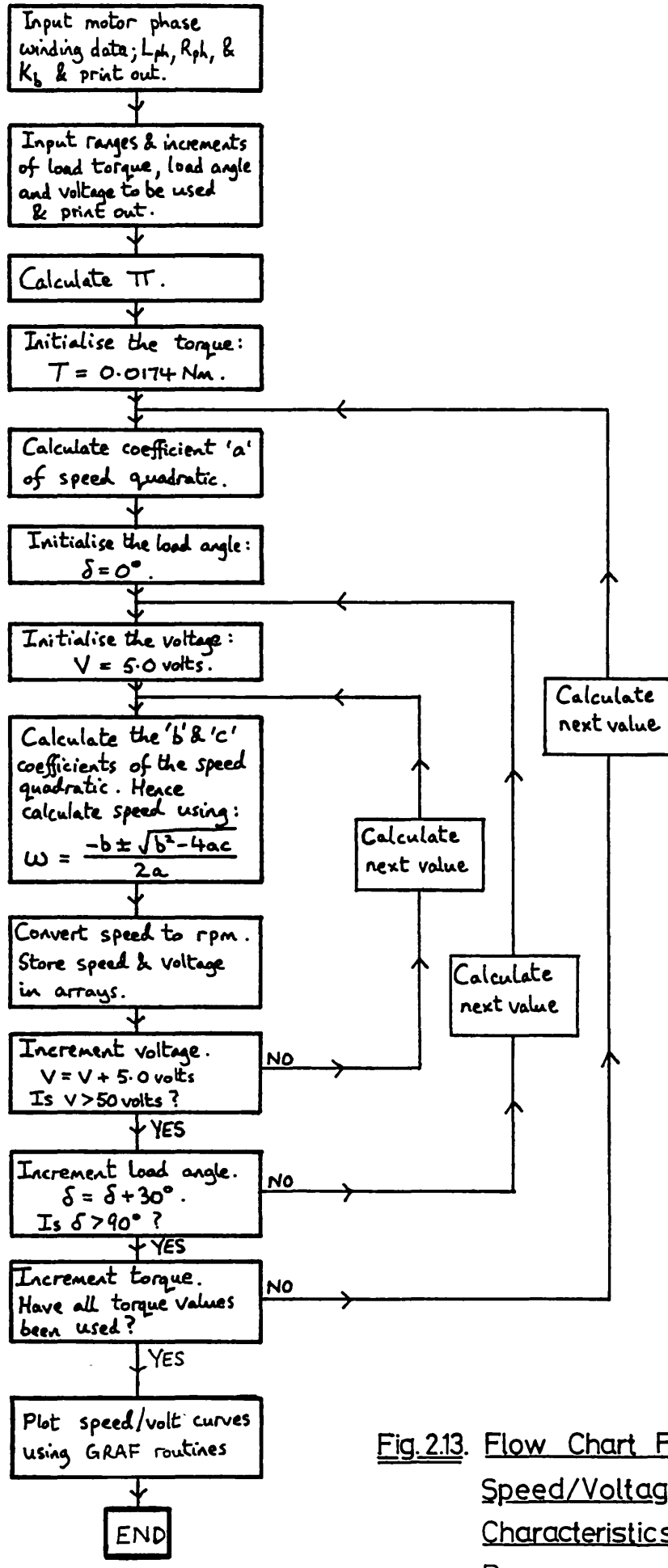


Fig. 2.13. Flow Chart For The Speed/Voltage-Characteristics Program.

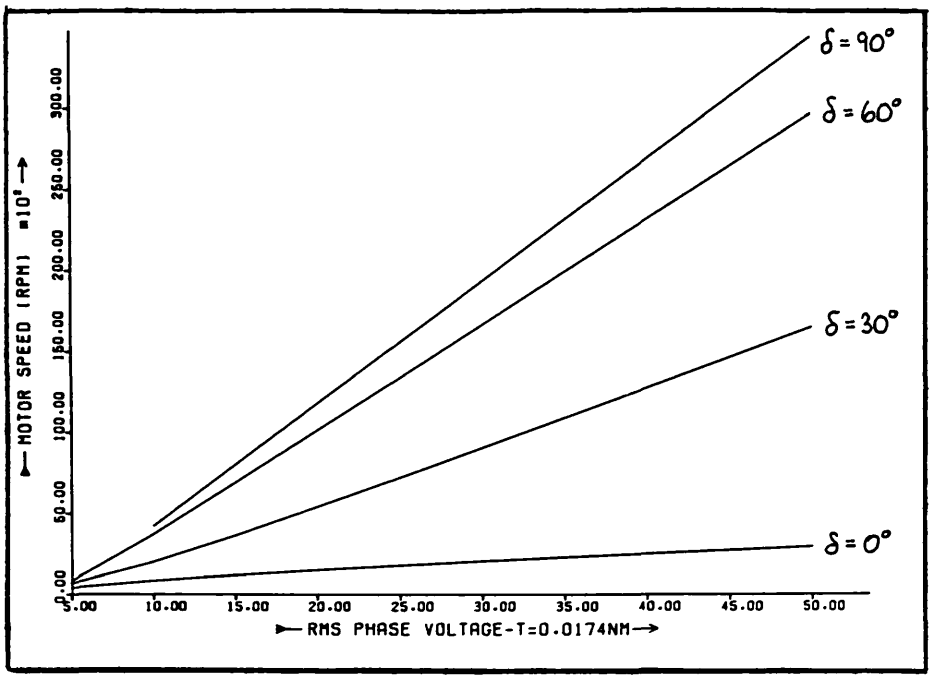


Fig.2.14. Predicted Speed Versus Voltage Curves With Load Torque Of 0.0174 Nm.

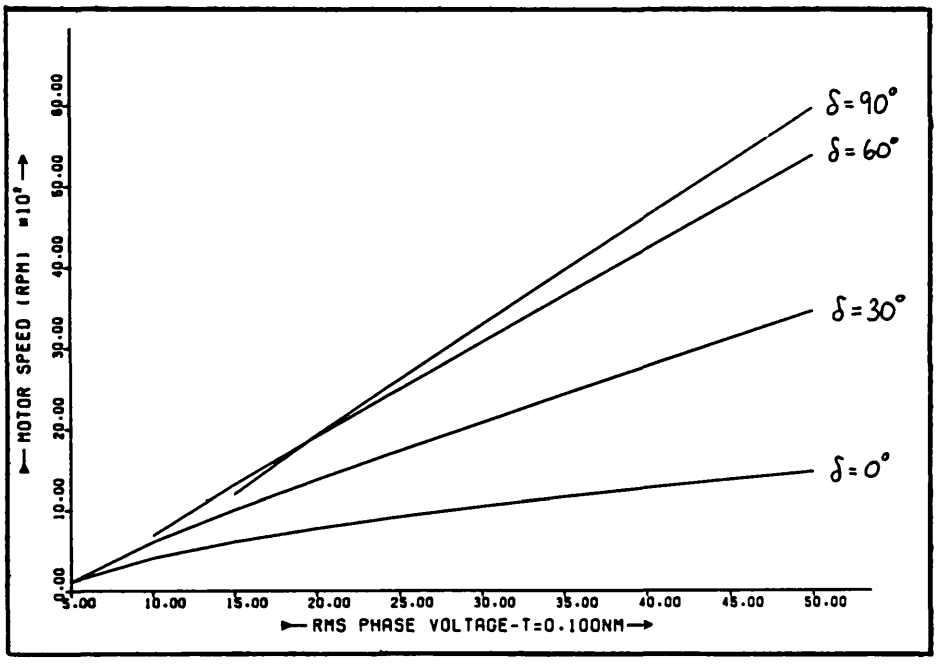


Fig.2.15. Predicted Speed Versus Voltage Curves With Load Torque Of 0.100 Nm.

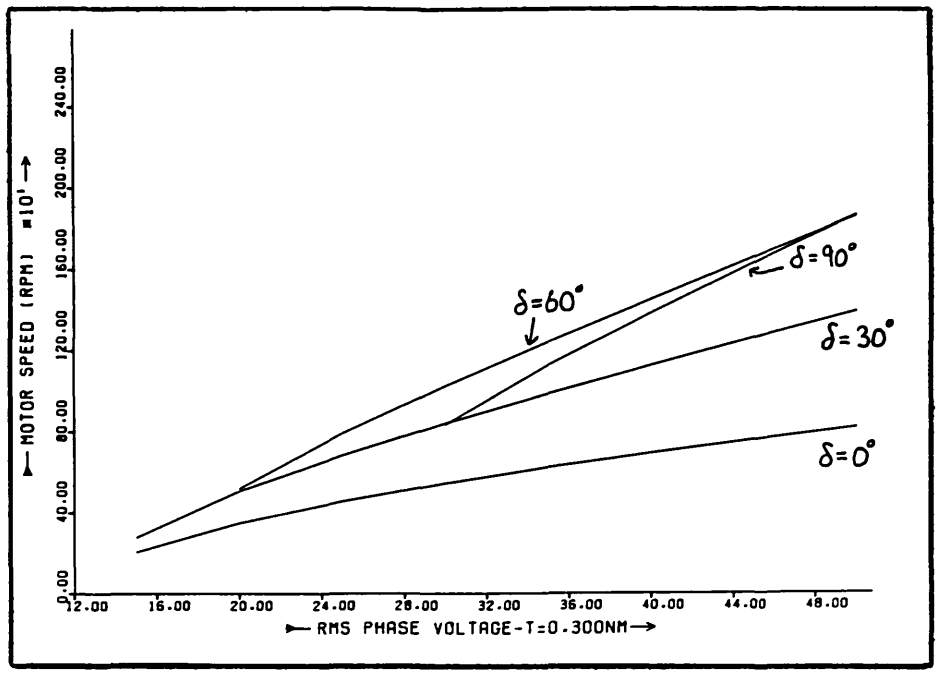


Fig.2.16. Predicted Speed Versus Voltage Curves With Load Torque Of 0.300 Nm.

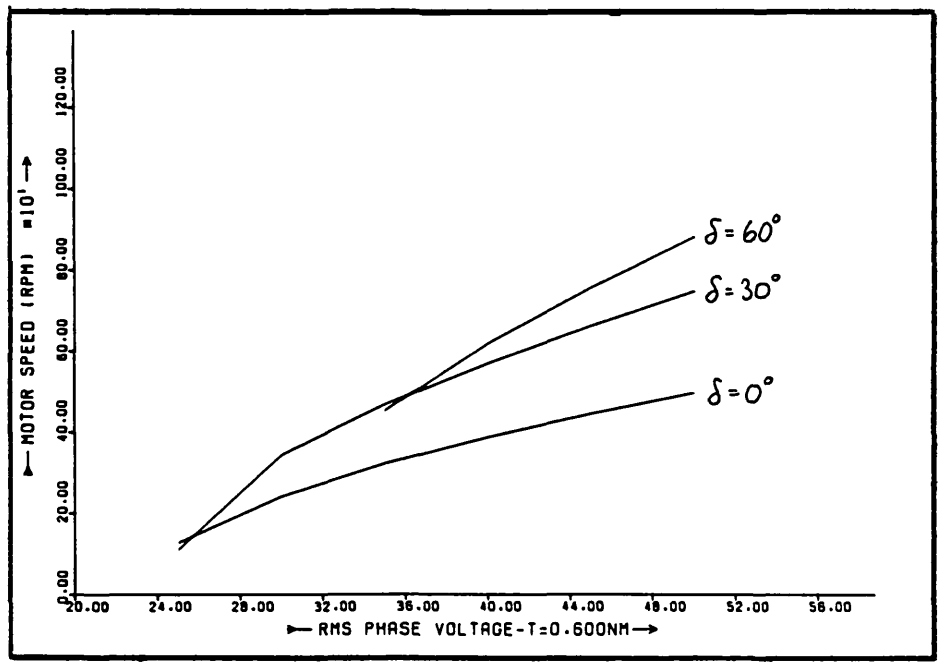


Fig.2.17. Predicted Speed Versus Voltage Curves With Load Torque Of 0.600 Nm.

$$W = \left(-18.K_b^3.R_{ph}.L_{ph}.\sin\delta + 18.K_b^2.L_{ph}^2.V_{ph}.\sin^2\delta + 12.T_l.L_{ph}^2.K_b.R_{ph}.\cos\delta\right) \quad 2.64$$

$$\text{and } Y = 9.K_b^2\left(K_b.R_{ph} - V_{ph}.L_{ph}.\sin\delta\right)^2 - 4.T_l.L_{ph}^2\left(T_l.R_{ph}^2 - 3.K_b.V_{ph}.R_{ph}.\cos\delta\right) \quad 2.65$$

It can be seen that the rate of change of speed with respect to voltage depends directly on the load angle δ, the supply voltage V_{ph} and the load torque T_l.

2.7 Determination of the Optimum Operating Load Angle for a Salient Pole Synchronous Motor

The optimum operating load angle for a salient pole synchronous motor can be derived from the torque expression (equation 2.15). Differentiating 2.15 with respect to δ and equating the result to zero, produces a long expression which after grouping of similar terms yields:

$$A.\cos 2\delta + B.\sin 2\delta + C.\cos\delta + D.\sin\delta = 0 \quad 2.66$$

$$\text{where } A = V_{ph}\left[\omega^2.L_d.L_q\left(L_d - L_q\right) + R_{ph}^2\left(L_q - L_d\right)\right] \quad 2.67$$

$$B = V_{ph}.\omega^2.R_{ph}\left[L_q^2 - L_d^2\right] \quad 2.68$$

$$C = \omega.K_b\left[\omega^2.L_d.L_q^2 + 2.R_{ph}^2.L_d - R_{ph}^2.L_q\right] \quad 2.69$$

$$D = R_{ph}.K_b\left[\omega^2.L_d.L_q - 2.\omega^2.L_q^2 - R_{ph}^2\right] \quad 2.70$$

Coefficients A, B, C, and D are independent of δ and so can be considered as constants in equation 2.66. It is interesting to note that if equation 2.18 is substituted into the four coefficients to impose cylindrical rotor

conditions, equation 2.66 reduces to the simple solution 2.31;

i.e. $$\delta = \tan (X_{ph}/R_{ph})$$

Equation 2.66 cannot be solved without further suitable substitutions to simplify it. The substitution

$$t = \tan (\delta/2) \qquad 2.71$$

can be used to replace the sin and cos terms of equation 2.66. From equation 2.71 it can be shown that:

$$\sin\delta = \frac{2t}{(1+t^2)} \qquad 2.72$$

$$\cos\delta = \frac{(1-t^2)}{(1+t^2)} \qquad 2.73$$

$$\cos 2\delta = \frac{t^4-6t^2+1}{(1+t^2)^2} \qquad 2.74$$

and $$\sin 2\delta = \frac{4t(1-t^2)}{(1+t^2)^2} \qquad 2.75$$

Substitution of equations 2.72, 2.73, 2.74, and 2.75 into equation 2.66 leads to a fourth order polynomial in t:

$$t^4 + a_3.t^3 + a_2.t^2 + a_1.t^1 + a_0 = 0 \qquad 2.76$$

where $$a_3 = \frac{(2D - 4B)}{(A - C)} \qquad 2.77$$

$$a_2 = \frac{-6A}{(A - C)} \qquad 2.78$$

$$a_1 = \left[\frac{4B + 2D}{A - C}\right] \qquad 2.79$$

$$a_o = \left[\frac{A + C}{A - C}\right] \qquad 2.80$$

The solution of a fourth order polynomial is not an easy matter but it can be achieved analytically. The method is described in appendix 2B. The analysis involves complex algebra, and the solution has four roots which can be real, imaginary or complex. The four values of t must be examined to determine which one is the correct solution. There is no real angle associated with a purely imaginary root. The angle associated with a complex root depends only on its real component. The load angles δ associated with each real or complex root t can be found by rearranging equation 2.71:

$$\delta = 2 \arctan (t) \qquad 2.81$$

For each value of t there are two angles θ and $(\theta + \pi)$ that the arctangent function produces between 0 and 2π radians. Therefore the values of δ given by equation 2.81 are 2θ and $(2\theta + 2\pi)$; i.e. there is only one value of δ for each value of t. Any angle outside the range 0° to 90° is not of interest. Finally, if there is more than one angle within the permitted range it is necessary to choose the angle which results in a maximum in the torque expression.

Differentiation of the torque equation 2.16 gives an equation of the form:

$$\frac{dT}{d\delta}\left[R_{ph}{}^2 + \omega^2.L_d.L_q\right]\frac{\omega}{n} = A.\cos 2\delta + B.\sin 2\delta + C.\cos\delta + D.\sin\delta \qquad 2.82$$

where A, B, C, and D are defined by equations 2.67, 2.68, 2.69, and 2.70. Differentiating again gives:

$$\frac{d^2T}{d\delta^2}\left[R_{ph}{}^2 + \omega^2.L_d.L_q\right]\frac{\omega}{n} = -2A.\sin 2\delta + 2B.\cos 2\delta - C.\sin\delta + D.\cos\delta \qquad 2.83$$

Now $(R_{ph}^2 + \omega^2.L_d.L_q)^2\,\omega/n$ is positive. Therefore the sign of $d^2T/d\delta^2$ is determined by the RHS of equation 2.83. The relevant solutions for δ are substituted into equation 2.83 and the angle associated with maximum torque gives:

$$\frac{d^2T}{d\delta^2} < 0 \qquad 2.84$$

A program was written to calculate and plot the optimum torque/speed characteristic for a salient pole machine. The program structure was essentially the same as the program used for the cylindrical rotor work, the main difference being the subroutine required to calculate the optimum load angle. A set of curves of optimum load angle versus speed for five different phase voltages is shown in fig. 2.18. The magslip phase inductance and phase resistance values were used for the program and a saliency ratio of 0.6 was assumed for the machine.

i.e. $L_d = 0.186$ H 2.85

$L_q = 0.112$ H 2.86

$R_{ph} = 9.5\ \Omega$ 2.87

The curves on figure 2.18 show that at low speed a negative value of load angle gives the optimum motor torque. It is also noticeable that the load angle depends very much on the phase voltage at any particular speed. A set of optimum torque/speed curves for the "hypothetical" salient pole magslip are shown in fig. 2.19 and they have the same form as the cylindrical rotor characteristics.

The program was checked for correct operation by putting $L_d = L_q = 0.186$ H. The results obtained were identical to those produced by the cylindrical rotor programs.

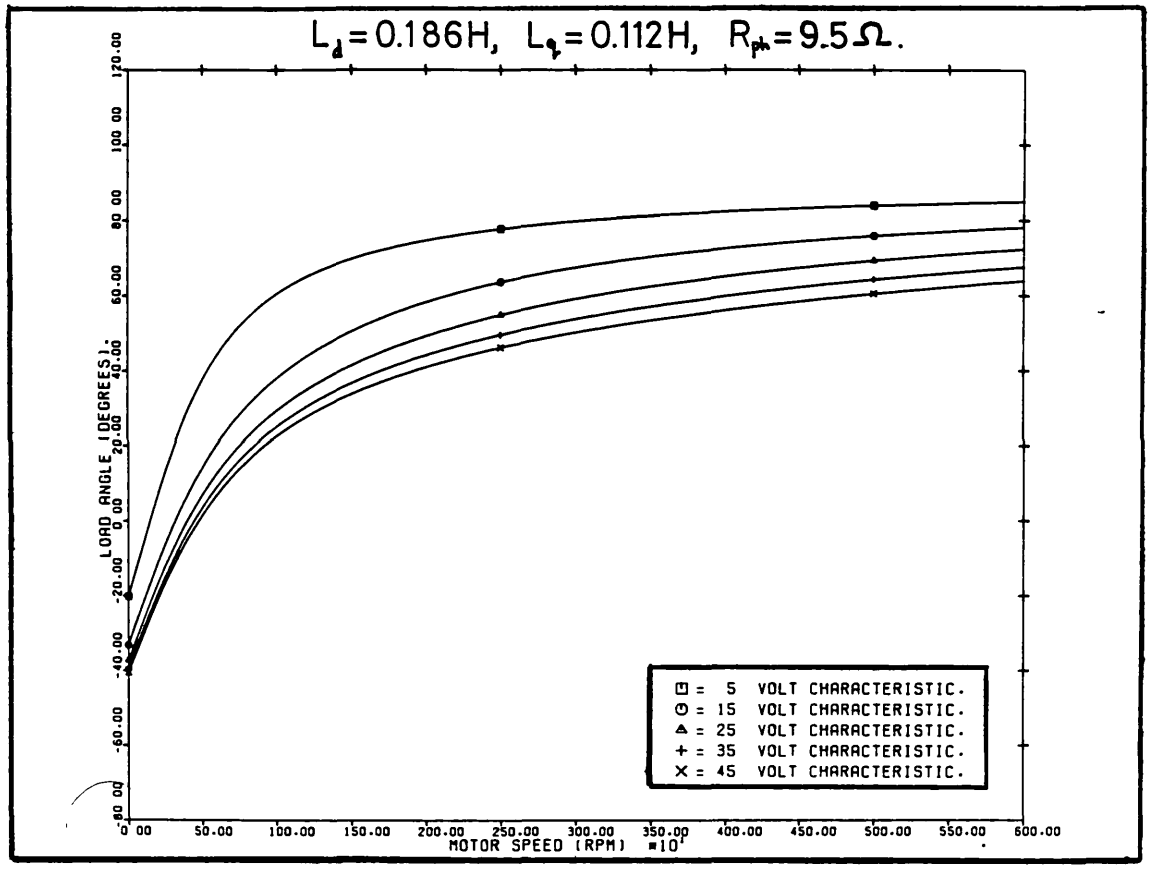


Fig.2.18. Predicted Optimum Load Angle Versus Speed For A Salient Pole Synchronous Motor.

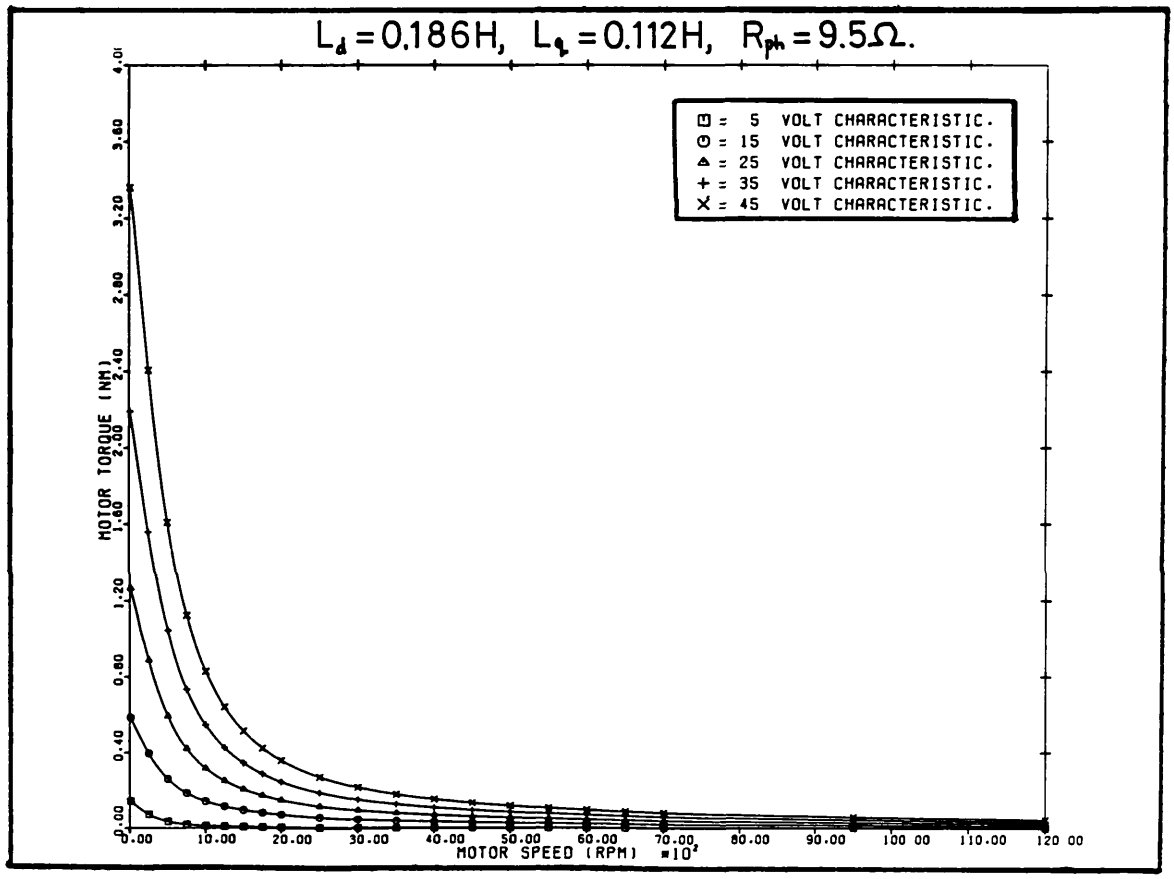


Fig.2.19. Predicted Torque Versus Speed For A Salient Pole Synchronous Motor With Load Angle At Any Speed Set To The Optimum Value Given By Equation 2.66.

The analysis demonstrates that accurate load angle selection is important if optimum performance is to be obtained from a voltage-fed autopiloted salient pole synchronous motor. However, the work was not pursued beyond this stage since the motors used in the practical work were not salient.

2.8 Attempts at Square Wave Analysis

Although the sinusoidal analysis produces results that can be applied to quasi-square wave systems with reasonable success, it was thought desirable to develop a computer program which could work with a square wave applied phase voltage and even a non-sinusoidal back-emf. A great deal of effort was expended on this work over a period of months, but unfortunately no successful results were obtained in the time available. The short summary of the work given below outlines the method used and discusses the possible reason for its failure.

It was decided that a useful first step would be to analyse a star connected three phase sinusoidally distributed machine, fed with quasi-square wave phase voltages as shown in fig. 2.20.

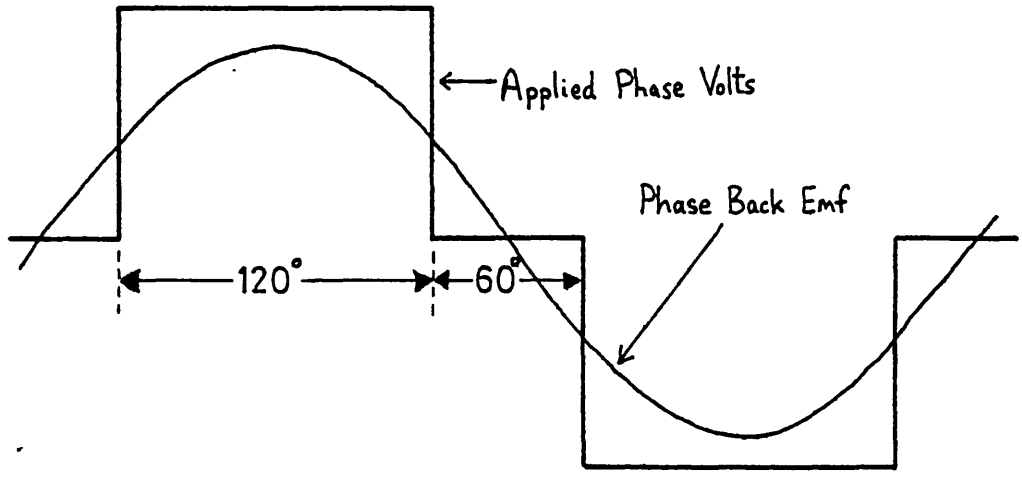


Fig. 2.20. 120°-60° Quasi-Square Applied Phase Voltage Waveform

To simplify the analysis it was assumed that the winding star point was connected to ground. If the program worked, a floating star point was to be incorporated to allow the magslip to be analysed. The inverter that generates the phase voltages was assumed to be of the form shown in fig. 2.21.

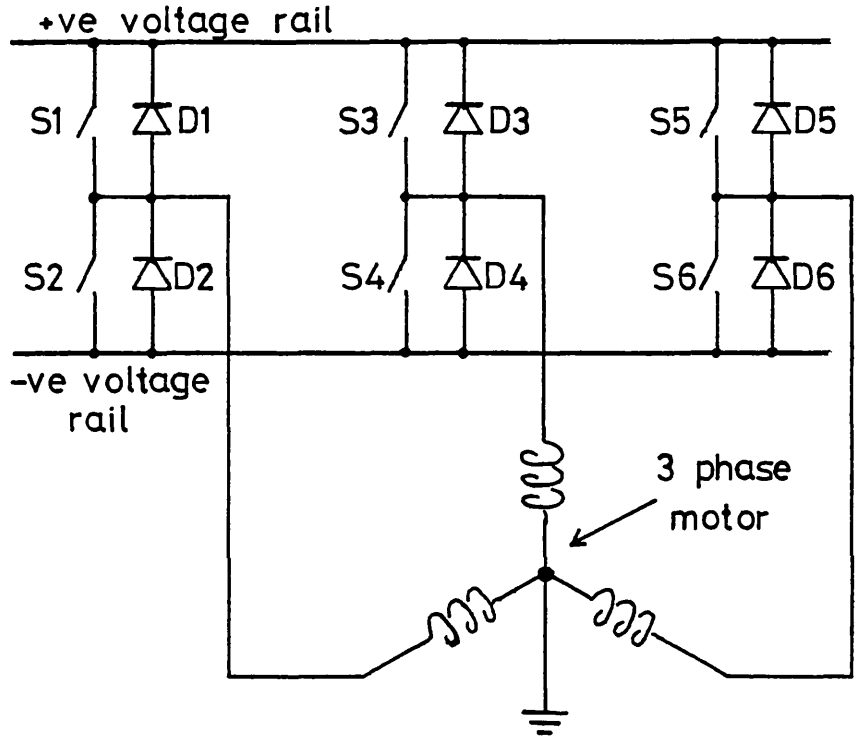


Fig.2.21. Inverter Circuit Used For The 'Square-Wave' Analysis

The transistors are represented by switches S1 to S6 and the freewheel paths by diodes D1 to D6. The inverter was assumed to function in the following manner. If a switch is closed, the appropriate voltage rail is connected to the relevant phase winding. When the switch opens, the current flowing in the phase causes the freewheel diode to connect the phase to the opposite voltage rail until the current falls to zero, at which point the phase is disconnected. Voltage drops in the transistors and diodes are ignored. Rather than analyse the motor using the conventional two axis methods, it was decided to work on the three phase windings directly; i.e. to use a phase analysis.

Thus for a typical phase winding "a" having a phase resistance R_{ph}, a self-inductance of L_{ph}, a mutual inductance of M_{ph}, and a generated emf of E^a_{ph}, the circuit equation at any instant with a phase current I^a flowing is given by

$$V^a_{ph} = L_{ph}\frac{dI^a}{dt} + M_{ph}\frac{dI^b}{dt} + M_{ph}\frac{dI^c}{dt} + I^a.R_{ph} + E^a_{ph} \qquad 2.88$$

where V^a_{ph} is the applied phase voltage and I^b and I^c are the instantaneous currents in phases b and c respectively. The three equations for the stator windings can be written in matrix form:

$$\begin{bmatrix} V^a_{ph} \\ V^b_{ph} \\ V^c_{ph} \end{bmatrix} = \begin{bmatrix} L_{ph} & M_{ph} & M_{ph} \\ M_{ph} & L_{ph} & M_{ph} \\ M_{ph} & M_{ph} & L_{ph} \end{bmatrix} \begin{bmatrix} \frac{dI^a}{dt} \\ \frac{dI^b}{dt} \\ \frac{dI^c}{dt} \end{bmatrix} + \begin{bmatrix} I^a.R_{ph} + E^a_{ph} \\ I^b.R_{ph} + E^b_{ph} \\ I^c.R_{ph} + E^c_{ph} \end{bmatrix} \qquad 2.89$$

To accomodate the square wave phase voltages it is necessary to calculate the instantaneous phase currents at discrete points in time. The derivatives of the phase currents are then represented by

$$\frac{dI}{dt} \simeq \frac{I_n - I_o}{\Delta t} \qquad 2.90$$

where I_n is the value of the current at the end of the time-step Δt, and I_o is the value of the current at the start of the time-step. If the time interval between calculations is small, equation 2.90 is a reasonable approximation. Equation 2.88 can be modified for use in a time-step procedure. The current gradient calculated by equation 2.90 is valid at the mid-point of the time-step, and so the current used in the resistance term of equation 2.88 should be the average $(I_n + I_o)/2$. In addition V_{ph} and E_{ph}

should be the mid-step values. The modified form of equation 2.88 is then:

$$V^a_{ph}\Big|_{t+\frac{\Delta t}{2}} = L_{ph}\left[\frac{I^a_n - I^a_o}{\Delta t}\right] + M_{ph}\left[\frac{I^b_n - I^b_o}{\Delta t}\right] + M_{ph}\left[\frac{I^c_n - I^c_o}{\Delta t}\right] + \left[\frac{I^a_n + I^a_o}{2}\right].R_{ph} + E^a_{ph}\Big|_{t+\frac{\Delta t}{2}} \qquad 2.91$$

where the n and o suffices denote "new" and "old" respectively and "t" denotes the elapsed time. Similar equations can be written for b and c phases. The "new" and "old" terms in equation 2.91 can be separated:

$$V^a_{ph}\Big|_{t+\frac{\Delta t}{2}} - E^a_{ph}\Big|_{t+\frac{\Delta t}{2}} - \left(\frac{R_{ph}}{2}\, I^a_o\right) + \left[\frac{L_{ph}.I^a_o + M_{ph}.I^b_o + M_{ph}.I^c_o}{\Delta t}\right] = \left[\frac{L_{ph}}{\Delta t} + \frac{R_{ph}}{2}\right] I^a_n + \left[\frac{M_{ph}}{\Delta t}\, I^b_n\right] + \left[\frac{M_{ph}}{\Delta t}\, I^c_n\right] \qquad 2.92$$

The three phase equations can then be grouped in matrix form:

$$\begin{bmatrix} \left(\frac{L_{ph}}{\Delta t} + \frac{R_{ph}}{2}\right) & \frac{M_{ph}}{\Delta t} & \frac{M_{ph}}{\Delta t} \\ \frac{M_{ph}}{\Delta t} & \left(\frac{L_{ph}}{\Delta t} + \frac{R_{ph}}{2}\right) & \frac{M_{ph}}{\Delta t} \\ \frac{M_{ph}}{\Delta t} & \frac{M_{ph}}{\Delta t} & \left(\frac{L_{ph}}{\Delta t} + \frac{R_{ph}}{2}\right) \end{bmatrix} \begin{bmatrix} I^a_n \\ I^b_n \\ I^c_n \end{bmatrix} = \begin{bmatrix} V^a_{rem} \\ V^b_{rem} \\ V^c_{rem} \end{bmatrix} \qquad 2.93$$

where

$$V^a_{rem} = \left[V^a_{ph}\Big|_{t+\frac{\Delta t}{2}} - E^a_{ph}\Big|_{t+\frac{\Delta t}{2}} - \left(\frac{R_{ph}}{2}\, I^a_o\right) + \frac{L_{ph}.I^a_o + M_{ph}.I^b_o + M_{ph}.I^c_o}{\Delta t} \right] \qquad 2.94$$

$$V^b_{rem} = \left[V^b_{ph}\Big|_{t+\frac{\Delta t}{2}} - E^b_{ph}\Big|_{t+\frac{\Delta t}{2}} - \left(\frac{R_{ph}}{2} I^b_o\right) + \frac{L_{ph}\cdot I^b_o + M_{ph}\cdot I^a_o + M_{ph}\cdot I^c_o}{\Delta t} \right] \qquad 2.95$$

and

$$V^c_{rem} = \left[V^c_{ph}\Big|_{t+\frac{\Delta t}{2}} - E^c_{ph}\Big|_{t+\frac{\Delta t}{2}} - \left(\frac{R_{ph}}{2} I^c_o\right) + \frac{L_{ph}\cdot I^c_o + M_{ph}\cdot I^a_o + M_{ph}\cdot I^b_o}{\Delta t} \right] \qquad 2.96$$

The values of the applied phase voltages and generated emf's are entered into the equations, as appropriate, to simulate the inverter output changes, the motor rotation, and the actual load angle.

A program was written to solve the instantaneous phase currents using equations 2.93. There were several special features that the program required for successful operation. They were:

(i) During a phase freewheeling period, the appropriate voltage polarity must be applied to the phase until the current decays to zero.

(ii) Smaller time-steps are required during freewheel periods than at other times. This is because the freewheel phenomenon is a transient effect and the rates of change of current are consequently high.

(iii) During freewheeling periods all three phases carry current, but at other times only two phases are conducting. This is a direct result of the 120° - 60° applied voltage waveform. During freewheeling it is necessary for all three equations to be employed, but otherwise two equations are sufficient since one current is assumed to be zero.

The Fortran program was carefully debugged but unfortunately two problems were present in the results and no

means could be found to remove them. Firstly, when the mutual phase inductance M_{ph} was set such that:

$$M_{ph} = -0.5\ L_{ph} \qquad 2.91$$

large undamped oscillations were observed in the phase currents. Reducing the ratio of M_{ph} to L_{ph} helped but a value of $M_{ph} = -0.25\ L_{ph}$ (i.e. large leakage) was needed to effectively remove the trouble. However, to be able to analyse the magslip, the program must be stable with $M_{ph} = -0.5\ L_{ph}$, since the magslip has a mutual to self inductance ratio very close to this. It is believed that the problems occurring for the conditions set by equation 2.91 may be due to the matrix appearing singular for small values of Δt. All the variables (E_{ph}, V_{ph}, speed, δ, etcetera) were varied to try to determine the cause of the problem but no cause could be isolated. Secondly, it was observed that when a phase winding was switched on, the current in that phase often flowed in the wrong direction for some time. This problem may be partly due to the fact that the model does not include any blocking capability equivalent to that of a transistor switch. Hence if the inverter is to be modelled accurately, this blocking must be incorporated.

Much effort was expended on the program with little progress. To test the program logic, the discretely varying imposed phase voltages were replaced by sinusoidally varying voltages, and results consistent with those expected were produced. This would seem to indicate inherent instability in the chosen mathmatical model. The instability may be due to the assumption of zero damping. Therefore it is probably essential to incorporate at least one damper circuit into the model. It was decided not to spend further time on this area of work since a floating star point had still to be incorporated to make the program suitable for the magslip analysis. Several network analysis computer packages were suggested (e.g. SPICE) but none appeared to be really suitable. It was hoped to return to this work at the end of the project if time permitted.

2.9 Synchronous Motor or D.C. Motor?

Before concluding this chapter it is worth considering whether a synchronous machine employed in a voltage forced autopiloted system, operates as a synchronous machine or as a brushless d.c. motor, in which the armature and field winding positions are reversed for convenience.

If the machine were to operate as a brushless d.c. motor, this would imply that the conventional copper/carbon brush commutator had been replaced by an electronic unit performing the same function. The commutator on a d.c. motor armature arranges for the supply current to enter and leave the correct coils, and the spacial position of the current around the armature winding is fixed. The inductance of the armature does attempt to cause the armature currents to lag the applied voltages, as the sparking on a commutator demonstrates, but the fact that the armature coils are connected in series means that "current forcing" is applied to any coil as it comes out from under the brushes.

In contrast, the situation when an inverter is used to voltage force a synchronous motor is completely different. Since the inverter can only fix the applied voltages, the resulting currents move about spacially depending on the frequency and winding parameters. If the motor is current forced, it does then have a current pattern fixed in space and so is similar to a d.c. motor with the exception that the speed is not then directly related to the applied input current, whereas in a d.c. motor the speed is proportional to the applied voltage. To achieve a true brushless d.c. motor system, using a synchronous motor as the prime mover, requires a control system with an inner loop to current force the motor, and an outer loop to control the amplitude of the current such that the speed of the motor is then proportional to a control signal.

An autopiloted synchronous motor fed from a voltage

source inverter is therefore exactly what its name describes, and it should not be thought of as a brushless d.c. motor. In the simplest terms, however, one can think of the synchronous motor phase winding as having a back-emf generated by a field flux, the phase angle δ between the back-emf and the applied phase voltage being set by the autopiloting "commutator" circuitry.

2.10 Conclusions

The analysis in this chapter gives an insight into the typical torque/speed characteristics that might be expected from synchronous machines operated in an autopiloted mode. It is apparent that in a voltage forced system the load angle can determine the characteristic of the motor. If the motor is to operate over a wide speed range and deliver the maximum possible torque at all times, it is normally essential to have some form of load angle control. The load angle should ideally be zero at low speeds and increase towards 90° as the speed rises, the optimum load angle being given by arctan (X_{ph}/R_{ph}) for a cylindrical rotor motor. Rougher load angle control is likely to be adequate in practice, (as found by Chalmers (2.9)), but some sacrifice in the torque per volt performance is then incurred. Possible load angles in a practical system should be 0° for starting and 45° for running, or perhaps even just a permanently fixed angle of 30°. Further work is desirable to investigate how load angle control can optimise various motor parameters such as the power factor, the efficiency, and the mean output power per input peak phase current x peak rail volts (a measure of converter switch utilisation).

Whilst many autopiloted systems use quasi-square wave voltage-source inverters, it is nevertheless valuable to use sinusoidal analysis to predict the probable performance characteristics. The advantages of the sinusoidal equations are that they are simple to understand and easy to use.

CHAPTER 3

A 3 PHASE SINUSOIDAL SYNCHRONOUS MOTOR DRIVEN BY A 3 PHASE QUASI-SQUARE WAVE VOLTAGE SOURCE INVERTER

3.1 Introduction

This chapter describes the experimental work that was carried out on a small autopiloted synchronous motor system, in which a 3 phase quasi-square wave voltage source inverter was used to drive a 3 phase machine of the magslip type, possessing sinusoidally distributed windings. The operation of an autopiloted synchronous motor is outlined in section 3.2. Methods of sensing rotor position by digital means are discussed in section 3.3 and section 3.4 goes on to cover the ways in which the load angle can be adjusted during motor operation. The circuits used for the inverter and autopiloting electronics are described in section 3.5. The experimentally measured performance curves are presented in section 3.6 and are compared with the sinusoidal predictions obtained using the equations derived in Chapter 2. The results indicate that load angle adjustment in an autopiloted voltage-forced synchronous motor is a useful facility. The final part of the chapter presents electronic circuits which enable easy and precise adjustable load angle control to be achieved.

3.2 The Operation of a Sinusoidal Motor from a Quasi-Square Variable Frequency Voltage Source

When a multiphase sinusoidally distributed synchronous motor is driven from a sinusoidal voltage supply, the airgap

flux generated by the stator windings travels around the airgap at a constant rate which is directly proportional to the supply frequency. However, if the stator windings are driven from a quasi-square voltage supply, the resultant airgap flux moves around the airgap in a series of discrete moves. In an autopiloted synchronous motor system the instants at which the flux moves from one location to the next are determined by a rotor position sensor, which directly controls the stator voltage drive waveforms produced by an inverter.

A common quasi-square voltage waveform used in 3 phase systems is the 120° - 60° type which is shown in fig. 3.2(a). (Note all angles in this chapter are in electrical degrees unless otherwise stated.) The 120° - 60° voltage waveforms can be readily generated by an inverter of the form shown in fig. 3.1.

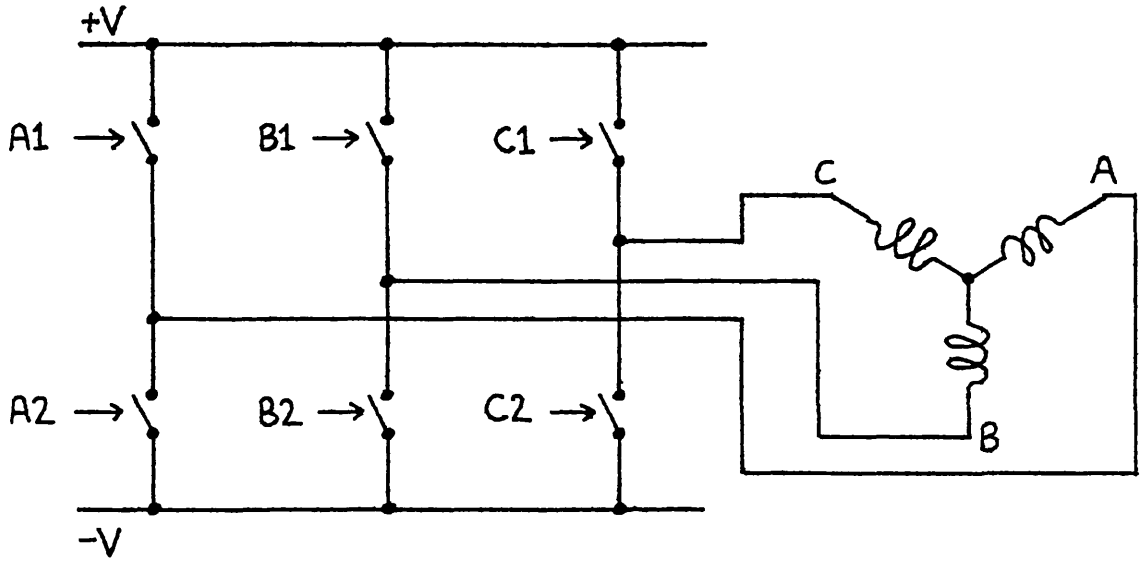


FIG. 3.1. A THREE PHASE VOLTAGE SOURCE INVERTER FORMED BY THREE HALF BRIDGES.

The inverter consists of three "half-bridges", with each half-bridge being made up from a pair of semiconductor switches. Each half-bridge controls the voltage applied to one phase winding terminal. Two control signals are required

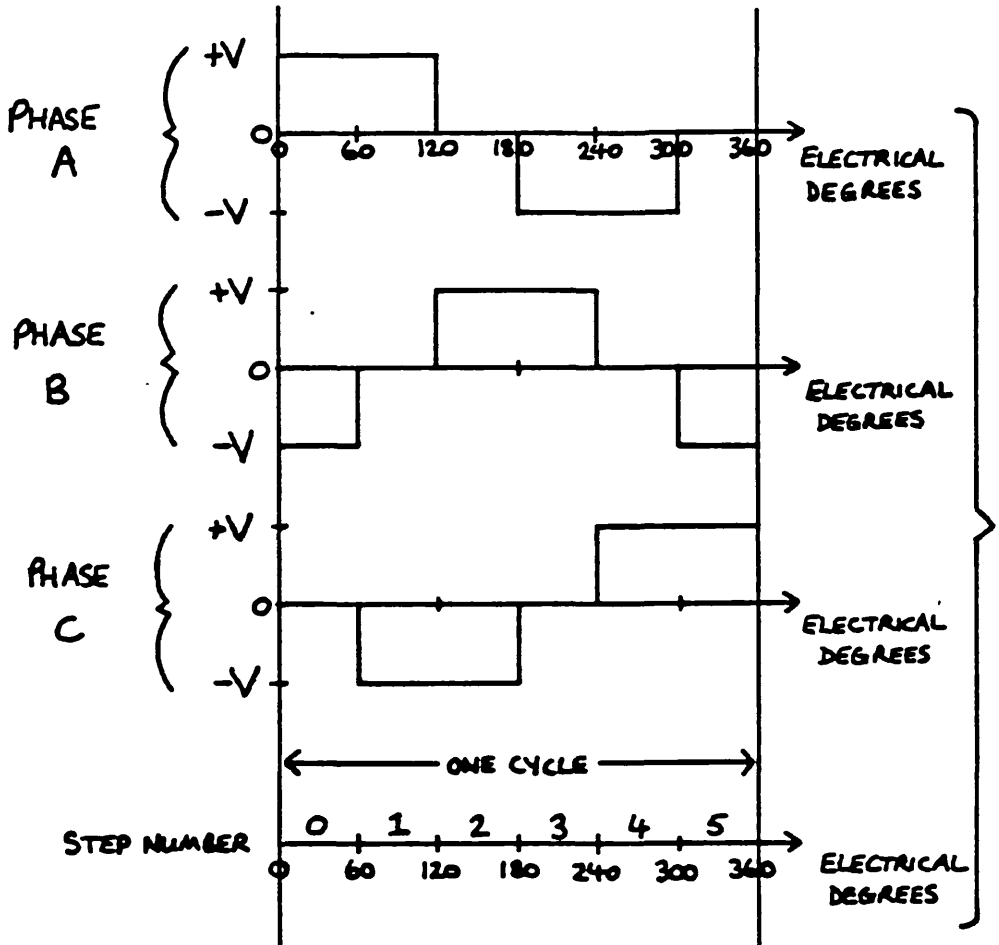


Fig.3.2a. Three Phase Quasi-Square Voltage Waveforms ($120^{\circ} - 60^{\circ}$).

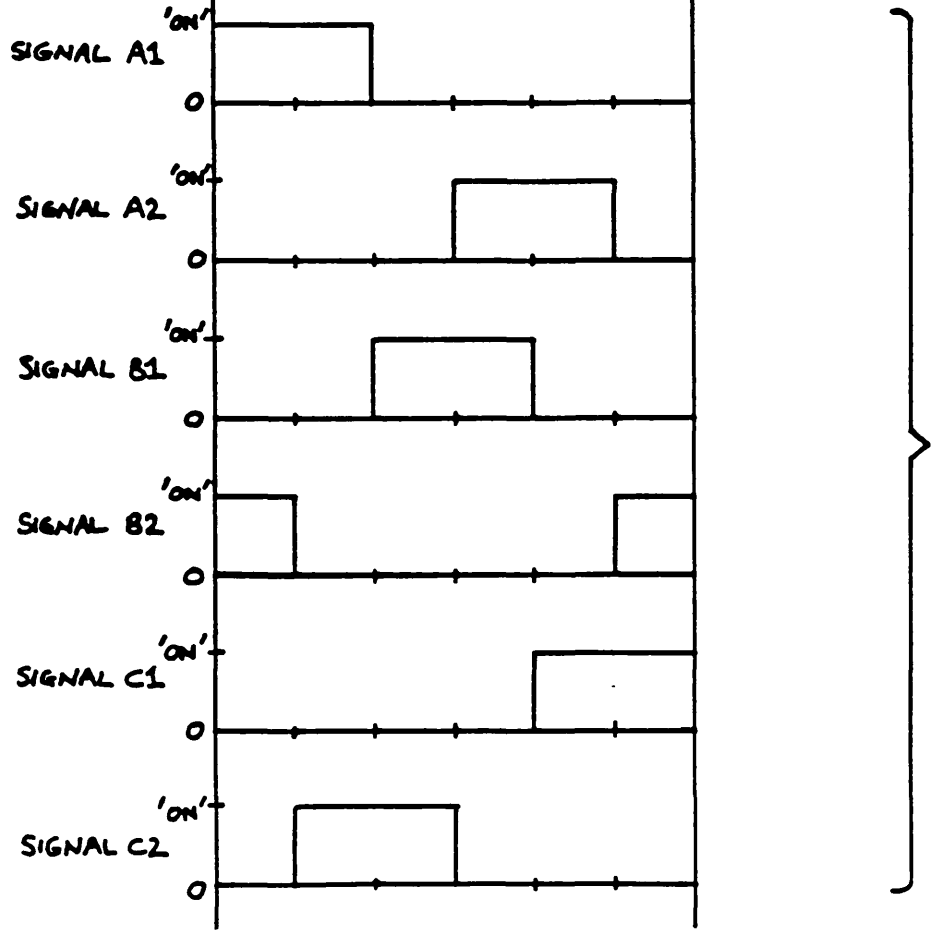


Fig.3.2b. The Inverter Control Signals Needed To Generate The Waveforms In Fig.3.2a.

per half-bridge. (A1 and A2 for phase A, B1 and B2 for phase B, etcetera.) The inverter control signals are shown in fig. 3.2(b).

The 120° - 60° voltage waveforms do not contain third harmonics and so a reduction in motor losses can be achieved by their use. The current that flows in a phase is neither constant nor sinusoidal because the applied voltage and the back-emf in the winding do not have similiar waveshapes. A possible current waveform resulting from such mismatch is indicated in fig. 3.3.

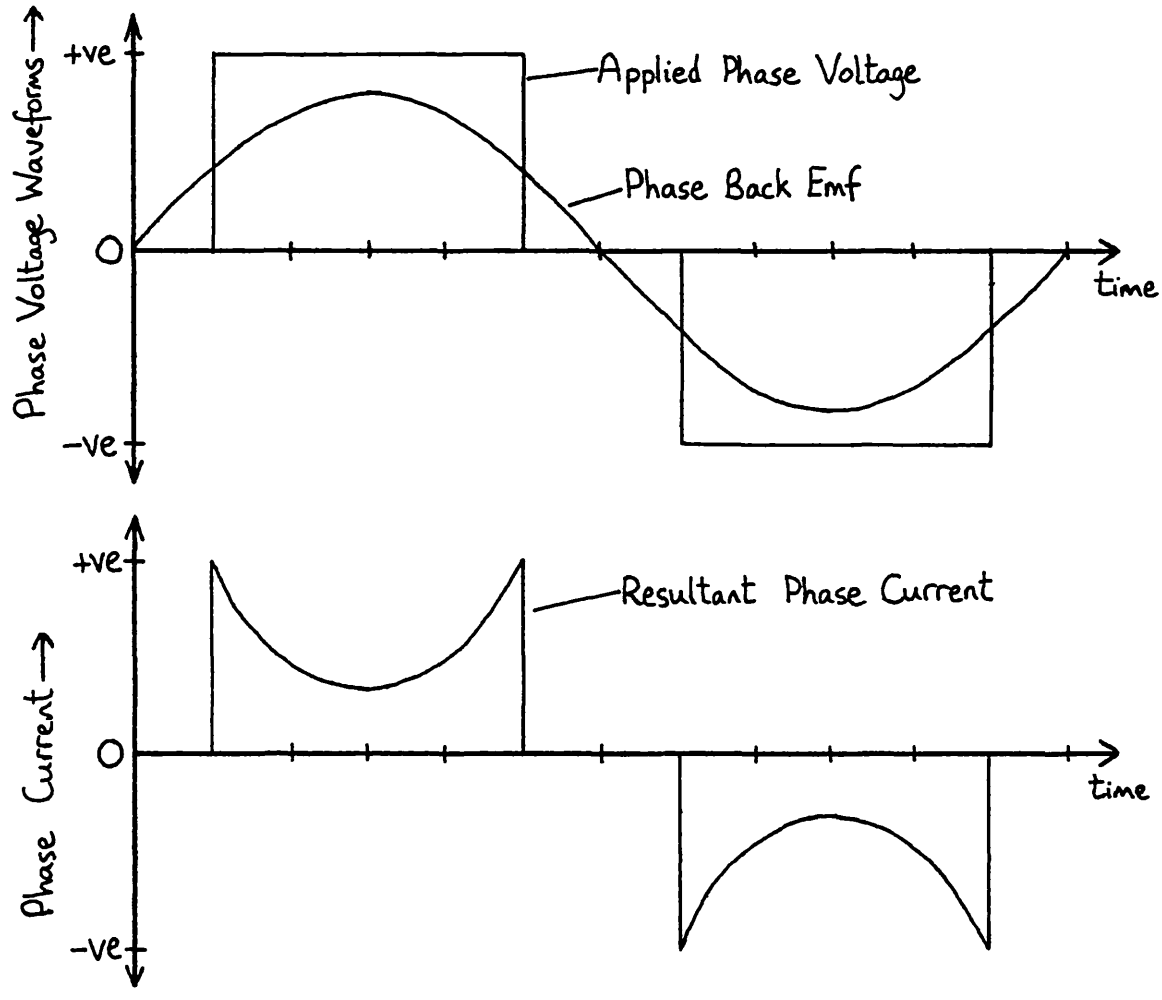


FIG. 3.3 TYPICAL PHASE CURRENT IN A SINUSOIDAL SYNCHRONOUS MOTOR WHEN DRIVEN BY QUASI SQUARE PHASE VOLTAGES.

The example shown is for a low frequency and so winding inductance has a negligible effect on the current. Current does not flow whenever the applied phase voltage is zero, even though some back-emf is present, because the simple form of inverter shown in fig. 3.1 does not provide a current path at these times.

When 3 phase 120° - 60° waveforms are used to drive a star connected motor, there are two phases energised at any instant. Since the windings are in series, the current depends on the sum of the back-emfs within both phase windings. The stator flux ϕ_s at any time is the resultant of the fluxes produced by the two energised phase windings. During each cycle there are six combinations of voltages across the stator phase windings as indicated in fig. 3.2(a). Therefore, in a two pole machine ϕ_s rotates by 360° for every six steps. The alignment of ϕ_s at each step is shown with respect to the phase winding axes in fig. 3.4; (stator fluxes ϕ_s^0 to ϕ_s^5).

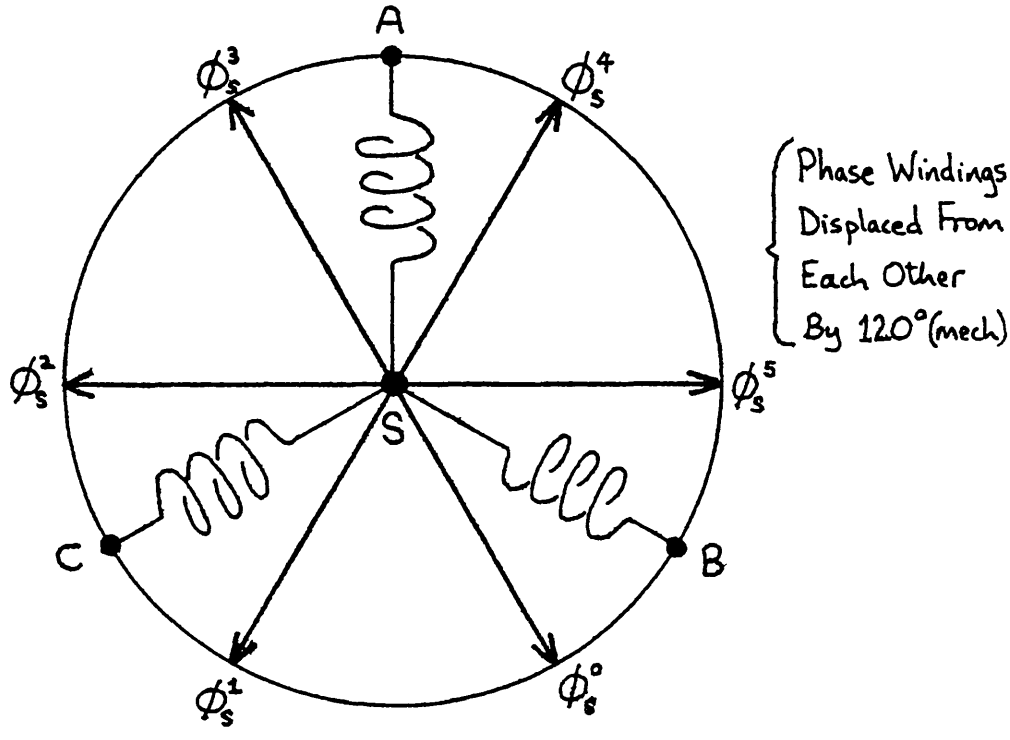


FIG. 3.4. THE RELATIONSHIP BETWEEN THE RESULTANT STATOR FLUXES AND THE PHASE WINDING AXES.

The diagram assumes that when positive volts are applied to a phase winding terminal, the phase current and resulting phase flux act along the axis of the winding in a direction from the phase terminal to the star point.

With reference to fig. 3.4 it can be more convenient to think of the machine as having three "resultant" phase windings arranged along the axes of the resultant fluxes, with each winding energised by an appropriate line to line voltage waveform. Torque is produced by the motor whenever the axis of the rotor flux ϕ_f is not aligned with the axis of the stator flux ϕ_s. Continuous rotation of the rotor can be achieved by arranging the rotor position sensor so that ϕ_s always steps on ahead of ϕ_f as the rotor moves towards the alignment axis. The torque T produced by the motor is basically a function of the stator flux ϕ_s, the rotor flux ϕ_f, and the angle ψ between the axes of ϕ_s and ϕ_f.

i.e. $$T = f(\phi_s, \phi_f, \psi) \qquad 3.1$$

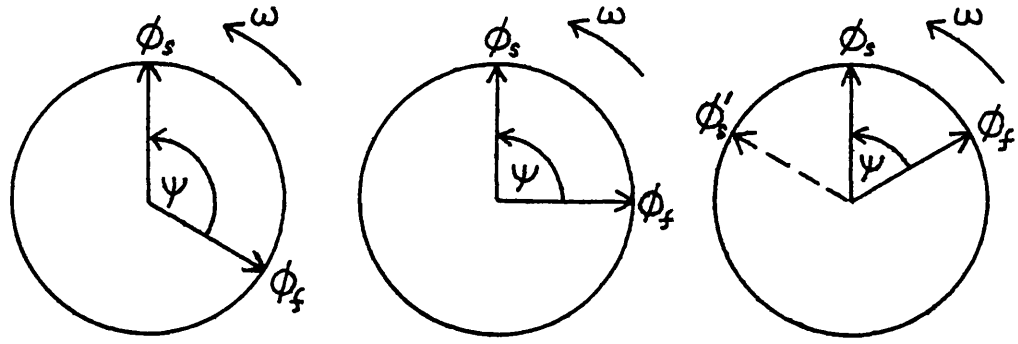
If the fluxes are sinusoidally distributed the equation becomes:

$$T = K.\phi_s.\phi_f.\sin\psi \qquad 3.2$$

where K is an arbitrary constant, and the angle ψ is termed the torque angle. For the purposes of this discussion it is convenient to use equation 3.2 even though in general a quasi-square voltage fed machine may not have sinusoidally distributed fluxes. It can be seen in equation 3.2 that the maximum rotor torque for a given stator current is achieved when the torque angle is 90^o. Unfortunately it is not possible to maintain a torque angle of 90^o throughout a step because ϕ_s only exists along six fixed axes whereas ϕ_f continuously rotates. However, it is possible to arrange the switching points of ϕ_s so that an average torque angle of 90^o is achieved.

The stator flux ϕ_s, damping and saturation effects

apart, is directly proportional to the phase currents. Therefore, if the phase currents flow in phase with the applied phase voltages, the stator flux ϕ_s can be precisely controlled by the phase voltages. This enables the average torque angle to be held at a value of 90° and so the torque per applied volt is maximised. The voltages must be switched onto the appropriate phase windings when the axis of ϕ_f is 120° behind the axis of ϕ_s, and they then remain connected until the axes of ϕ_f and ϕ_s are 60° apart. The effective torque angle over the step is then 90°. The situation during the step is shown in figures 3.5(a),(b) and (c).



(a) Start of Step $\psi = 120^\circ$ (b) Middle of Step $\psi = 90^\circ$ (c) End of Step $\psi = 60^\circ$

FIG 3.5. EFFECTIVE TORQUE ANGLE OF 90° OVER A STEP ACHIEVED BY APPROPRIATE CONTROL OF ϕ_s

At the end of the step the phase voltages are switched so that ϕ_s steps on by a further 60° (position ϕ_s' shown dotted on fig. 3.5(c)).

A phase current will only flow in phase with the applied phase voltage if the phase winding impedance is resistive. This is only true either in a motor which has negligible stator inductance or at low switching frequencies when any inductive reactance is small. Therefore, in a typical motor operating at a reasonable speed, the inductive reactance of the phase windings does result in a significant

time constant for the currents. If the phase voltages are applied when $\psi = 120^{\circ}$ there is a finite delay before the phase currents and ϕ_s reach their maximum values. Similarly, when the phase voltages are removed, the phase currents and ϕ_s do not disappear immediately but decay away exponentially via the inverter freewheel paths. The nett effect of this is a reduction in the motor torque predicted by equation 3.2 for two reasons. Firstly, the delay in ϕ_s rising and decaying means that the average value of ψ over the step is less than 90°. Secondly, the average value of ϕ_s is reduced over the step period because the phase currents do not attain their maximum values immediately.

To limit these problems it is necessary to apply the phase voltages earlier than the $\psi = 120^{\circ}$ position and also remove the voltages before the $\psi = 60^{\circ}$ position. By advancing the voltages in time, it is possible to make the phase currents and hence ϕ_s occur over approximately the required period and with the correct average spacial displacement from the axis of ϕ_f. The advance needed is similar to the brush shift required for good commutation in d.c. commutator machines.

The amount by which the phase voltages must be advanced depends on the speed of the motor. As mentioned earlier in this section, the machine can be considered as having three "resultant" phase windings arranged along the resultant stator flux axes. If the fundamental components of the applied line to line voltages are then considered, the step nature of the process can be neglected and a phasor diagram of the variables can be drawn as shown arbitrarily in fig. 3.6(a). The stator and rotor fluxes are included for comparison in fig. 3.6(b). The symbols used on fig. 3.6(a) and 3.6(b) refer to a "resultant" phase winding and they represent the following:

V - the applied line to line voltage.

E - the "resultant" phase winding back-emf voltage due

to the rotor flux ϕ_f.

I - the "resultant" phase winding current.

Z - the "resultant" phase winding impedance.

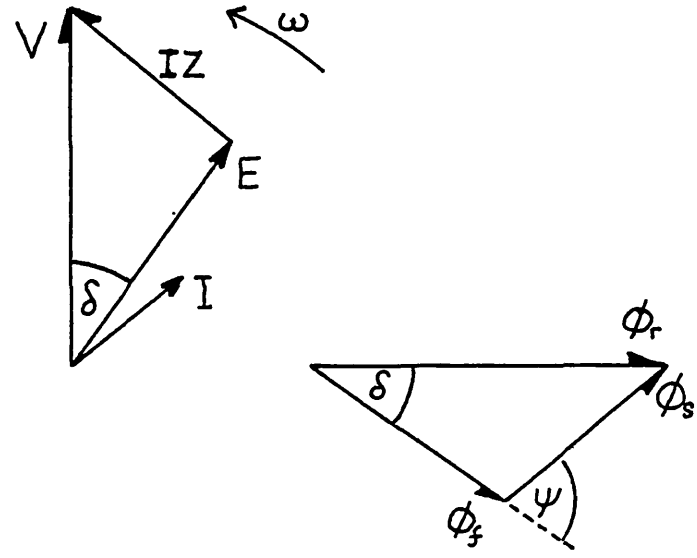
δ - the load angle between V and E.

ϕ_f - the rotor field flux.

ϕ_s - the stator flux due to the winding.

ϕ_r - the total airgap flux due to ϕ_f and ϕ_s.

ω - the angular frequency in the winding.



(a) Phasor Diagram (b) Flux Diagram

FIG 3.6 TYPICAL PHASOR AND FLUX DIAGRAMS FOR A SYNCHRONOUS MOTOR

The position of ϕ_f is equivalent to the average position it has over a step. The process of advancing V with respect to ϕ_f in order to maximise the torque per volt is equivalent to increasing the load angle δ. Therefore, the sinusoidal theory discussed in Chapter 2 has some relevance to the situation of a sinusoidal motor driven by quasi-square waves of voltage.

In practice the control of the load angle to maximise the motor torque over a speed range is generally easier than

the control of the torque angle. This is because the position of E_{ph} is directly related to the rotor direct axis (the axis of ϕ_f) and it is the position of this axis that is monitored by the rotor position sensor. Therefore, the information needed for load angle control is readily available. In contrast, torque angle control requires a knowledge of the relative positions of both ϕ_f and ϕ_s: in addition to the need for a rotor position sensor to monitor the axis of ϕ_f, it is also necessary to monitor the phase currents in order to control the axis of ϕ_s.

It should be noted from the magslip data sheet contained in Appendix 2A that the magslip possesses appreciable phase inductance and so requires load angle control in order to operate effectively over a wide speed range when used in an autopiloted motor system.

Having outlined the method of continuous torque production in an autopiloted synchronous machine it is not proposed to cover the subject in any greater detail. Numerous papers have been published with descriptions of the operation of such brushless d.c. motor systems. Notable papers are those by Fink (3.1, 3.2), Yates (3.3), and Woodbury (3.4, 3.5). However, it is appropriate to stress at this point that the control of speed in an autopiloted synchronous motor system is not achieved by setting the frequency of the inverter. The inverter frequency is solely controlled by the rotor position information. Speed control is typically achieved by a combination of voltage and load angle adjustments.

3.2.1 Methods of Achieving Reverse Motor Rotation

The various position sensors discussed in section 3.3 of this chapter can all be arranged to provide the appropriate sequence of inverter control signals for either direction of motor rotation. However, sequencing an inverter through its output states, in the reverse order to that followed for forward rotation, is not sufficient to cause

an autopiloted synchronous motor to turn in the reverse direction. This is because the position sensor is deliberately adjusted to keep the axis of the field flux ϕ_f displaced from the axis of the stator flux ϕ_s in such a way that continuous unidirectional torque is produced. To achieve reverse motor rotation it is necessary to reverse the motor torque. The torque can be reversed if the relative positions of the stator and rotor fluxes are swapped. Figs. 3.7(a) and (b) illustrate the method. The stator flux ϕ_s is assumed to be firmly aligned by the motor windings and the field flux ϕ_f is free to move. The two fluxes, ϕ_s and ϕ_f, attempt to align themselves coaxially.

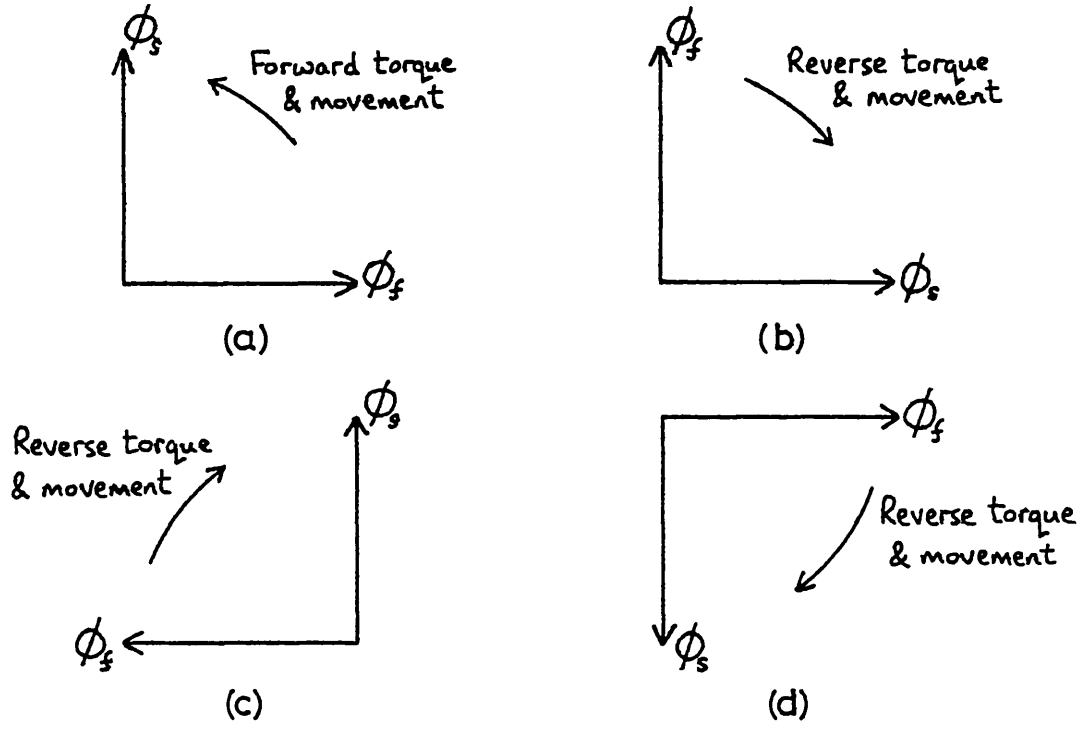


FIG. 3.7. THE RELATIVE POSITIONS OF STATOR AND ROTOR FLUX DETERMINE THE DIRECTION OF ROTATION IN A SYNCHRONOUS MOTOR

Hence in fig. 3.7(a) the rotor field flux tries to swing round anticlockwise, whereas in fig. 3.7(b) the direction of rotation is clockwise. The change in the positions of ϕ_s and ϕ_f can be achieved in two ways:

(a) by reversal of the rotor field flux direction.

Fig. 3.7(c) shows the effect of reversing the phasor ϕ_f shown in fig. 3.7(a);

(b) by reversal of the stator flux direction. Fig. 3.7(d) shows the effect of reversing the phasor ϕ_s shown in fig. 3.7(a).

The reversal of field flux is simple on a wound rotor synchronous motor since it is only necessary to reverse the field current. However this is not possible on a permanent magnet rotor machine. The rotor field flux can be "reversed" on such machines by mechanically rotating the rotor with respect to the position sensor unit by one pole pitch. For example, a two pole rotor would be rotated by 180 mechanical degrees. However, such a solution is inconvenient in practical systems. Method (b) is preferable in such circumstances since the reversing can be achieved purely by electronic means. To reverse the stator flux it is necessary to advance the flux through one pole pitch. This can be achieved by advancing the inverter through half a cycle. Therefore, if the inverter produces 'm' output voltage combinations per cycle it must be moved on by m/2 combinations to reverse ϕ_s. A three phase inverter producing six steps per cycle would be advanced through its sequence by three steps. The same effect can be achieved by moving the inverter back through its sequence by m/2 steps. Since the inverter control signals are derived from the position sensor unit, it is possible to process the signals in an electronic manner so as to advance the inverter through m/2 steps. Hence no mechanical adjustments are required for reversal. This method can also be used on wound field motors. It is advantageous on such motors since it removes the need to break the inductive field current. In addition, it is not necessary to employ a switch capable of reversing the field current.

It should be noted that for motor reversal by either of the above simple schemes, the position sensor should be adjusted so that the inverter driving the motor sets up a

0^{o} load angle in either direction. If this is not so, the motor will run distinctly better in one direction than in the other. This is because any load angle displacement is positive for one direction of motor rotation but is negative for the other. For example, if the position sensor was set so that the load angle was 30^{o} in one direction, it would be -30^{o} in the opposite direction. If electronic load angle adjustment is available (as outlined in section 3.4 of this chapter) the restriction in the position sensor setting does not necessarily apply, although it may still be more convenient to operate with a position sensor that provides basic 0^{o} load angle information. The methods of electronic load angle adjustment discussed in section 3.4 are, with the exception of the time delay method, based on a position sensor providing 0^{o} load angle information suitable for either direction of rotation. Other load angles are derived from that basic information.

Both field flux and stator flux reversal methods were implemented on the magslip. Further details are given later in this chapter.

3.3 Digital Position Sensors for the Detection of Rotor Position

The angular resolution θ_p of a digital position sensor is fixed by the number of steps, b, that it can distinguish in one revolution.

i.e. $$\theta_p = (360/b) \text{ degrees} \qquad 3.3$$

The 3 phase voltage waveforms shown in fig. 3.2(a) have six distinct combinations of equal duration per cycle. The rotor position sensor must accurately determine the six switching points as the rotor turns. A 2 pole machine such as the magslip, rotates once every cycle and so the switching information from the position sensor has to occur at

regular intervals that are 60 mechanical degrees apart. For machines with a higher pole number the position sensor must produce the switching information at smaller angular intervals. The commonly used quasi-square voltage waveforms are such that the number of voltage combinations 'd' during one cycle of a Q phase voltage supply is given by:

$$d = 2.Q \qquad 3.4$$

If the Q phase voltage supply drives a synchronous motor with P poles, the number of voltage combinations per revolution, C, is given by:

$$C = d \times P/2 \qquad 3.5$$

i.e.
$$C = Q.P \qquad 3.6$$

C is actually the number of inverter "steps" per revolution. Therefore, assuming the inverter steps are of equal duration, the maximum angular interval α_s at which the position sensor must produce switching information is given by:

$$\alpha_s = (360/C) \text{ degrees} \qquad 3.7$$

i.e.
$$\alpha_s = \left(\frac{360}{Q.P}\right) \text{ degrees} \qquad 3.8$$

The assumption of equally long inverter steps is valid providing the machine phase voltages are either:

(a) switched positive and negative alternately for 180^o each.

or (b) Switched positive and negative for periods of $[180(Q-1)/Q]$ degrees each with zero volt periods of $(180/Q)$ degrees inbetween. (This is the type of waveform used for the magslip; i.e. Q = 3.)

If the zero volt periods lie between 0^o and $(180/Q)$ degrees

there is a need for more frequent switching information, and hence a switching sensor with a resolution better than α_s degrees is necessary. However, in the work reported in this thesis the waveforms employed were of type (b). Thus a position sensor with a resolution of α_s degrees is sufficient for correct machine operation.

To ensure that the position sensor resolution θ_p given by equation 3.3 is sufficient to meet the required resolution α_s given by equation 3.8, it is necessary to impose the conditions:

$$\theta_p \leqslant \alpha_s \qquad 3.9$$

and $$n.\theta_p = \alpha_s \qquad 3.10$$

where n is a positive integer greater than zero and is given by:

$$n = \frac{\text{number of position sensor steps per revolution}}{\text{number of inverter steps per revolution}} \qquad 3.11$$

i.e $$n = b/C \qquad 3.12$$

The condition imposed by equation 3.10 ensures that if a step signal from the position sensor is made to coincide with an inverter switching point, the succeeding "coincidence points" will occur after every n signals of the position sensor.

Substituting equations 3.3 and 3.8 into equation 3.10 gives an equation for b:

$$b = n.Q.P \qquad 3.13$$

Therefore, the resolution of a position sensor given by equation 3.3 becomes:

$$\theta_p = \left(\frac{360}{n.Q.P}\right) \text{ degrees} \qquad 3.14$$

The minimum number of steps, b_{min}, that must be detected by a position sensor for a given motor system to function correctly is obtained from equation 3.13 when n = 1:

$$b_{min} = Q.P \qquad 3.15$$

3.3.1 Available Types of Position Sensors

The positions of the inverter switching points can be detected by several techniques. The most reliable position sensor techniques do not involve contact between stationary and moving parts; i.e. they are "brushless". Practical contactless position sensor techniques include:

(a) optical sensors: light sources shine through a patterned disc attached to the rotor shaft and photo diodes detect the presence or absence of light. Both the light source and photo diodes are stationary.

Advantages of the optical sensor method include the fact that the signals from the photo diodes rise and fall quite abruptly and so the switching points are well defined. Also, the signals are d.c. and so do not require rectification or filtering. However, the method does suffer from the disadvantage that the light sources are liable to wear out and fail suddenly. In addition, the photo diode signal is usually very small and so may require amplification before it can be used for control purposes.

(b) Reluctance sensors: a toothed magnetic wheel mounted on the rotor shaft rotates between stationary poles which carry sense windings. A high frequency a.c. flux is directed around the magnetic circuit, and the signal amplitude induced into each sense winding depends on the reluctance of the airgap between the toothed wheel and the relevant stationary pole. Since the airgap reluctance varies with the rotor position it is possible to determine the rotor orientation from the induced signal amplitudes.

Advantages of reluctance sensors include the fact that the method has no inherent wear out mechanism. However, this is offset by the fact that the signal must be rectified and filtered before it is in a suitable form to use. Also there is always some induced voltage in the sense windings due to leakage flux and the signal builds up gradually rather than abruptly. Hence a trigger circuit is required to detect the correct switching point.

(c) Magnetic field detectors: the position of a magnet attached to the rotor can be sensed by either hall effect generators or magneto-sensitive resistors. In suitable situations the rotor magnetic field can be directly sensed, thus eliminating the need for an extra rotor mounted magnet.

The magnetic field detector systems do provide d.c. signals but in general suffer from the same disadvantages as reluctance sensors.

(d) Capacitance sensors: the position of a suitably shaped wheel mounted on the rotor shaft can be detected by capacitance measuring probes.

Capacitance sensors are compact and hence can be easily incorporated within motor frames, but considerable amplification is needed to boost the sensor output signal to a usable level.

(e) Back-emf sensors: it is possible to detect the rotor position by monitoring the back-emf waveforms generated in the phase windings by the rotor flux.

However, there is often a great deal of spurious noise on the signals due to the switching action on the phase windings and this can make accurate position detection impossible. In addition, the rotor position cannot be determined at very low speeds since the back-emf tends to zero.

All of the position detection systems discussed above are used in practical systems, and either (a), (b), or (c) were thought to be suitable for use in the magslip drive system. However, an optical system is relatively easy to construct and the switching pattern on the rotating disc is easy to modify. Therefore, it was decided to use an optical position sensor system for both the magslip system described in this chapter, and the square wave motor described in Chapter 6. Optical systems can be made with extremely good resolutions, especially by using Moiré interference fringe techniques. Fortunately, such techniques are not usually needed to obtain the resolution required by autopiloted motor systems. All the disc patterns described in this thesis are very simple.

3.3.2 Possible Arrangements of the Sensors

There are a variety of ways of arranging the chosen sensors in order to generate the required inverter control signals. The different arrangements affect:

(a) the number of sensors required;

(b) the amount of decoding logic required;

(c) the ease by which load angle control can be achieved.

The papers by Binns et al (3.6), Rieke (3.7), and Le-Huy et al (3.8) give examples of typical optical position sensor arrangements.

Three sensor arrangements were implemented on the magslip system. They can be categorised by the type of logic needed to convert the position sensor signals into the inverter control signals. The first form of sensor arrangement does not require any logic to process the position sensor signals, and so it is referred to here as the NOLOG

(NO LOGIC) method. The second form of sensor arrangement uses combinational logic to process the position sensor signals and so it is referred to here as the COMLOG (COMbinational LOGIC) method. The third form of sensor arrangement is based on sequential logic to process the position sensor signals and so is referred to here as the SELOG (SEquential LOGIC) method. The methods are discussed below.

3.3.2.1 NOLOG (No Logic) Position Sensor Systems

A NOLOG position sensor produces the necessary inverter control signals directly. Some amplification and pulse shaping circuitry may be necessary in order to match the signals to the requirements of the inverter drive circuitry, but the individual sensor signals are not combined with each other in any way. A NOLOG position sensor arrangement is shown in fig. 3.8. The disc pattern and sensor positions produce the necessary control signals for a three phase inverter of the type shown in fig. 3.1, and the resulting inverter voltage waveforms are of the form shown in fig. 3.2(a).

The principal advantages of the NOLOG system are that it is simple and it produces the inverter control signals directly. In addition, the system does not require any initialisation and so it always produces the correct position information even when it is first switched on. Therefore, the motor is always synchronised. This feature means that the motor can always produce continuous torque when trying to start and consequently starting should not be a problem with a NOLOG arrangement. Also, the sequence of the position sensor signals is directly related to the direction of rotation, and this means that it is impossible for the motor to fall out of synchronism. A disadvantage of the NOLOG arrangement is that a relatively large number of sensors are required to detect the required switching points during each revolution. (Six sensors are required for the arrangement shown in fig. 3.8.)

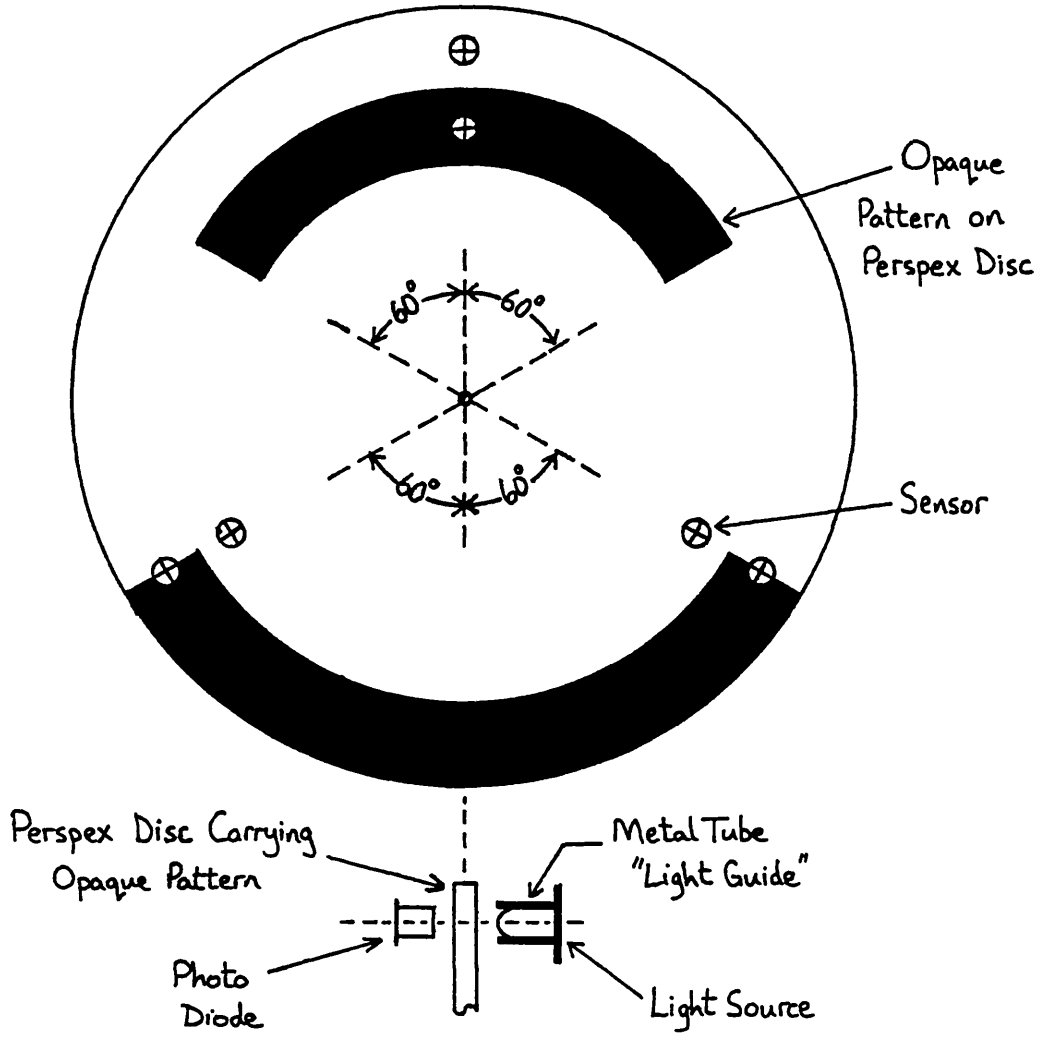


FIG 3.8. A SIMPLE "NOLOG" POSITION SENSOR ARRANGEMENT

The large number of sensors and light sources leads to a high initial cost and results in an appreciable power consumption. In addition, the reliability of the system is not likely to be very high because of the large number of position sensor components. A further disadvantage is that a potentially damaging situation for the inverter can arise in the event of a light source failure. This is because the signals generated by the arrangement shown in fig. 3.8 are active when the opaque pattern blocks the light path, and this is equivalent to a light source failure. Hence it is possible for both of the control signals to a phase of the

inverter to be active, causing a direct short across the power supply voltage rails. This problem can be overcome either by reversing the opaque and transparent portions of the pattern (i.e. light detected then results in an active control signal), or by the use of simple logic to detect if both control signals to any phase are active.

Reversal of a NOLOG controlled synchronous motor can be achieved by simply reversing the field flux direction. The alternative reversing technique in which the stator flux is reversed, relies on modification of the inverter control signals to advance the inverter through its sequence. Such signal processing requires logic of some form and hence cannot be achieved by a basic NOLOG system. However, stator flux reversal is actually quite simple in practice. For example, the six step voltage waveform shown in fig. 3.2(a) can be advanced through half a cycle by swapping over the control signals going to each phase of the inverter. (Signal A1 then controls inverter switch A2 and vice versa.) Such "logic processing" can be achieved by simple change over switches. Therefore, stator flux control to achieve motor reversal is possible by using NOLOG position sensor signals and so permanent magnet motors can be reversed. However, the signal processing needed between the position sensor and inverter means that such a system is not then strictly of a NOLOG form.

In conclusion it can be said that whilst the NOLOG arrangement is potentially simple, it also has sufficient drawbacks to prevent it from being an automatic choice for the sensor arrangement.

3.3.2.2 COMLOG (Combinational Logic) Position Sensor Systems

A COMLOG position sensor arrangement produces a binary coded output. The value of the binary code is determined by the position of the rotor. The binary code is not usually

directly equivalent to the required inverter control signals and so some combinational logic is needed to convert the binary code into the necessary form. The method can most easily be explained by means of an example. It is instructive to assume that the six inverter output voltage combinations shown in fig. 3.2(a) are related to the binary position code as shown in fig. 3.9. The associated inverter control signals are also shown in fig. 3.9.

Step Nº ↓	Binary Coded Sensor Output			Inverter Control Signals						Inverter Phase Voltage Outputs		
	D_2	D_1	D_0	A_1	B_1	C_1	A_2	B_2	C_2	Phase A	Phase B	Phase C
0	0	0	0	1	0	0	0	1	0	+V	-V	0
1	0	0	1	1	0	0	0	0	1	+V	0	-V
2	0	1	0	0	1	0	0	0	1	0	+V	-V
3	0	1	1	0	1	0	1	0	0	-V	+V	0
4	1	0	0	0	0	1	1	0	0	-V	0	+V
5	1	0	1	0	0	1	0	1	0	0	-V	+V

FIG. 3.9. EXAMPLE OF HOW THE SWITCHING STATES OF A THREE PHASE INVERTER CAN BE RELATED TO THE BINARY CODED ROTOR POSITION OF A MOTOR

It can be seen in fig. 3.9 that the binary coded sensor output is not directly suitable for use as the inverter control signals. Therefore, a conversion is needed and this can be performed by combinational logic. The necessary logic can be implemented by standard discrete logic gates or alternatively by a "look-up" table stored on a Read Only Memory (ROM). In a look-up table solution each distinct binary code is used as an address to select a memory location in the ROM. The selected location contains the inverter control signals relevant to the binary input code. Therefore, there is no problem in the conversion of binary coded position information into inverter control signals.

However, the example binary code (D_2 D_1 D_0) shown in fig. 3.9 is not satisfactory in practice because it is NOT a "unit-distance" code. This means that more than one binary digit changes at some, or all, of the transitions between successive code values. Unfortunately in practical systems it is not possible for two or more digits to change state at precisely the same instant. This is due to the combined effects of the mechanical alignment tolerances and the slightly different response times of the optical position sensors generating the binary digits. The result is that a spurious code can be generated during the transition, and this could cause a damaging condition for the inverter. This possibility can be eliminated by the use of a unit-distance binary code. The reflected-binary, or "Gray code", is a unit-distance binary code, and it is commonly used for position sensors. The Gray code must be cyclic so that at the end of a revolution the code changes back to its original value as illustrated by the six step Gray code (G_2 G_1 G_0) shown in fig. 3.10(a).

Step Number	Position Sensor Binary Grey Code Output		
	G_2	G_1	G_0
0	0	0	0
1	0	0	1
2	0	1	1
3	0	1	0
4	1	1	0
5	1	0	0

(a)

Alternative Grey Code Output		
G_2	G_1	G_0
0	0	0
0	0	1
0	1	1
1	1	1
1	1	0
1	0	0

(b)

FIG. 3.10 TWO POSSIBLE SIX STEP GRAY CODES

The Gray code shown in fig. 3.10(a) can be generated by a disc pattern and sensor arrangement of the form shown in fig. 3.11.

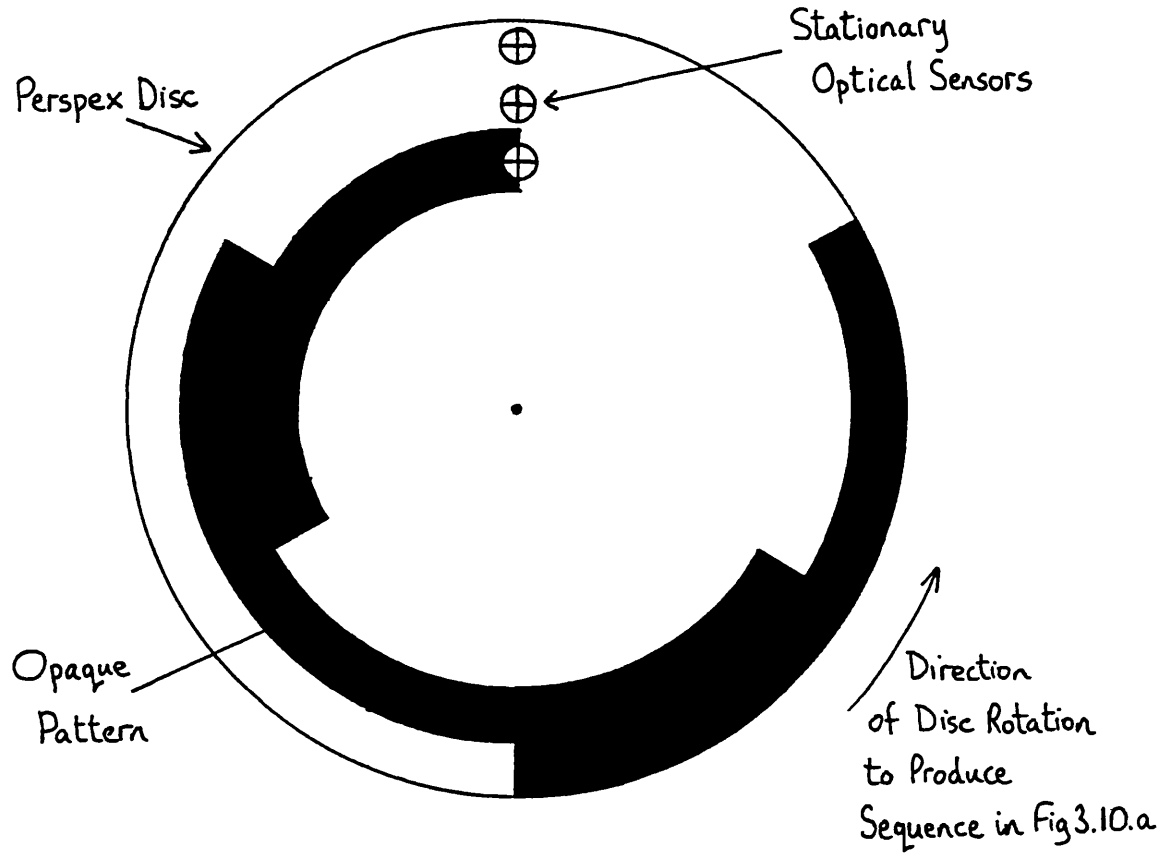


FIG. 3.11. SUITABLE DISC PATTERN AND SENSOR ARRANGEMENT TO PRODUCE THE GRAY CODE SHOWN IN FIG. 3.10.a.

It is possible to generate Gray codes for any even cycle length. Commercial Gray code position sensors can provide up to 4096 coded values per revolution, but such high resolution is not necessary to autopilot a synchronous motor. It is interesting to note that for a given number of binary digits there can be more than one set of codes for a particular cycle length. For example, the three bit, six step, Gray code shown in fig. 3.10(b) is different from the one shown in fig. 3.10(a). It is not proposed to go into the methods of designing Gray codes of particular cycle lengths. Details can be found in many texts, including that by Taub and Schilling (3.9).

It has previously been stated in this sub-section that the binary position information is not necessarily of the correct form to use directly as the inverter control signals. It therefore follows that since Gray code is simply a re-

arrangement of natural binary code, it is also likely to require some processing by combinational logic in order to generate the required inverter control signals.

COMLOG systems have two useful features that are also possessed by NOLOG systems. In the same manner as a NOLOG system, a COMLOG system is always synchronised, even at switch on. In addition, the sequence of the position codes indicates the direction of the rotor. Therefore, in these two respects, the NOLOG and COMLOG systems are identical.

A COMLOG system can be used with either field flux reversal or stator flux reversal to reverse the direction of rotation of an autopiloted motor. Control of the inverter to achieve stator flux reversal merely requires additional combinational logic to process the binary position code. Therefore, permanent magnet rotor machines, autopiloted by COMLOG position sensors, can be readily reversed.

A COMLOG system employing a Gray code plus some combinational logic is not necessarily as simple as a NOLOG arrangement, but it may require fewer light sources and sensors which can result in improved reliability. For example, the six step Gray code discussed in this sub-section requires only three sources and sensors as compared with the six required by the NOLOG example given in sub-section 3.3.2.1. However, the combinational logic required by the COMLOG arrangement may be undesirable or too expensive in a commercial application. It is virtually impossible to state in general terms when a COMLOG arrangement should be used in preference to a NOLOG arrangement and vice versa. It is true to say that the choice depends very much on the requirements of each particular autopiloted system.

3.3.2.3 Basic Pulse Counting "Single Edge" Sensing SELOG (Sequential Logic) Position Sensor Systems

The previously described NOLOG and COMLOG position

sensor arrangements both produce coded outputs that provide full position information at all times. An alternative method of determining the angular position of a shaft is to use a SELOG position sensor arrangement. A SELOG system provides a pulse signal each time the shaft rotates through a known number of degrees. The pulses divide each revolution into a number of segments and so henceforth they are referred to as SEG' pulses. Since the pulses occur at regular angular intervals it is possible to logically compute the position of the shaft by counting the number of pulses that are detected. The counting can be conveniently performed by sequential logic. In order that the accumulated count has some meaning in terms of the shaft position, it is necessary to have a "home" signal to initialise the count. This signal zeroes the counting mechanism when the shaft is in a known position, and because it synchronises the count value to the shaft position it is henceforth called the SYNC' pulse. The count recorded by the sequential logic follows a fixed sequence during each revolution in the same way as a NOLOG or COMLOG position sensor provides a repeating sequence of code values. For example, a SELOG sensor capable of detecting six steps per revolution could produce a sequence cycling through binary 0 to binary 5.

To maintain the synchronous motor in synchronism it is necessary to relate the position count to the state of the inverter phase outputs by suitable logic. The sequential logic counter can be arranged to output the "count" value in a variety of codes. The various codes are basically of two types. The first type of counter code is of a binary form with the binary number changing on successive SEG' pulses. In general the binary code is not directly compatible with the required inverter control signals. Therefore some combinational logic, similar to that used in COMLOG position sensors, is needed to produce the inverter control signals from the binary count information. It is possible to use standard numerical binary code if a synchronous counter is employed. (All the outputs of a synchronous counter

change simultaneously and so the spurious outputs that can arise with a COMLOG position sensor when using standard binary code do not occur.) Therefore it is possible to implement a SELOG counter using standard TTL or CMOS integrated circuit synchronous binary counters. A block diagram of a "binary code" SELOG position sensor is shown in fig. 3.12.

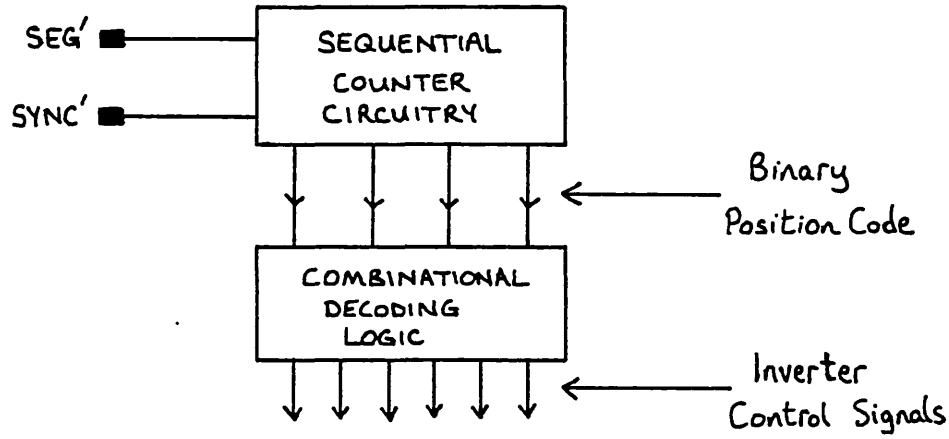


FIG. 3.12 BASIC FORM OF A "SELOG" POSITION SENSOR.

The second type of counter code is directly compatible with the inverter control signals. Therefore, the sequential logic counter does not count in the numerical sense, but it steps the outputs through a series of combinations that are actually the required inverter control signal sequence. However, the amount of logic needed to achieve a non-binary count sequence can be significant. Hence, although the method provides the inverter control signals directly, it is not necessarily simpler than the first method. Therefore, the second type of code and the associated counting circuitry is not discussed further in this thesis. The practical SELOG position sensor circuit described later in this chapter uses standard binary code to indicate the position count.

The number of equal sized steps per revolution that a basic pulse counting SELOG position sensor can distinguish is equal to the number of SEG' pulses generated per revolution. It is shown earlier (in section 3.3) that the number of steps 'b' per revolution that a position sensor must detect, in order to successfully autopilot a Q phase inverter driving a P pole synchronous motor, can be calculated by using the equation:

$$b = n.Q.P \qquad (3.13)$$

The equation can be used to calculate the number of SEG' pulses required per revolution in a basic pulse counting SELOG system:

$$SEG' = n.Q.P \qquad 3.16$$

The simplest form of SELOG system would require only one sensor to detect a combined SYNC' and SEG' pulse stream of the form shown in fig. 3.13.

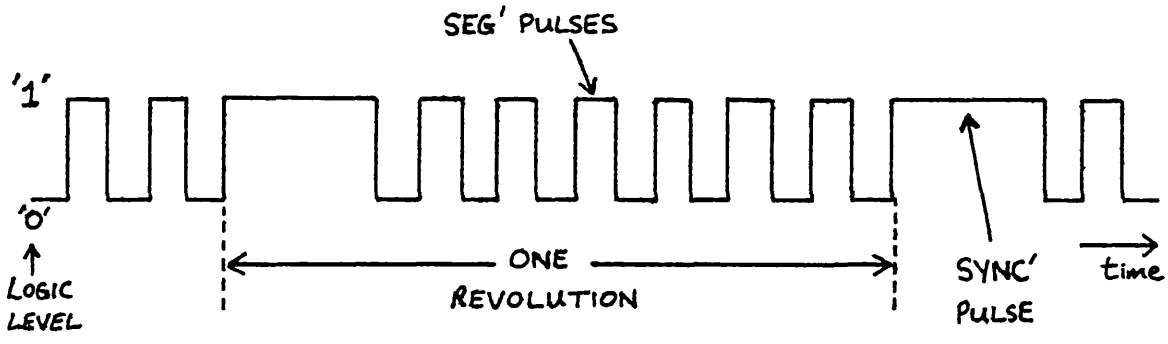


FIG. 3.13 A POSSIBLE COMBINED SYNC' AND SEG' PULSE STREAM FOR A "SELOG" POSITION SENSOR SYSTEM.

Unfortunately it is not easy for a logic system to reliably distinguish between the SYNC' and SEG' pulses when they are combined in the same pulse train, given the tremendous frequency range involved as the motor accelerates from rest to full speed. It is much easier to detect the SYNC' and

SEG' signals with independent sensors. Typical SYNC' and SEG' pulse waveforms for a SELOG position sensor are shown in fig. 3.14(a). This example has six SEG' pulses per revolution and so it could control a six step three phase inverter.

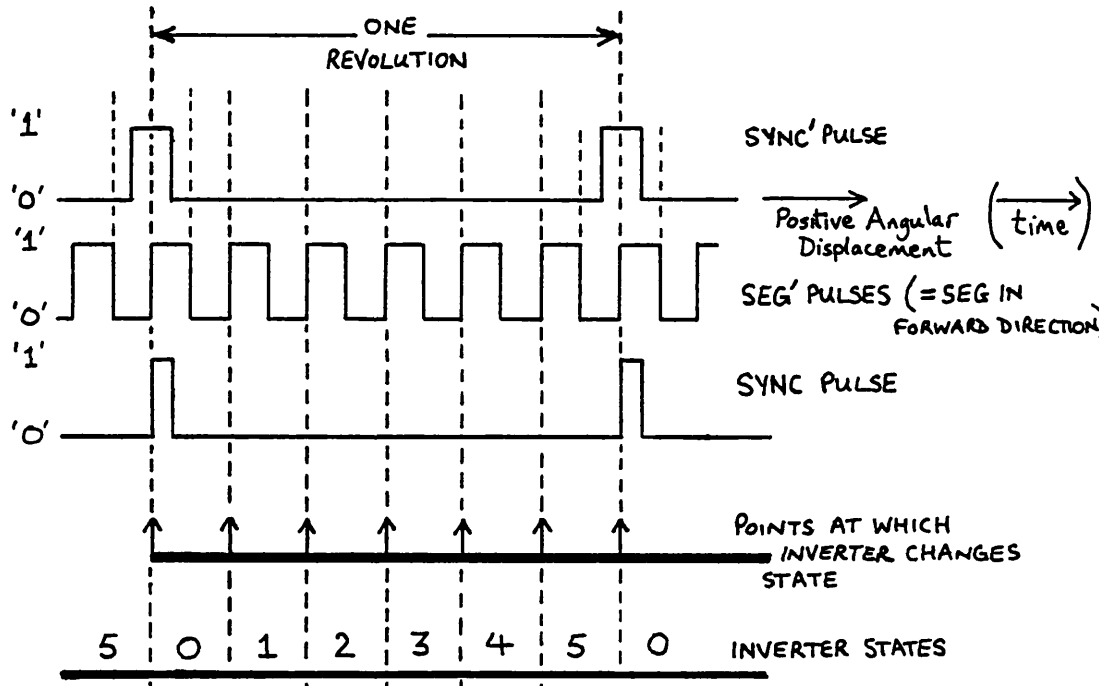


FIG 3.14(a) SELOG WAVEFORMS IN FORWARD DIRECTION.

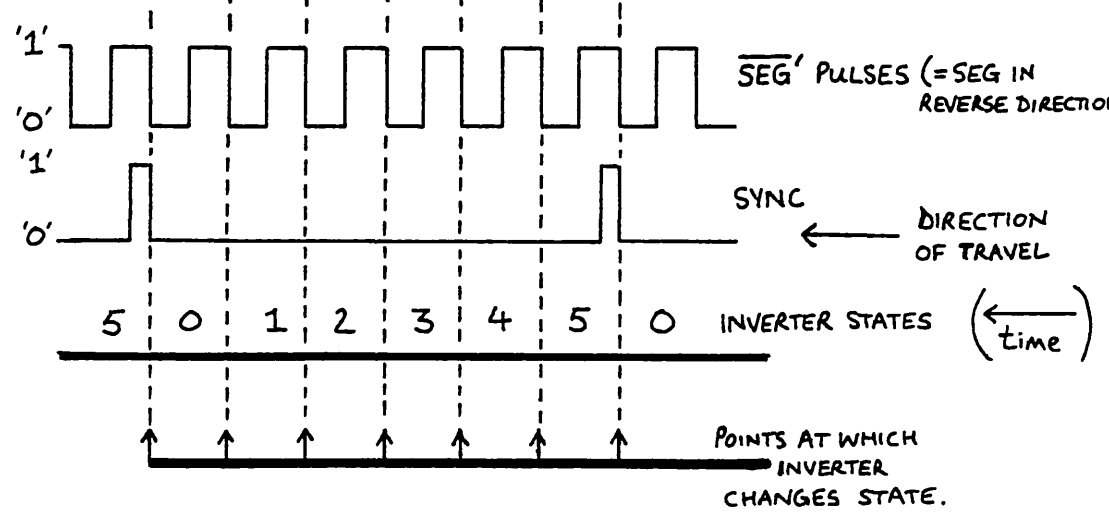


FIG 3.14(b) SELOG WAVEFORMS IN REVERSE DIRECTION.

The alignment of the position sensor with respect to the rotor, is adjusted so that the points at which the inverter outputs must change to maintain the required load angle, coincide with the rising edges of the SEG' pulses as shown in fig. 3.14(a). (The system could equally well be aligned on the falling edges.) Since a basic pulse counting SELOG system only senses one edge of each SEG' pulse, it can be described more accurately as a single-edge-triggering SELOG system. It is useful to base the position sensing on the detection of edges for two reasons. Firstly, the edges can be reliably sensed and so the inverter can be made to switch at exactly the same points during successive revolutions. Secondly, many sequential logic circuits, including binary counters, change states on the rising or falling transitions of the input signal and so they can be easily incorporated into an edge sensing SELOG system.

The rising edges of the SEG' pulses only coincide with the inverter switching points for one direction of shaft rotation. If the shaft rotates in the opposite direction it is necessary to detect the falling edges of the SEG' pulses in order to ensure that the inverter switches at exactly the same angular points. This is because a transition that is sensed as a rising edge in one direction is sensed as a falling edge in the opposite direction. The detection of falling edges to allow motor operation in a reverse direction is most easily achieved by inverting the SEG' pulses. If this is done the system only has to sense rising edges in either direction of rotation. Obviously when the system operates in the reverse direction, it is necessary for the sequential logic to decrement the position count on successive $\overline{\text{SEG}}$' pulses so that the inverter steps backwards through its sequence.

Correct bidirectional operation of a SELOG sensor arrangement is also dependent on the position of the SYNC' pulse with respect to the SEG' pulses. The purpose of the SYNC' pulse is to indicate precisely the rotor position

once every revolution and so initiate a new sequence of inverter states. Therefore, in a forward direction the SYNC' pulse causes the inverter to enter state 0. Subsequent SEG' pulses increment the inverter through states 1,2,3,4, and 5. In a reverse direction the SYNC' pulse causes the inverter to enter state 5, and subsequent $\overline{\text{SEG}}$' pulses decrement the inverter through steps 4,3,2,1, and 0. The SYNC' pulse can provide the necessary position information in either direction if it is placed symmetrically about the boundary separating inverter state 0 from inverter state 5, as shown in fig. 3.14(a). If the SYNC' pulse is logically ANDED either with the SEG' pulses in the forward direction, or with the $\overline{\text{SEG}}$' pulses in the reverse direction, the resultant output SYNC is correct for either direction of rotation. The simple combinational logic that is required to process the SEG' and SYNC' signals so that forward and reverse rotation is possible is shown in fig. 3.15. The forward/reverse command input (F/R) determines whether the SEG' signals are inverted. In the circuit shown in fig. 3.15 the SEG' signals are inverted when input F/R is at logic 1. The output segment signals are called SEG pulses to distinguish them from the unprocessed input SEG' pulses produced by the position sensor disc. Equation 3.15 is valid for the calculation of the number of SEG pulses required in a given application. Replacing SEG' by SEG yields:

$$\text{SEG} = n.Q.P \qquad 3.17$$

The $\overline{\text{SEG}}$' and SYNC waveforms for reverse rotation (F/R = logic 1) are shown in fig. 3.14(b). It should be noted that in order to show the correct angular relationship between the signals in figures 3.14(a) and 3.14(b), it has been necessary to draw fig. 3.14(b) "back to front". Therefore, it should be read from right to left in order to see the correct rising edges in the various pulse waveforms; i.e. time goes from right to left.

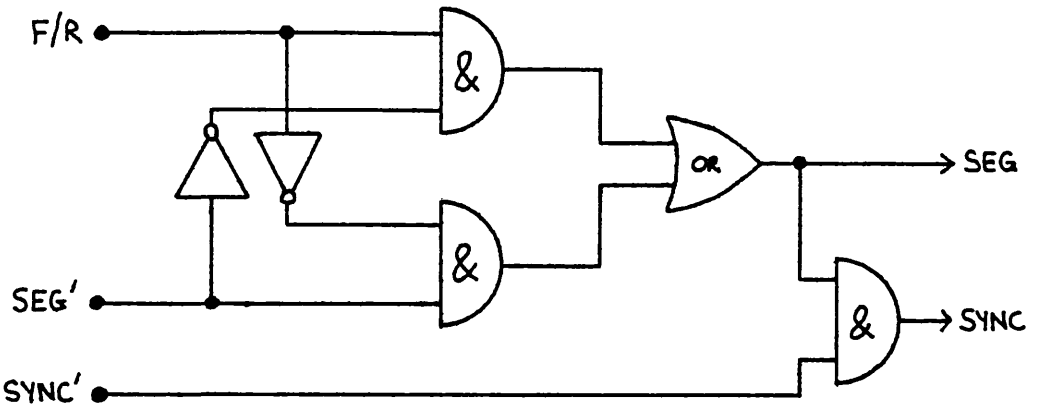


FIG. 3.15 CIRCUIT TO PROCESS BOTH SYNC' AND SEG' TO ENABLE FORWARD AND REVERSE MOTOR ROTATION

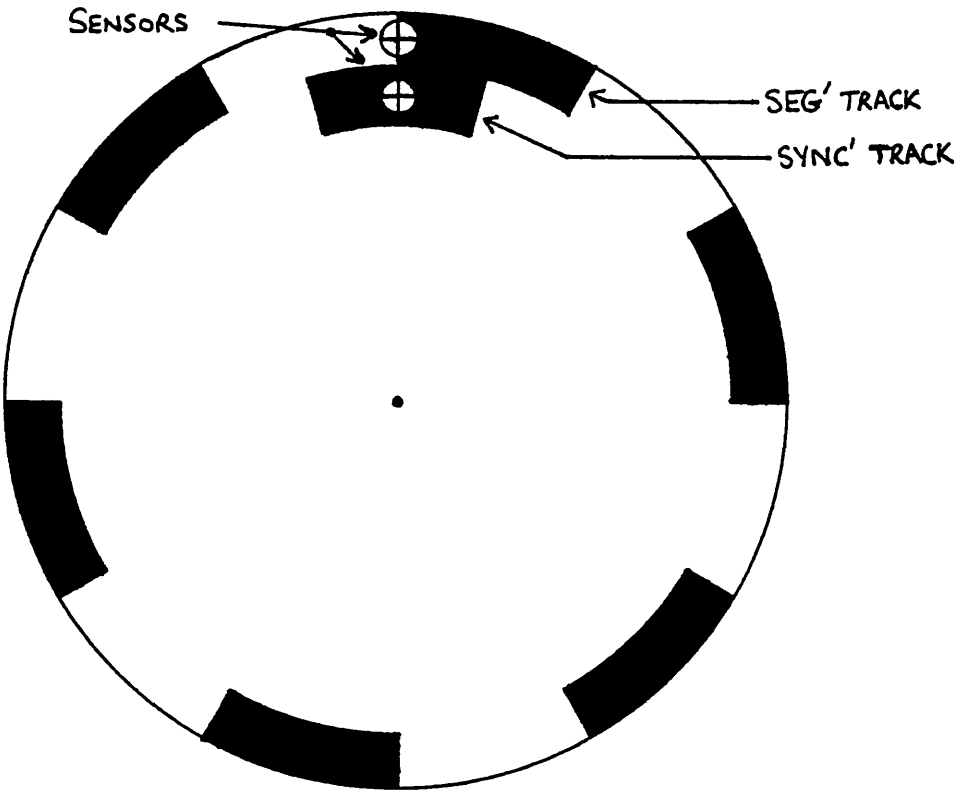


FIG 3.16. DISC PATTERN AND SENSOR ARRANGEMENT CAPABLE OF PRODUCING THE SEG' AND SYNC' SIGNALS SHOWN IN FIG 3.14(a).

The SEG' and SYNC' waveforms illustrated in fig. 3.14(a) can be generated by a disc pattern and sensor arrangement of the form shown in fig. 3.16.

There are two problems associated with SELOG type position sensors which could make them unsuitable for use in some applications. The first problem is that the motor system cannot be synchronised until a SYNC' pulse is detected. Therefore, in order to start a synchronous motor that is controlled by a SELOG position sensor, it is necessary to slowly step the inverter through its sequence until a SYNC' pulse is detected. This method relies on the synchronous motor "following" the inverter steps and because it is an open-loop process it can fail. The second problem is that a SELOG system cannot detect the direction of shaft rotation from the SEG pulses. This is because a SEG pulse merely indicates that the shaft is passing through a segment boundary. The absence of directional information from the SEG pulses should not normally be a problem. However, because the counter (and hence the inverter) steps on through its sequence on each SEG pulse, it is possible for the inverter and motor to get unsynchronised. For example, this could occur if the load on the motor shaft sets up a torsional vibration such that the position sensor oscillates several times over each segment boundary as the rotor turns. Fortunately this should not occur in a correctly designed system. The system is more likely to become unsynchronised if the motor shaft is forcibly halted and reversed by the load. When this happens the rotor turns one way whilst the SEG pulses cause the inverter to step the stator flux round in the opposite direction. However, it is possible to detect whether the rotor is travelling in the opposite direction to that expected by examining the SYNC' and SEG pulses together.

When the motor is rotating in the required direction the SEG, SYNC', and SYNC signals are as shown in fig. 3.17(a). If the motor is rotating in the opposite direction to that

required, the SEG, SYNC', and SYNC signals are as shown in fig. 3.17(b).

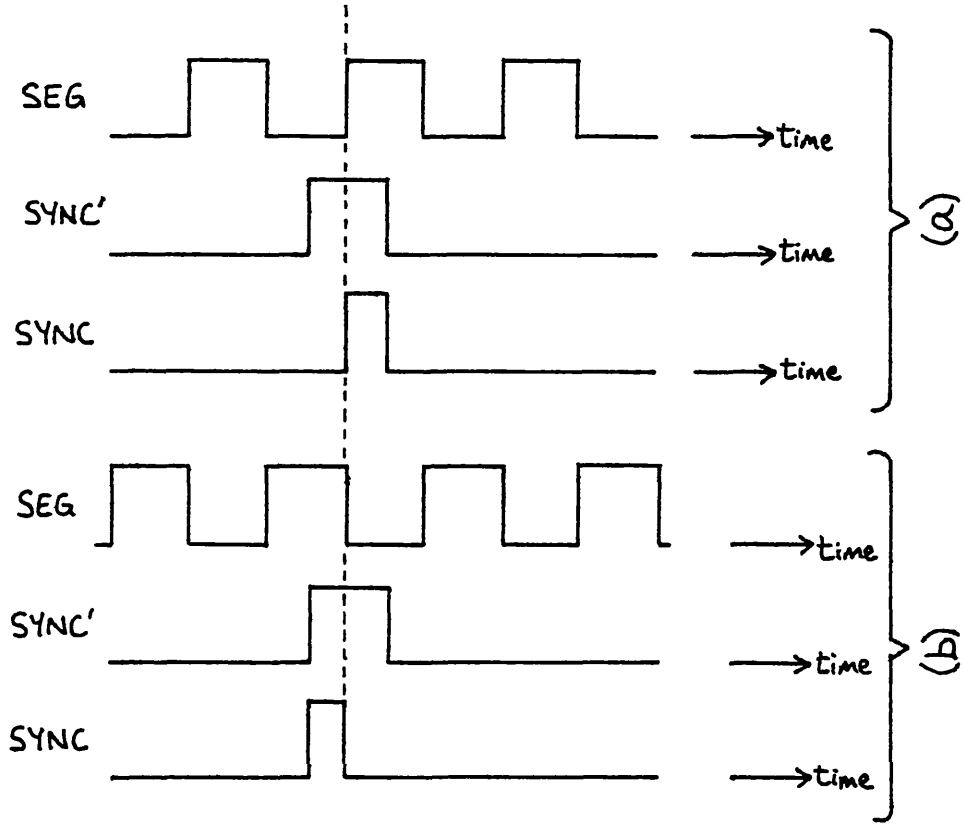


Fig. 3.17. SELOG SIGNALS WITH MOTOR ROTATING: (a) IN CORRECT DIRECTION, (b) IN WRONG DIRECTION.

Fig. 3.17(a) shows that when the motor is rotating in the correct direction, the SEG signal is at logic 0 when the SYNC' signal goes from logic 0 to logic 1. Fig. 3.17(b) shows that when the motor is rotating in the wrong direction, the SEG signal is at logic 1 when the SYNC' signal goes from logic 0 to logic 1. Therefore, the incorrect direction of rotation can be detected by feeding the SEG and SYNC' signals to a D-type flip flop as shown in fig. 3.18. The Q output goes to logic 1 if the motor rotates in the wrong direction for any reason. This system can only check the direction once per revolution, and because it relies on a SYNC' pulse, it is possible for the shaft to rotate in the wrong direction for almost a whole revolution, before it is detected.

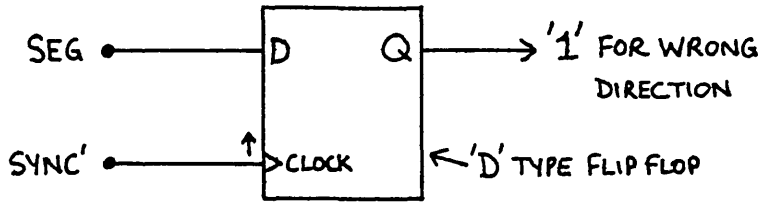


FIG.3.18. CIRCUIT TO DETECT "MOTOR ROTATION IN THE WRONG DIRECTION" USING SELOG SIGNALS.

The two problems discussed above are peculiar to a SELOG type of position sensor. Both problems could be minimised or eliminated by increasing the number of sensors and the complexity of the disc pattern. However, in the majority of applications a simple two sensor SELOG position sensor is adequate. A SELOG position sensor is attractive for commercial applications because it only requires a very simple disc pattern and two optical sensors. The need for some sequential logic to count the SEG pulses is not a major disadvantage because such logic is cheap, easy to implement, and reliable.

The reversal of a SELOG controlled synchronous motor can be achieved by either field flux reversal or stator flux reversal. The combinational logic that relates the SELOG position count to the inverter control signals requires some expansion to allow stator flux reversal. The main point to be noted about motor reversal with a SELOG system is that the sequence in which the sequential logic circuit counts must be reversed. Thus if the counter counts "up" during forward motor operation it must count "down" for reverse motor operation. This ensures that the inverter sequences in the correct order in both directions. The direction of counting can be controlled by the forward/reverse command (F/R) logic line that is also required to control the SEG' pulse inverter circuitry. The need for SELOG systems to be "instructed" which way the position sensor disc is supposed to be rotating in, so that correct position information is produced, is a major difference

between SELOG sensors on the one hand and NOLOG and COMLOG sensors on the other. UP/DOWN sequential counters suitable for SELOG position sensors are readily available in TTL and CMOS.

The resolution θ_{pse} of a "single-edge" sensing SELOG position sensor can be calculated by using equation 3.3. Rewriting the equation in terms of the number of SEG' pulses per revolution gives:

$$\theta_{pse} = (360/SEG') \text{ degrees} \qquad 3.18$$

or in terms of the "processed" segment pulses:

$$\theta_{pse} = (360/SEG) \text{ degrees} \qquad 3.19$$

The minimum number of SEG' pulses per revolution, $SEG'_{min(se)}$ that are needed for successful motor operation in a "single-edge" sensing SELOG system is given by equation 3.16 with n = 1:

$$SEG'_{min(se)} = Q.P \qquad 3.20$$

i.e. $SEG'_{min(se)}$ = Number of Phases x Number of Poles.

The "processed" segment pulses, SEG, are directly derived from the input SEG' pulses. Hence the minimum number of SEG pulses per revolution, $SEG_{min(se)}$, is simply given by:

$$SEG_{min(se)} = SEG'_{min(se)} \qquad 3.21$$

The resolution $\theta_{pse(min)}$ of a "single-edge" sensing SELOG position sensor with the minimum number of SEG' pulses is given by equation 3.18:

$$\theta_{pse(min)} = (360/SEG'_{min(se)}) \text{ degrees} \qquad 3.22$$

or $\theta_{pse(min)} = (360/Q.P)$ degrees 3.23

3.3.2.4 Improved-Resolution SELOG Position Sensor System

The resolution θ_{pse} given by equation 3.18 is not actually the best resolution that can be obtained from a SELOG position sensor. This is because the SELOG system described in section 3.3.2.3 is based on the detection of only one of the edges on each SEG' pulse. The resolution can be improved by a factor of two if both edges of each SEG' pulse are sensed. When this is done, two SEG pulses are obtained for each SEG' pulse on the position sensor disc. The technique requires a SEG' pulse waveform with a 50:50 mark/space ratio in order to ensure that successive edges occur at equal angular increments. Therefore, "double-edge" sensing can be used to improve the resolution of SELOG position sensors providing the SEG' pulse waveform is suitable. Alternatively, the improvement in resolution can be used to reduce the number of SEG' pulses required per revolution in a given application. For example, six segment boundaries per revolution can be indicated by only three SEG' pulses.

Double-edge sensing can be achieved by the use of a pair of monostables as shown in fig. 3.19. One monostable triggers on the rising edge of the input SEG' pulse and the other monostable triggers on the falling edge of the SEG' pulse. The period of each monostable must be shorter than the minimum expected time between successive edges. The outputs of the two monostables are ORED to generate the SEG signal needed by the sequential counter circuits. The relationship between a six step monostable-generated SEG output and the required "double-edge" input SEG' pulses is shown in fig. 3.20, along with the SYNC' and SYNC pulses. A "single-edge" six step SEG output waveform is also shown for comparison. The SEG' and SYNC' "double-edge" waveforms shown in fig. 3.20 can be generated by a disc pattern and

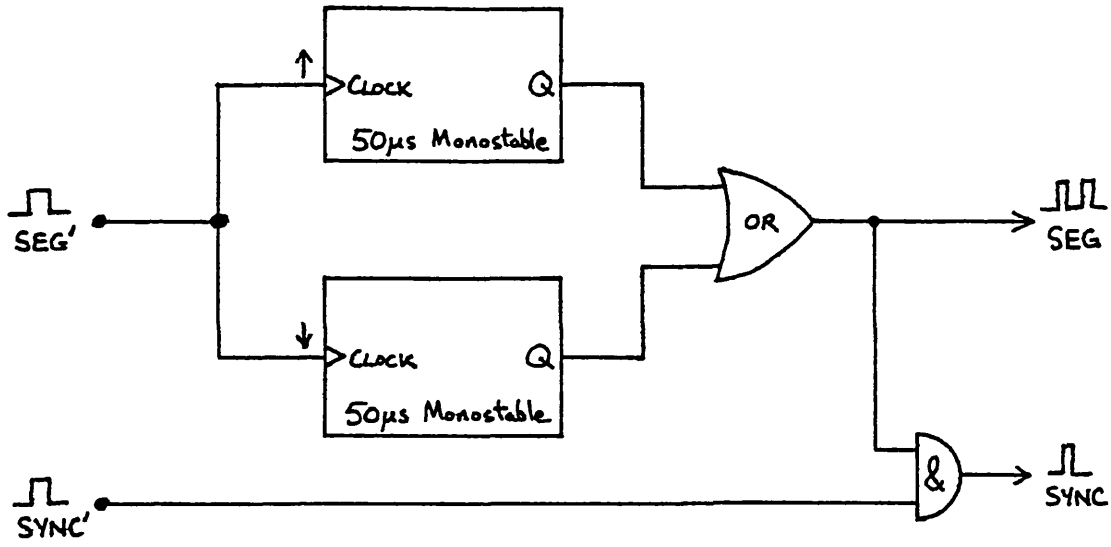


FIG. 3.19. CIRCUIT TO IMPLEMENT DOUBLE-EDGE SENSING ON THE SEG' SIGNALS

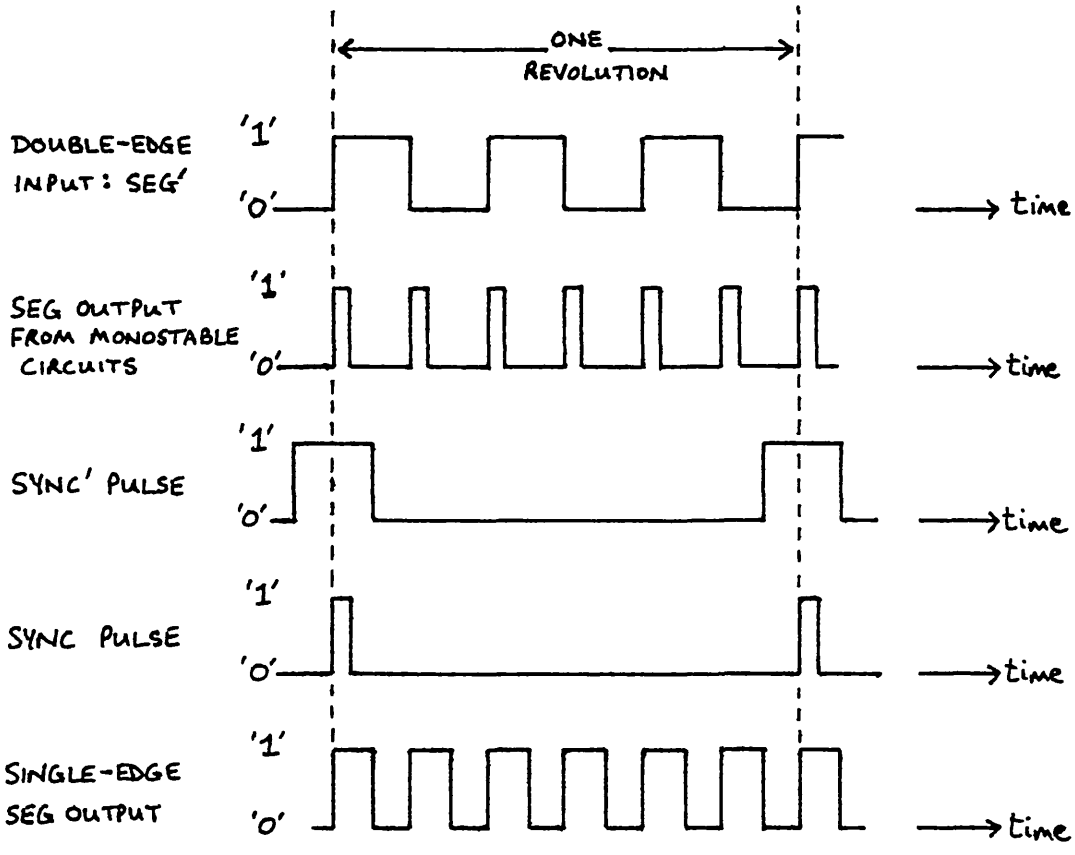


FIG. 3.20. A SIX STEP SEG SIGNAL PRODUCED FROM A THREE PULSE SEG' SIGNAL BY THE CIRCUIT SHOWN IN FIG. 3.19.

sensor arrangement of the form shown in fig. 3.21.

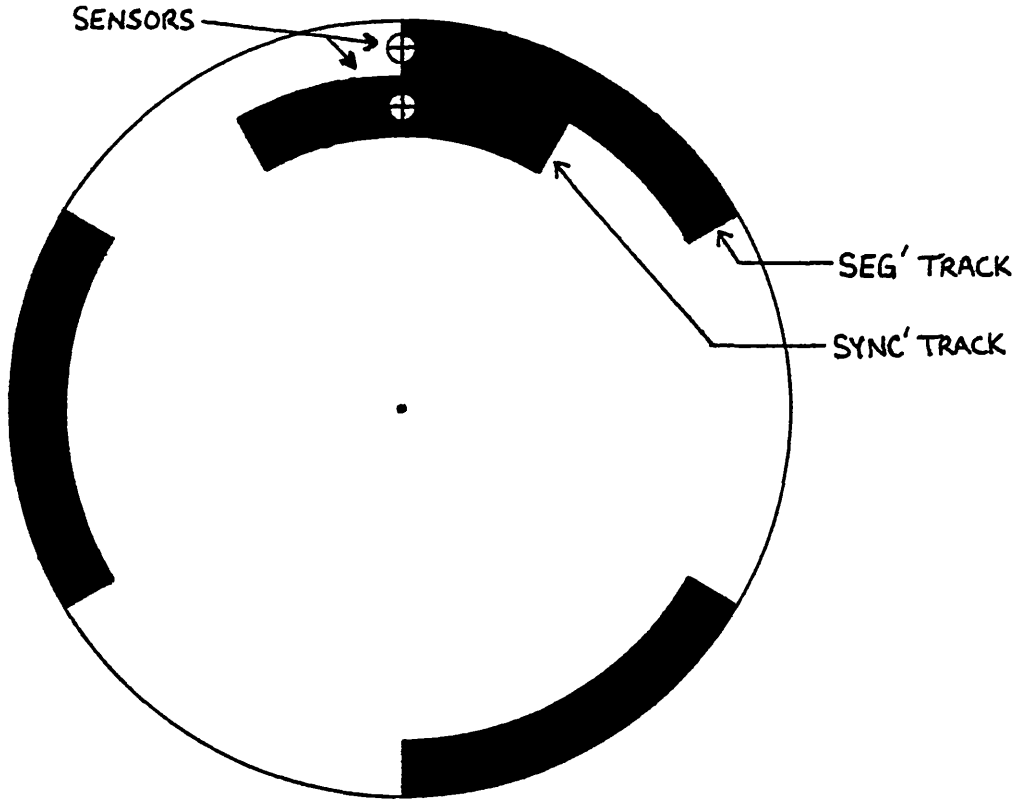


FIG.3.21. DISC AND SENSOR PATTERN SUITABLE FOR USE WITH THE CIRCUIT OF FIG 3.19 TO PRODUCE SIX SEG PULSES PER REVOLUTION

The reduction in the required SEG' pulse pattern density on the position sensor disc is not the only benefit available from double-edge sensing. The rising edges of the SEG pulses generated by the monostable circuits always coincide with the same angular points irrespective of the direction in which the shaft is rotating. The inverting function of the circuit shown in fig. 3.15 is therefore redundant. The SYNC signal can be generated in a "double-edge" system by ANDING the SEG output from the monostables to the SYNC' pulse as shown in fig. 3.19.

In a similar manner to a "single-edge" sensing SELOG system, a "double-edge" sensing SELOG system must be able to detect 'b' steps per revolution, where 'b' is defined by equation 3.13. The number of segment pulses, SEG, must

equal the number 'b'.

i.e. $SEG = b$ 3.24

However, the "double-edge" feature of the system means that:

$$SEG = 2.SEG' \quad 3.25$$

Combining equations 3.25, 3.24, and 3.13 gives:

$$SEG' = \frac{n}{2}.Q.P \quad 3.26$$

Hence the number of SEG' pulses for a given system can be determined. The minimum number of SEG' pulses needed in a "double-edge" sensing system, $SEG'_{min(de)}$, is given when n = 1:

$$SEG'_{min(de)} = Q.P/2 \quad 3.27$$

i.e. $SEG'_{min(de)}$ = Number of Phases x Number of Pole Pairs.

The resolution, θ_{pde}, of a "double-edge" sensing SELOG system is double that for a "single-edge" sensing system with the same number of SEG' pulses.

i.e. $\theta_{pde} = \theta_{pse}/2$ 3.28

Hence from equations 3.28 and 3.18:

$$\theta_{pde} = 180/SEG' \quad 3.29$$

or substituting from equation 3.25:

$$\theta_{pde} = 360/SEG \quad 3.30$$

Finally the resolution, $\theta_{pde(min)}$, of a "double-edge" system with the minimum number of SEG' pulses is given by equation 3.31:

$$\theta_{pde(min)} = 180/SEG'_{min(de)} \qquad 3.31$$

3.3.2.5 Brief Summary of the Pros and Cons of NOLOG, COMLOG, and SELOG Position Sensors

Having discussed the various digital position sensor methods it is possible to conclude that the NOLOG and COMLOG systems:

(a) always give the correct position;

(b) give position information that is directional.

However, these features are gained at the expense of the need for:

(a) a relatively large number of sensors;

(b) a relatively complicated disc pattern.

The SELOG systems are attractive because they only require:

(a) two sensors for sufficient position information;

(b) a very simple disc pattern.

In addition, the "double-edge" sensing SELOG system allows a reduction in the disc pattern density by a factor of two for a given angular resolution. Unfortunately, the SELOG benefits are gained at the cost of having:

(a) no continuous directional information. If the shaft direction changes between pulses the system cannot tell. Hence desynchronisation is a possibility;

(b) no synchronisation at starting; hence a special open-loop starting procedure is required. The system needs at the most, one revolution to synchronise.

Motor reversal is possible with any of the three

position sensor arrangements and so they can all be used to autopilot bidirectional synchronous motor systems.

In general terms the SELOG systems require fewer sensors than the NOLOG or COMLOG systems. The logic required by any of the systems can be implemented by either discrete logic circuits (TTL or CMOS) or by microprocessor. The discrete logic solutions are demonstrated for the magslip system in this chapter. A microprocessor solution for SELOG position sensor systems used on the magslip and the 7 phase square wave motor is the topic covered in Chapter 4.

3.4 Adjustment of Load Angle

The load angle of an autopiloted synchronous machine is held at a particular average value by the rotor position sensor. With reference to figures 3.6(a) and (b), the applied voltage V directly determines the spacial position of the resultant flux ϕ_r. The spacial position of the rotor flux ϕ_f is determined by the rotor direct axis. The position of the rotor direct axis is detected by the shaft position sensor and the resulting signals are used to control the applied voltage V. Hence the average load angle δ between ϕ_r and ϕ_f or V and E is determined by the shaft position sensor. The angle can be changed by manipulating the position sensor signals. This can be achieved in two ways:

(a) mechanically: the position detector sensors can be rotated with respect to the stator windings. This causes ϕ_r to shift with respect to ϕ_f.

(b) Electronically: the position sensor signals can effectively be shifted by a variety of electronic techniques.

The mechanical variation of load angle is simple and reliable. Twisting the position sensors in the opposite

direction to the rotation of the rotor increases the load angle. Moving the sensors in the same direction as the rotation of the rotor, reduces the load angle. In a two pole machine the load angle changes by an amount equal to the physical twist angle. The main advantages of mechanical load angle control are that the angle is continuously variable, thus allowing the torque to be maximised at any speed, and the position sensor only requires a resolution sufficient to detect the number of switching steps per revolution; i.e. six for a 3 phase 120^{o} - 60^{o} waveform. The main disadvantage of mechanical load angle variation is the need to move the sensors. A sensor that can move is inherently less reliable than one that is firmly located. In addition, if the load angle is controlled by an electronic signal, there is a requirement for an electro-mechanical servo to rotate the sensors. If the motor is required to operate in both directions, the position sensor must likewise be able to shift in both directions from the 0^{o} load angle setting.

The electronic variation of load angle can be achieved by processing the position sensor signals either by:

(a) time delay circuits, or:

(b) combinational or sequential logic circuits.

3.4.1 Time Delay Adjustment of Load Angle

A time delay imposed on the position sensor signals as they occur can be used to vary the operating load angle over the required range of $+90^{o}$ to 0^{o}. This can be achieved by arranging the position sensor so that with zero time delay, the motor operates with a $+90^{o}$ load angle. The introduction of a time delay into the position sensor signals delays the switch on of the applied winding voltage, V. Therefore, during the step following the delay, the average load angle δ is less than $+90^{o}$. The load angle can

actually become negative if the time delay is long enough.

The selection of a particular load angle at a given speed is best illustrated by an example. The situation with zero time delay is shown in figures 3.22(a) and (b). Fig. 3.22(a) shows the relationship between the applied winding voltage V and the back-emf E. Fig. 3.22(b) shows the average spacial positions of the machine fluxes over a step. The machine is operating with a $+90^{\circ}$ load angle. When a time delay of t_d seconds is introduced into the position sensor signals the nett effect is to cause the average position of the rotor flux ϕ_f to change. This is because at a motor speed of N rpm the rotor moves through an angle θ_t during the delay time t_d.

i.e. $$\theta_t = 6.N.t_d \quad \text{degrees} \qquad 3.32$$

Therefore, the average position of the rotor flux during the step following the time delay is shifted by the angle θ_t. For example, in a two pole motor spinning at 3000 rpm, a time delay of 1.67ms results in the average position of the rotor flux ϕ_f shifting by 30°. The shift in the rotor flux is shown in fig. 3.22(d) and it causes the load angle to change to 60° as shown in fig. 3.22(c).

The actual load angle δ for any time delay is given by:

$$\delta = 90 - \theta_t \quad \text{degrees} \qquad 3.33$$

The time delay required to achieve a particular load angle is found by combining equations 3.32 and 3.33 to give:

$$t_d = \left(\frac{90 - \delta}{6.N}\right) \quad \text{seconds} \qquad 3.34$$

Equation 3.34 shows that the time delay method has a serious drawback if a load angle other than 90° is required at zero speed (N = 0). The necessary time delay given by

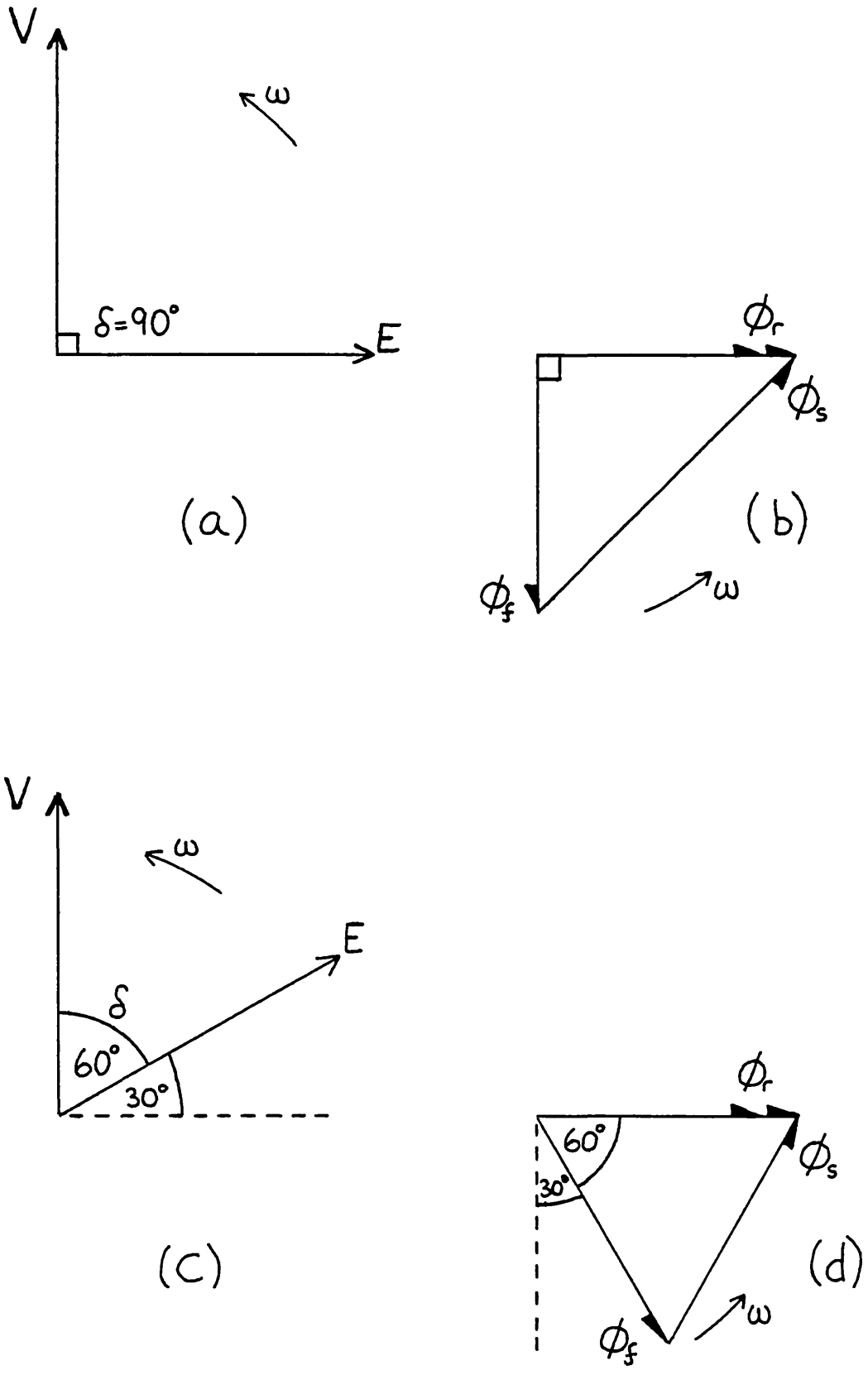


FIG. 3.22 (a) & (b) — ZERO TIME DELAY IN POSITION SENSOR SIGNALS.

FIG. 3.22 (c) & (d) — 1·67ms DELAY IN SENSOR SIGNALS AT 3000 RPM.

FIG 3.22 "TIME DELAY" CONTROL OF WORKING LOAD ANGLE

equation 3.34 is infinite for such conditions. Unfortunately, since a cylindrical rotor synchronous motor produces no torque at zero speed with a load angle of 90°, it is impossible to start such a motor if the load angle is to be varied between 0° and 90° by the time delay method. One way of overcoming the problem to some extent is to reduce the load angle δ_o that exists for $t_d = 0$. Equation 3.34 then becomes:

$$t_d = \left(\frac{\delta_o - \delta}{6.N} \right) \quad \text{seconds} \qquad 3.35$$

The value of δ_o chosen depends on the starting torque required but a suitable value might possibly be 60° say. The system could then successfully start and once moving the load angle could be varied from 60° down to 0°. The fact that load angles in the range 90° to δ_o° cannot then be obtained is not necessarily a disadvantage since the maximum required load angle depends on the maximum desired operating speed of the motor.

An alternative method of overcoming the zero starting torque problem is to make use of the fact that if a position sensor can detect 'b' steps per revolution, the load angle can be changed in increments of $(360/b)^{\circ}$ by means of electronic logic. The method by which this can be achieved is explained in section 3.4.2. A six step position sensor allows 60° steps in the motor load angle. The 60° increments are a result of the six steps associated with the position sensor and the motor applied voltages. It is therefore possible to change the zero time delay load angle δ_o by 60° from 90° to 30°. Once the motor has been successfully started, δ_o can be switched back to 90°.

The starting torque problem inherent with the 0° to 90° load angle system can thus be overcome but a much more difficult problem is inherent in the time delay method. Reference to equation 3.34 shows that the required time

delay t_d is inversely proportional to the speed N. Therefore, at any instant the system must:

(a) have an accurate measure of the motor speed;

(b) compute the required time delay;

(c) implement the computed time delay.

The accurate measurement of speed and the calculation of the required time delay are both relatively easy especially since the advent of microprocessors. The time delay can be implemented with either a monostable type of circuit or some form of programmable timer which can include a microprocessor. However, the main problem with the time delay generation is the large range of time delays needed to operate over a wide speed range.

Against the disadvantages of the time delay method, it can be said that the example 3 phase 2 pole motor only needs a position sensor able to detect six points every revolution, (for example the NOLOG position sensor pattern shown in fig. 3.8), and it can provide continuous load angle variation from δ_o to 0^o. However, it was decided that the problems in implementing such a system outweighed the benefits and so consideration was given to logical methods of load angle control. These methods are discussed in section 3.4.2.

3.4.2 Logical Methods of Adjusting Load Angle

The digital position sensor systems discussed in section 3.3 are able to detect which step of a revolution a motor shaft is passing through. A set of position code signals are generated during each revolution, and the signals control the inverter outputs. Each combination of output voltages from the inverter can be associated with a particular binary code from the position sensor. The relationship between the binary position code and the inverter

output states is such that the motor is synchronised and runs with a fixed "average" load angle as explained in section 3.2.

The adjustment of load angle by mechanical rotation of the position sensor can be simulated electronically by changing the relationship between the binary position codes and the inverter output combinations. The method is most easily explained by way of an example. The inverter output voltages for the magslip system are as shown in fig. 3.2(a) and there are six possible combinations during a cycle. Therefore, the minimum number of steps that the position sensor has to be able to detect is also six. A suitable three bit Gray code $B_2B_1B_0$ is shown in fig. 3.23(a). The sequence of inverter output voltages is shown in fig. 3.23(b).

	Position Sensor Code		
Step Number P(n)	B_2	B_1	B_0
P(0)	0	0	0
P(1)	0	0	1
P(2)	0	1	1
P(3)	0	1	0
P(4)	1	1	0
P(5)	1	0	0

(a)

	Inverter Output Voltage Polarity		
Step Number I(n+k)	Phase A	Phase B	Phase C
I(0)	$+V_{ph}$	$-V_{ph}$	0
I(1)	$+V_{ph}$	0	$-V_{ph}$
I(2)	0	$+V_{ph}$	$-V_{ph}$
I(3)	$-V_{ph}$	$+V_{ph}$	0
I(4)	$-V_{ph}$	0	$+V_{ph}$
I(5)	0	$-V_{ph}$	$+V_{ph}$

(b)

FIG. 3.23. A TYPICAL POSITION SENSOR GRAY CODE SEQUENCE (a), AND THE INVERTER OUTPUT VOLTAGE SEQUENCE (b).

By using suitably designed electronic logic circuits it is possible to produce a function f_k such that the mapping of the position sensor codes to the inverter voltage outputs is defined as:

$$f_k : P(n) \longrightarrow I(n+k);\ k,n \in Z_6 \qquad 3.35$$

where Z_6 is the commutative ring of residues modulo 6; i.e. $\{0,1,2,3,4,5\}$. The system can be physically adjusted so that the motor operates with an average load angle of 0^o for a value of k=0. Then for k=1, the inverter output voltages are one step ahead of the 0^o load angle inverter output state. Since the applied phase voltages determine the axis of the resultant flux ϕ_r in the motor, the effect with k=1 is that the axis of ϕ_r is moved 60^o on from its 0^o load angle position. That is the angle between ϕ_r and ϕ_f is increased by 60^o and so the average load angle also increases by 60^o. Hence the average value of the load angle becomes 60^o instead of 0^o. If k=2, the inverter output voltages are advanced by two steps and the load angle increases by 120^o. The 60^o adjustments that can be made to the load angle are rather coarse, but they would be useful in a simple motor system in which a 0^o load angle could be used for starting and low speeds, with a 60^o load angle used for higher motor speeds. The load angle can be retarded from the k=0 setting by selecting negative values of k. For k=-1 the load angle is reduced by 60^o. Therefore, if a 0^o load angle occurs for k=0, a value of k=-1 would give a -60^o load angle. A coarse negative load angle adjustment would be useful in the time delay method of load angle control described in sub-section 3.4.1. In such a system the k=0 value of load angle would be set at 90^o. Then a value of k=-1 would select a load angle of 30^o (90^o-60^o) and so overcome the starting problem discussed in sub-section 3.4.1.

The mapping obtained for k=3 is useful since it causes the inverter to advance half a cycle through its sequence. This is precisely the condition required to reverse the stator flux and hence reverse the motor direction. Therefore, increasing the load angle by 180^o puts the motor into reverse. If k=0 gives 0^o load angle in one direction, then k=3 gives 0^o load angle in the reverse direction.

The resolution δ_r available in the adjustment of load angle by the logical mapping method depends on the mechanical resolution θ_p of the position sensor. The mechanical resolution θ_p is related to the electrical load angle resolution by the number of poles P in the machine stator windings.

i.e. $\delta_r = \theta_p.(P/2)$ degrees 3.36

Using equation 3.14 to substitute for θ_p gives:

$$\delta_r = 180/(n.Q) \text{ degrees} \quad 3.37$$

Therefore, for the basic magslip system with n=1 and Q=3, the load angle can be varied in 60° steps. By dividing each revolution into more steps, it is possible to improve the resolution of the load angle adjustment. For example, a value of n=2 results in the position detector being able to detect 12 steps per revolution as indicated in fig. 3.24(a). The six possible inverter states each occur for two consecutive steps as shown in fig. 3.24(b).

The mapping function is then defined as:

$$f_k : P(n) \longrightarrow I(n+k);\ k,n \in Z_{12} \quad 3.38$$

where Z_{12} is the commutative ring of residues modulo 12; i.e. $\{0,1,2,3,4,5,6,7,8,9,10,11\}$. The resultant load angle resolution with n=2 and Q=3 is found to be 30° by equation 3.37. Hence with such a system it is possible to select load angles of 0°, 30°, 60°, and 90°. The size of load angle adjustment can be made even smaller by making n=3 and so on. However, logical load angle adjustment can only provide stepped variation in the load angle whereas continuously variable adjustment is possible with both the mechanical and time delay methods. Nevertheless, the logical method of load angle adjustment is superior to the time delay method in two respects. Firstly, the "shifts"

	POSITION SENSOR GRAY CODE			
STEP NUMBER P(n)	B_3	B_2	B_1	B_0
P(0)	0	0	0	0
P(1)	0	0	0	1
P(2)	0	0	1	1
P(3)	0	1	1	1
P(4)	0	1	1	0
P(5)	0	1	0	0
P(6)	1	1	0	0
P(7)	1	1	1	0
P(8)	1	1	1	1
P(9)	1	0	1	1
P(10)	1	0	0	1
P(11)	1	0	0	0

(a)

	INVERTER OUTPUT VOLTAGE POLARITY		
STEP NUMBER I(n+k)	PHASE A	PHASE B	PHASE C
I(0)	$+V_{ph}$	$-V_{ph}$	0
I(1)	$+V_{ph}$	$-V_{ph}$	0
I(2)	$+V_{ph}$	0	$-V_{ph}$
I(3)	$+V_{ph}$	0	$-V_{ph}$
I(4)	0	$+V_{ph}$	$-V_{ph}$
I(5)	0	$+V_{ph}$	$-V_{ph}$
I(6)	$-V_{ph}$	$+V_{ph}$	0
I(7)	$-V_{ph}$	$+V_{ph}$	0
I(8)	$-V_{ph}$	0	$+V_{ph}$
I(9)	$-V_{ph}$	0	$+V_{ph}$
I(10)	0	$-V_{ph}$	$+V_{ph}$
I(11)	0	$-V_{ph}$	$+V_{ph}$

(b)

FIG. 3.24. A 12 STEP POSITION SENSOR GRAY CODE (a), AND THE ASSOCIATED INVERTER OUTPUT VOLTAGE STATES (b).

in load angle are not speed dependent. Secondly, there is no problem in the selection of zero load angle at zero speed.

It is important to note that the load angle shift achieved by logic circuits is related to a particular direction of motor rotation. Therefore, a 30^{o} shift in the forward motor direction would be a -30^{o} shift if the motor were running in reverse. Hence the mapping f_k must be in the correct sense with respect to the inverter sequence and and motor direction.

The logic required to implement the function f_k is relatively simple and there are several methods by which it can be implemented. The logic solution chosen for a particular system depends to some extent on the type of position sensor used. For example, the combinational logic required by a COMLOG position sensor basically provides one mapping of shaft positions to inverter states. By suitable design, it is possible to extend the combinational logic to provide several alternative mappings; i.e. several load angles. The logical method of load angle adjustment was actually implemented on the magslip system in three ways. They were respectively:

(a) a discrete combinational logic circuit based on a COMLOG position sensor;

(b) a discrete sequential plus combinational logic circuit based on a SELOG position sensor;

(c) a software "look-up" table implemented by microprocessor and based on a SELOG position sensor.

The basic approach used for each solution is briefly outlined below.

3.4.2.1 COMLOG - Based Load Angle Adjustment Circuit

The mapping from the position sensor code to the required inverter control signals can be split into two stages. The first stage of the mapping relates a particular position sensor code to one of the possible inverter states. The second stage then relates the selected inverter state to the corresponding inverter control signals.

The first stage mapping depends on the required motor load angle. Therefore, in order to have a selection of 'y' load angle settings (δ_0, δ_1, ... δ_{y-1}), there must be y "first stage" mapping units, each one performing the mapping from position codes to inverter states for one load angle. The y first stage logic units operate in parallel. They are indicated schematically as blocks A_0 to A_{y-1} on fig. 3.25. A motor can only operate with one particular load angle setting at any given time. Therefore the outputs of the relevant first stage logic block are selected by logic block B and routed to the second stage mapping unit, which is represented by block C on fig. 3.25. Block C maps the inverter states to the corresponding inverter control signals. This mapping does not depend on the load angle and so only one logic unit is necessary.

The Boolean logic functions necessary to perform the mapping functions in the first and second stage blocks can be realised by the combination of standard logic gates (And, Or, Nand, Nor, and Not). Alternatively it is possible to use "look-up" tables implemented in Read-Only-Memory (ROM) integrated circuits to perform the mappings.

No mention of the direction of motor rotation has been made with regard to the system shown schematically in fig. 3.25. However, if it is assumed that logic blocks A_0 to A_{y-1} provide load angle values between 0° and 90° in the forward motor direction, then a further set of logic blocks are necessary to achieve the identical load angle settings

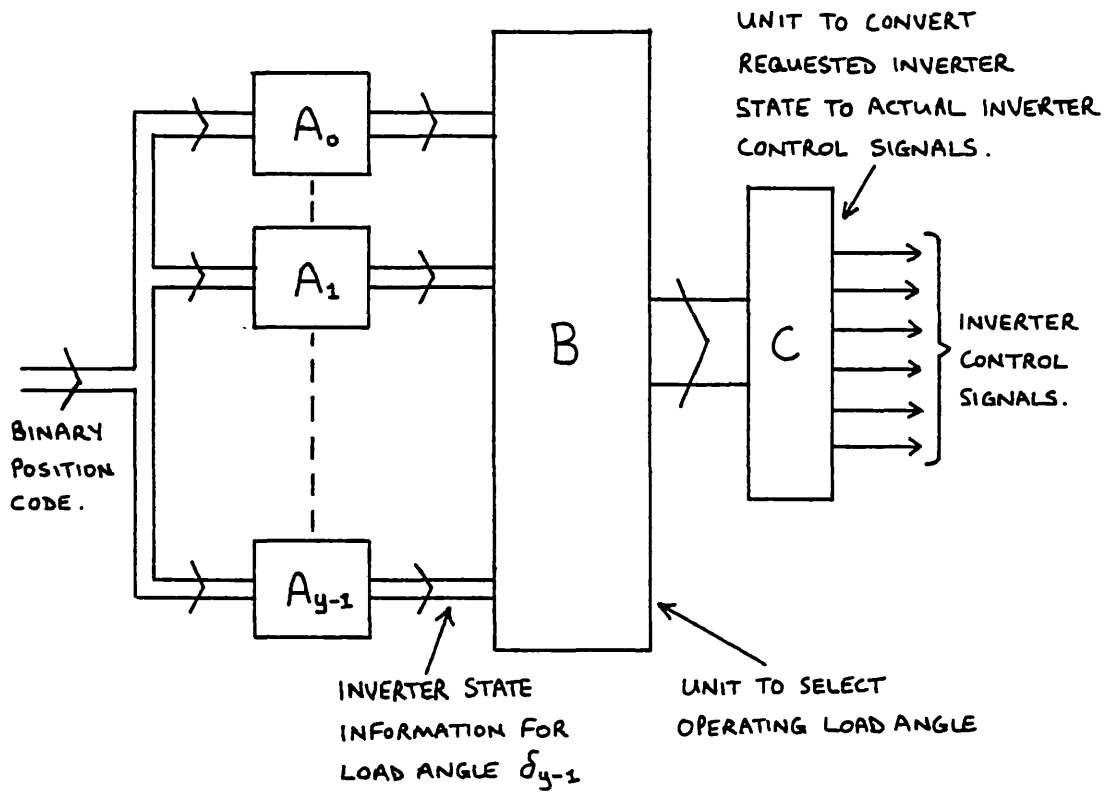


FIG. 3.25. SCHEMATIC OF A COMLOG BASED LOAD ANGLE ADJUSTMENT CIRCUIT

in the reverse direction. The logic block performing the mapping for the 0° load angle can be used in either direction of rotation providing the motor is reversed by means of field flux reversal. Hence, to achieve load angles of 0°, 30°, 60°, and 90° in both directions requires at least seven first stage logic blocks. Since the output of only one of the blocks is in use at any time, this particular method does not make efficient use of the logic gates. If such a system is implemented by TTL logic gates, the total current consumption is large even though only a small fraction of the total circuit is being used at any time.

A combinational logic system of the form shown in fig. 3.25 was constructed for the magslip system. The circuit provides load angles of 0°, 30°, 60°, and 90° in the forward

direction. The circuit diagram and a brief circuit description are given in section 3.7.1.

3.4.2.2 Discrete - Logic SELOG - Based Load Angle Adjustment Circuit

A SELOG position sensor system can be arranged so that the adjustment of load angle is a simple mathematical process. To achieve this the position of the motor shaft is registered in the form of a standard binary count by a synchronous counter. The counter advances through its sequence of binary numbers once during every revolution of the motor shaft. Since the counter is synchronised with the shaft, each binary number is related to a distinct sector of the shaft's rotation.

Fixed load angle operation can be achieved by mapping the binary numbers to the corresponding inverter control signals using combinational logic. However, different load angles can be selected if the binary numbers are mathematically processed prior to being mapped into inverter control signals. A "shift" can be imposed between the binary numbers and the associated inverter control signals by adding (or subtracting) a binary offset number 'B_{shift}' to the counter output number 'B_{count}' to give a resultant binary number 'B_{res}':

$$B_{res} = B_{count} + B_{shift} \qquad 3.39$$

The addition of an offset number can cause the resultant number to go outside the range of numbers generated by the synchronous counter. For example, a 12 step binary counter can be designed to cycle through the numbers 0 to 11. When counting up, the counter reaches 11 and then on the following change goes to 0. Similarly, when the counter is counting down and reaches 0 the next number is 11. Therefore, the counter never registers a number outside the range of 0 to 11, and so the mapping of the counter

output to the inverter control signals is restricted to twelve possibilities. However, when an offset number is added to the counter output it is possible for numbers outside the 0 to 11 range to occur. Two simple examples using equation 3.39 demonstrate this:

For $B_{count} = 11$ and $B_{shift} = 1$, $B_{res} = 12$;

and for $B_{count} = 0$ and $B_{shift} = -1$, $B_{res} = -1$.

In both cases B_{res} is an "illegal" number since it is outside the range of numbers for which the inverter control mapping is defined. The problem can be overcome by forcing B_{res} to "wrap back" into the other end of the permitted number scale, (e.g. 12 becomes 0, and -1 becomes 11). The "wrap back" logic must be able to deal with any possible illegal result in a given system.

The schematic arrangement of the complete load angle variation system is shown in fig. 3.26a.

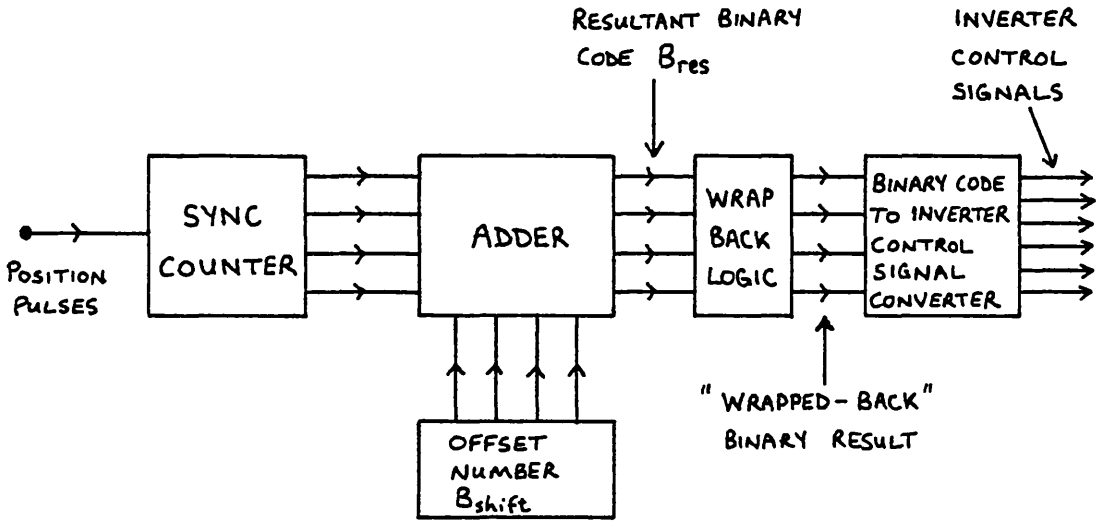


FIG. 3.26a. SELOG BASED LOAD ANGLE ADJUSTMENT SYSTEM.

The method of operation is best illustrated by an example table of possible states based on the magslip inverter. Fig. 3.26(b) shows the relationship between the shaft position and the binary counter output for a 12 step position sensor. Fig. 3.26(c) shows how the inverter states can be related to the binary number B_{res}.

Shaft Position (Step Number)	Counter Output B_{count}
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011

(b)

Binary Number B_{res}	Inverter Phase Outputs		
	A	B	C
0000	$+V_{ph}$	$-V_{ph}$	0
0001	$+V_{ph}$	$-V_{ph}$	0
0010	$+V_{ph}$	0	$-V_{ph}$
0011	$+V_{ph}$	0	$-V_{ph}$
0100	0	$+V_{ph}$	$-V_{ph}$
0101	0	$+V_{ph}$	$-V_{ph}$
0110	$-V_{ph}$	$+V_{ph}$	0
0111	$-V_{ph}$	$+V_{ph}$	0
1000	$-V_{ph}$	0	$+V_{ph}$
1001	$-V_{ph}$	0	$+V_{ph}$
1010	0	$-V_{ph}$	$+V_{ph}$
1011	0	$-V_{ph}$	$+V_{ph}$

(c)

FIG. 3.26:- RELATIONSHIP BETWEEN SHAFT POSITION AND B_{count} (b); AND RELATIONSHIP BETWEEN B_{res} AND INVERTER OUTPUT STATES (c).

Assume that the motor is rotating such that the count sequence is from 0 to 11, and the system is set up to operate with zero load angle for $B_{shift} = 0$. The selection of $B_{shift} = 1$ causes the inverter output state selected by B_{res} to advance one twelfth of a cycle ahead of the 0° ($B_{shift} = 0$) inverter state. Similarly, a shift of one sixth of a cycle is achieved for $B_{shift} = 2$. These shifts result in motor load angles of $+30^{\circ}$ and $+60^{\circ}$ respectively. Selection of negative values of B_{shift} when the count

sequence is from 0 to 11 causes the inverter output state to shift backwards by fractions of a cycle. Negative load angles of -30°, -60°, and -90° can be selected by making B_{shift} equal to -1, -2, and -3 respectively.

In the reverse motor direction the count sequence is from 11 to 0. The inverter states must be shifted in the opposite sense to achieve positive load angles. Hence a load angle of $+30^{\circ}$ is obtained by setting $B_{shift} = -1$.

The adder unit shown in fig. 3.26a can deal with negative values of B_{shift} if they are entered in two's complement binary code. Therefore, a standard four bit full adder circuit would be suitable for the binary codes shown in figures 3.26(b) and (c).

Reversal of the motor direction of rotation can also be achieved by suitable manipulation if B_{shift}. The stator flux in the machine can be reversed by choosing B_{shift} so that the inverter is advanced through half a cycle. In the 12 step example discussed above, this can be brought about by adding 6 to whatever value of B_{shift} is required for a particular load angle. The wrap-round logic must of course cope with any "illegal" resultant numbers.

The method discussed above is very attractive in practical systems since it can achieve load angle variation and electronic motor reversal by a simple arithmetic operation. A circuit of this form was constructed for the magslip system. The circuit provides load angles of 0°, 30°, 60°, and 90° in both motor directions. The circuit diagram and a brief circuit description are given in subsection 3.7.2.

The arithmetic method of load angle control can equally well be applied to COMLOG systems providing the position sensor code has a straight numerical sequence. Therefore, if the position sensor produces Gray code, this

must first be converted to an ordered numerical code. The converted code can then be processed by a full adder and the associated wrap-round logic just as in the example discussed above.

3.4.2.3 Microprocessor - Based Load Angle Adjustment Circuit

A microprocessor can be used to autopilot a synchronous motor. The software logic functions available in the instruction set are ideal for the implementation of an adjustable load angle system. All of the control electronics between the position sensor unit and the inverter can be replaced by a microprocessor.

The form of the available position sensor signals are important when choosing the manner in which a microprocessor should be used. For example, if the position sensor signals contain only incremental position information (e.g. the SEG' and SYNC' signals in a SELOG system), the first task for the microprocessor is to count the signals and calculate the rotor position. Having determined the position it can then map the count value to the inverter control signals by means of a look-up table in memory. Simple software adjustment of the table pointer enables changes in the mapping to be achieved without any extra hardware being necessary, (i.e. the load angle adjustments are implemented by software). However, if the position sensor information is in the form of binary code, the microprocessor actually has fewer tasks to perform. For example, the information could be Gray code produced by a COMLOG type of position sensor, or pure binary code produced by a counter in a SELOG position sensor system. Regardless of the type of code, it is usually a simple task for the microprocessor to read in the code and map it to the relevant inverter control signals. Hence, the microprocessor does not have to determine the shaft position in this case; it merely has to act as an adjustable mapping function.

Further discussion on the implementation of microprocessor based autopiloted motor systems is the subject of Chapter 4. The magslip was controlled by a microprocessor using both of the methods outlined above. The hardware and software details of the magslip microprocessor based systems are given in Chapter 4. The systems allowed load angles of 0°, 30°, 60°, and 90° to be selected in either motor direction. The maximum speed achieved under microprocessor control was the same as that achieved with the conventional logic circuits.

3.5 Basic Magslip Motor Drive System

The basic magslip drive system used to obtain the performance characteristics presented in section 3.6 is shown in schematic form in fig. 3.27.

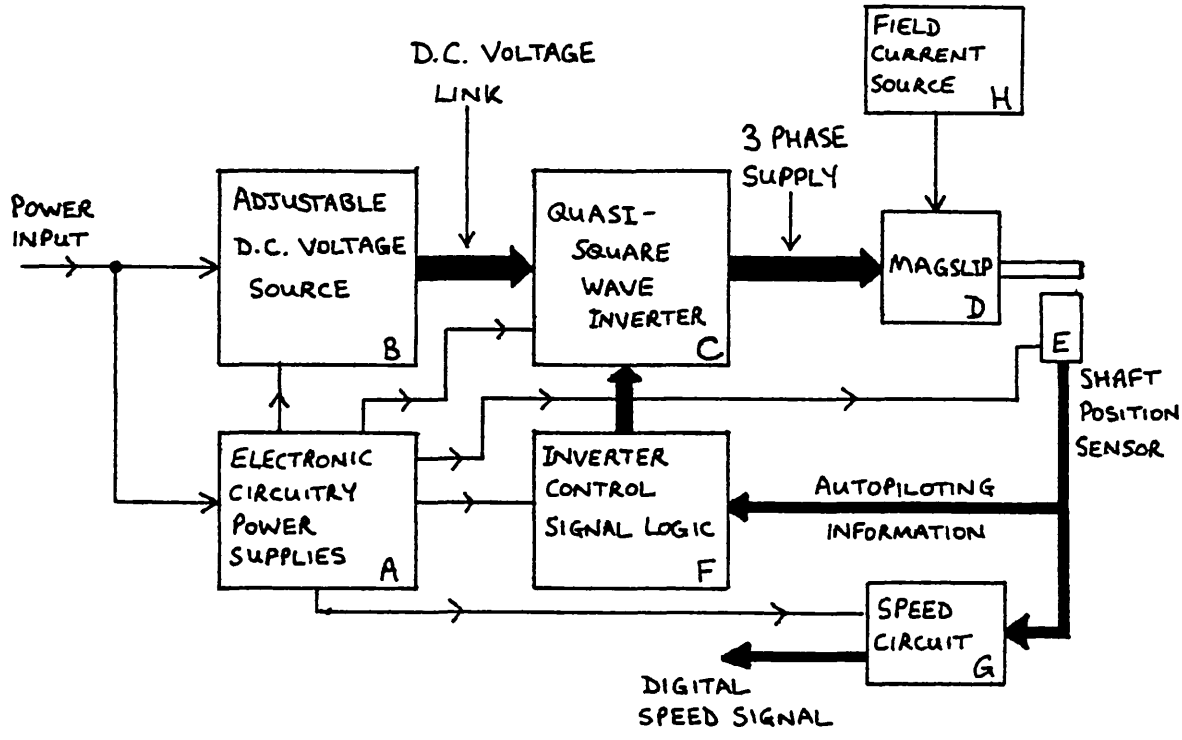


FIG. 3.27. SCHEMATIC OF THE MAGSLIP MOTOR DRIVE SYSTEM.

The system basically consists of seven blocks (labelled A to H on fig. 3.27). Block A represents a set of regulated d.c. power supplies that provide all the necessary low voltage rails for the electronics in the system. The adjustable d.c. voltage supply (block B) provides the regulated d.c. voltage rails for the quasi-square wave inverter (block C). The inverter drives the magslip motor (block D) and the rotor position of the magslip is sensed by the shaft position sensor (block E). The position sensor signals are processed by the inverter control logic (block F) and hence are used to schedule the inverter switching sequence. The position sensor signals are also processed by the speed circuit (block G) which produces a suitable digital speed signal for the tachometer discussed in Chapter 8. The field current for the magslip is provided by a constant current circuit (block H). The circuit blocks are each discussed briefly in the following sub-sections (3.5.1 to 3.5.7).

3.5.1 The Magslip (Block D)

A data sheet for the magslip machine is contained in Appendix 2A. It was decided that the inverter system should be capable of supplying a voltage waveform of up to ± 50 volts on to the magslip phase terminals. (i.e. the terminal to terminal voltage would be a maximum of 100 volts.) This does result in the magslip being over driven because the data sheet gives a maximum stator voltage rating of 53 volts (rms) between phase terminals (75 volts peak). However, the risks involved in overdriving the magslip were felt to be worthwhile because the higher voltage allowed a wider operating speed range to be achieved for the experiments. The maximum current I_{max} drawn by the magslip with 100 volts across two phase windings in series occurs at d.c. conditions. The d.c. resistance between terminals is 19Ω and so I_{max} is about 5 amps. Hence the adjustable d.c. voltage source was fused at 5 amps. In practice the maximum continuous stator current that the magslip can withstand is

much less than 5 amps and a limit of 2 amps was arbitrarily chosen. Therefore, the adjustable d.c. voltage supply and the inverter were designed for a maximum rated current of 2 amps.

3.5.2 The Regulated D.C. Power Supplies (Block A)

The system was designed to run from four voltage rails: +15 volts; -15 volts; +5 volts; and -5 volts. The supplies were constructed using standard electronic techniques. Little purpose is served by presenting a circuit diagram and it is sufficient to say that both the ±15 volt supply and the ±5 volt supply were based on a transformer feeding a smoothing capacitor via a rectifier unit, with semiconductor regulators producing the stabilized voltage rail outputs. Each voltage rail was capable of supplying a current of 1 amp.

3.5.3 The Adjustable D.C. Voltage Source (Block B)

The circuit diagram of the adjustable d.c. voltage source is shown in fig. 3.28. The circuit obtains unregulated ±55 volt supply rails from a three phase variac and a rectifier/capacitor arrangement. The output voltages (V_p and V_n) are set by a programming voltage which can be varied by potentiometer VR1. The positive and negative output voltages are regulated independently and can be varied from about 0 volts up to ±50 volts. The TIP31A and TIP32A transistors, used for TR1 and TR4 respectively, are not rated for the 70 volts that can occur across them, but neither failed during the system tests. The output power transistors (TR3 and TR6) were mounted on independent 1.1 °C/W heatsinks. The circuit could supply ±2 amps continuously at any output voltage. There was an insignificant amount of ripple on the output voltages, and the voltage regulation was adequate for the required application.

Some care was necessary when connecting the adjustable

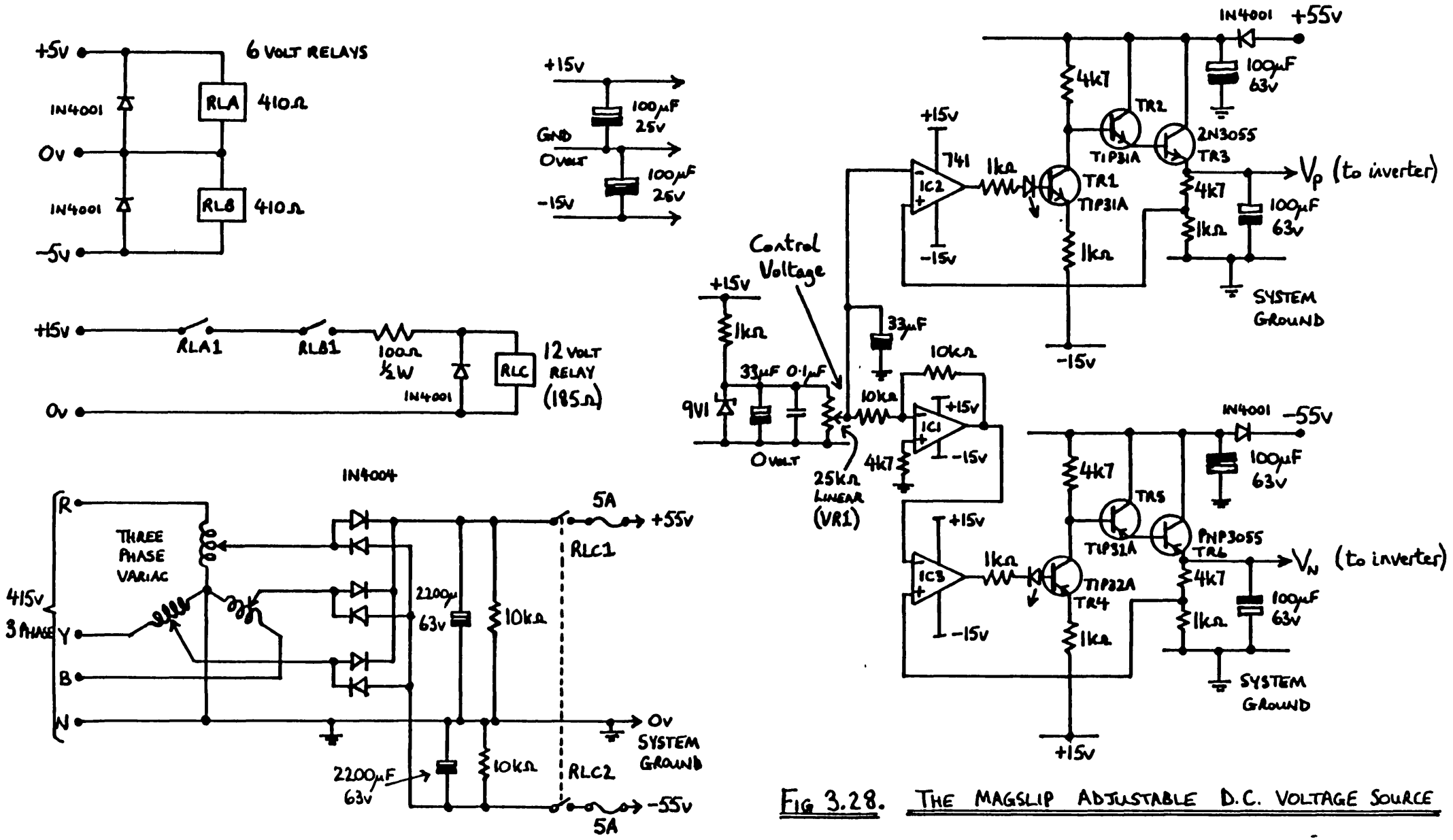


FIG 3.28. THE MAGSLIP ADJUSTABLE D.C. VOLTAGE SOURCE

d.c. power supply to other parts of the system because the neutral connection of the supply variac was connected to the 0 volts rail (system ground). The outputs of the supply were fused to afford some short circuit protection, but no electronic current limit facility was incorporated. The relays (RLA, RLB and RLC) were added to prevent the ±55 volt supply being connected to the circuitry unless the low voltage supplies were present. This precaution reduces the chances of damage to the power supply and inverter.

The circuit worked well without any problems during the test period.

3.5.4 The Shaft Position Sensor (Block E)

The basic shaft position sensor produced the inverter control signals directly and hence can be classified as a "NOLOG" unit as defined in sub-section 3.3.2.1. The switching pattern was of the form shown in fig. 3.8. The pattern was transferred from an ink drawing onto a transparent acetate sheet by means of a photo-mechanical transfer. The acetate sheet was glued onto a four inch diameter perspex disc by using cyanoacrylate adhesive. Small 14 volt L.E.S. bulbs were used for the light sources in the position sensor. Metal tubes were glued onto the bulbs as shown in fig. 3.8. This was done to ensure that a bulb only directs light towards the photodiode directly opposite it. To allow the 14 volt bulbs to run from the 15 volt supplies, they were fed via two series connected silicon diodes (type 1N4001) which drop just over 1 volt.

Six light bulbs and six photodiodes were required to provide the necessary inverter control signals. The photodiodes were employed in a reverse biassed mode. Each photodiode required a transistor amplifier of the form shown in fig. 3.29. The amplifiers were mounted directly adjacent to the photodiodes to reduce the possibility of electromagnetic noise disturbing the very small photodiode

signals.

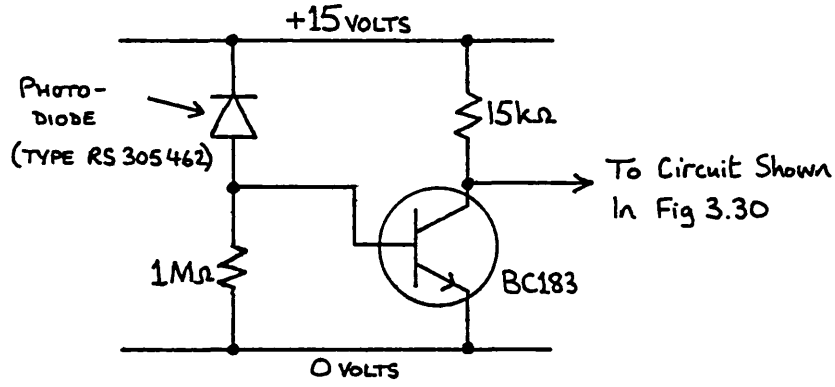


FIG. 3.29. TYPICAL PHOTODIODE AMPLIFIER IN THE SHAFT POSITION SENSOR.

The signals from the six photodiode amplifiers then pass to the circuit shown in fig. 3.30. This circuit performs two functions. Firstly, it converts the amplified photodiode signals into clean square waves by means of TTL schmitt triggers (type 7414). Secondly, it senses the control signals for each phase (e.g. A1 and A2) to check that only one control signal per phase is active. If both phase signals are active at any time it means that either one of the light bulbs has failed or a photodiode is faulty. Two active control signals per phase cause both power transistors to come on and the inverter can consequently be damaged. Therefore, if both signals are active, neither are allowed to pass on to the inverter driver circuits. The processed signals are indicated by means of a dash, (e.g. A1') on the circuit diagram. The outputs labelled A1T, A2T, etcetera are required by the tachometer circuit described in sub-section 3.5.6. The circuit shown in fig. 3.30 was assembled on a standard piece of prototyping circuit board. No problems were experienced in operation.

3.5.5 Quasi Square Wave Voltage Source Inverter (Block C)

The basic form of the three phase inverter is shown

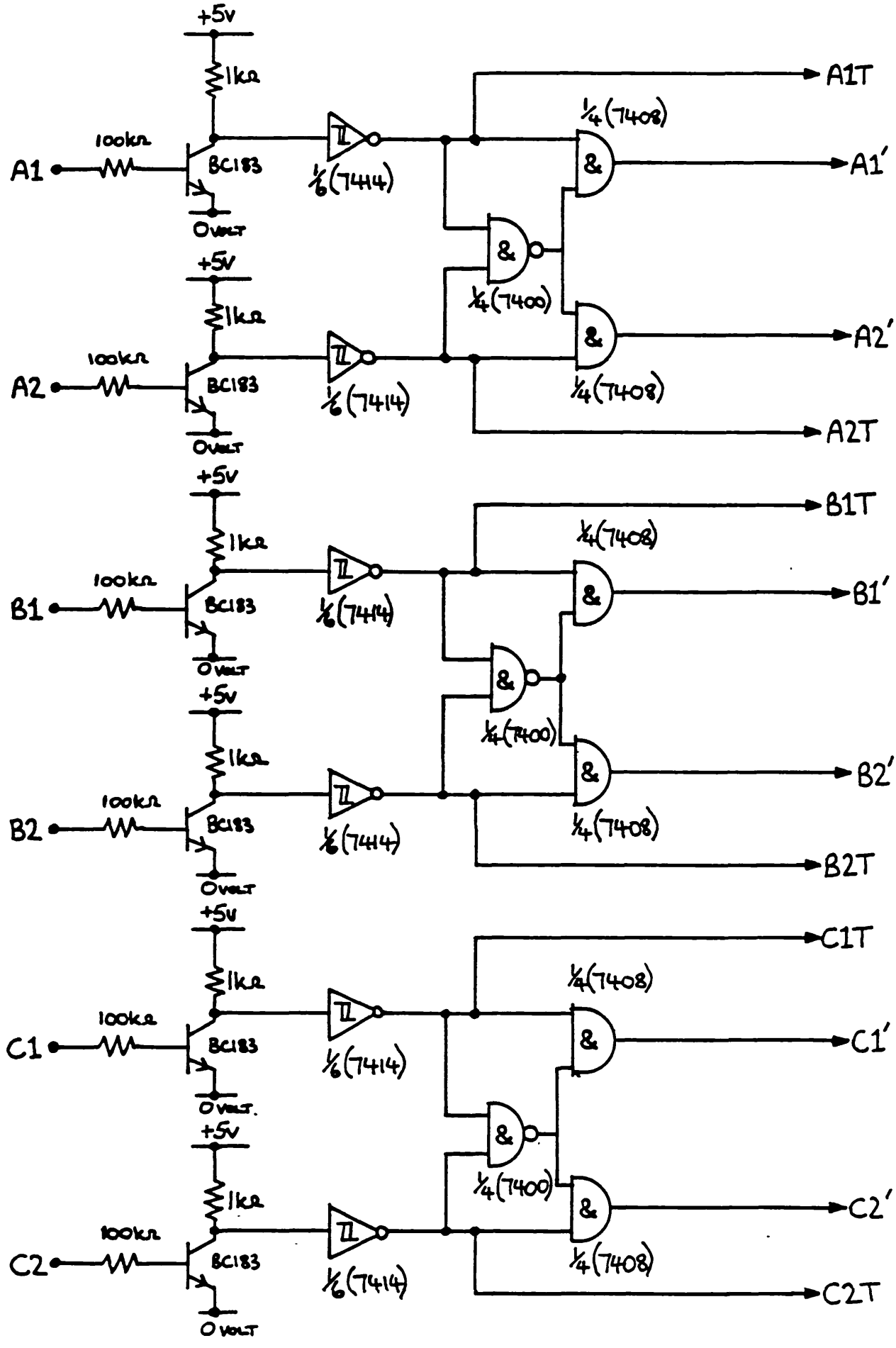


FIG. 3.30. SHAFT POSITION SENSOR SIGNAL PROCESSING LOGIC.

in fig. 3.31.

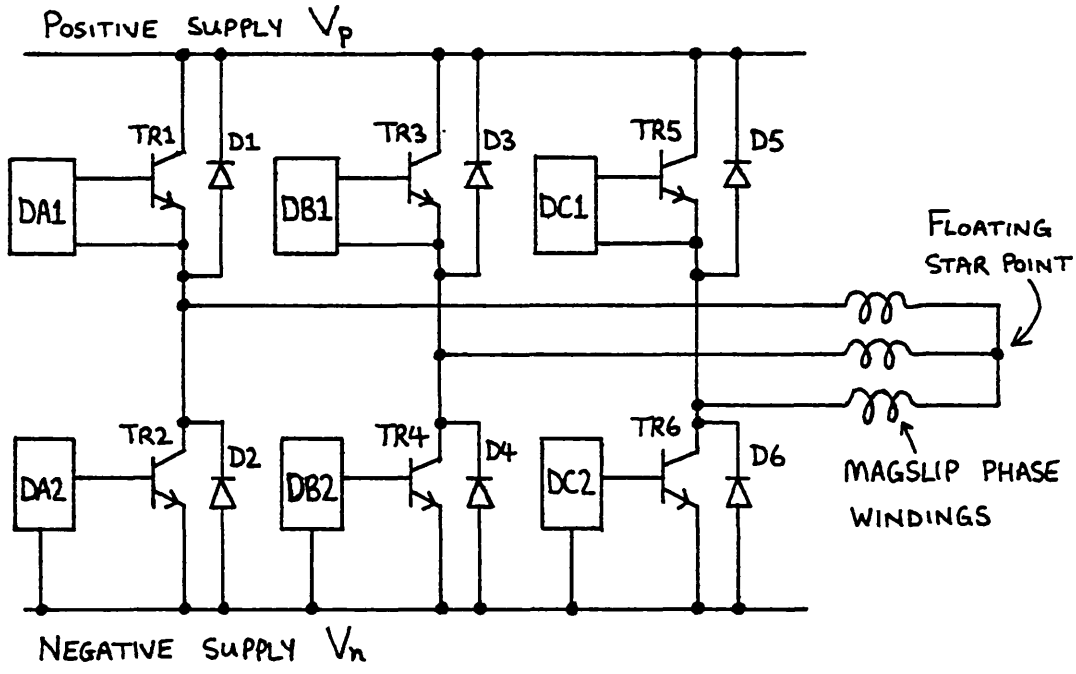


FIG. 3.31. THE BASIC FORM OF THE MAGSLIP INVERTER.

None of the emitters of the transistors TR1 to TR6 are at 0 volts potential. Hence the driver circuits (DA1, DA2, DB1, DB2, DC1, and DC2) have to level shift the transistor control signals. The driver circuits are referred to by the position sensor signals that they are activated by. For example, driver DA1 is controlled by signal A1' (produced by the circuit shown in fig. 3.30.)

The freewheel diodes (D1 to D6) must be fast acting and be rated for the motor phase current. BYW29 fast recovery diodes (35ns reverse recovery time, 150V, 7A) were used in the circuit.

The transistors connected to the positive supply rail (TR1, TR3, TR5) were driven by a different type of driver circuit than that used for the transistors (TR2, TR4, TR6)

connected to the negative supply rail. The driver circuits are described separately below.

3.5.5.1 Positive Rail Transistor Driver Circuit

Each positive rail transistor requires a totally floating driver circuit because the emitters of the transistors are not at fixed potentials. To minimise the base current requirements it was decided to use a power darlington transistor for TR1, TR3, and TR5. The TRW SVT6001 device was chosen and a data sheet is included in Appendix 3B. The SVT6001 can withstand a collector-emitter voltage of 450 volts and a collector current of 15 amps. It has a d.c. current gain of about 150 and can switch in about 1.0μs. The SVT6001 is significantly "under run" in the magslip inverter application but this increases the chances of reliable circuit operation.

The form of each SVT6001 driver circuit is shown in fig. 3.32.

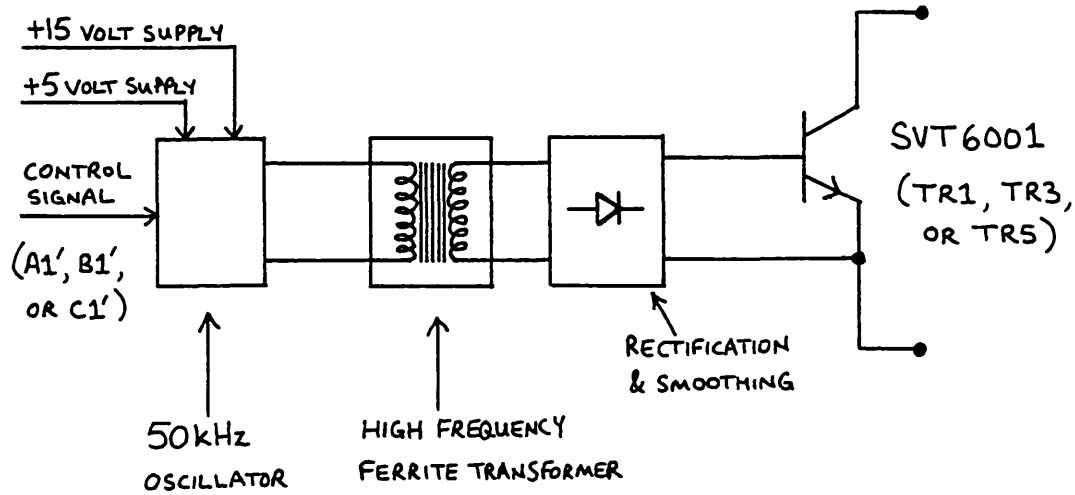


FIG. 3.32. SCHEMATIC FORM OF THE SVT6001 TRANSISTOR DRIVER CIRCUIT.

The isolation between a control signal (A1', B1', or C1') and its associated transistor (TR1, TR3, or TR5) is achieved by a transformer. The control signal is used to gate a 50 kHz square wave to a ferrite core transformer. The 50 kHz signal is a.c. coupled by the transformer and is then demodulated by a rectifier and capacitor arrangement. The "detected" signal is used to provide the base current of the SVT6001 transistor.

The base current required by an SVT6001 to support a collector current of 2.0 amps is about 14 milliamps (with a current gain of 150). The circuit shown in fig. 3.33 is capable of providing up to 50 milliamps. The basic task of the circuit is to modulate the SVT6001 switching information (i.e. the control signal) onto the 50 kHz carrier waveform. To prevent a drift towards saturation in the transformer core, the drive waveforms for transistors TT3 and TT4 have even mark/space ratios as shown in fig. 3.34. In addition, flip flops A, B, C, and D ensure that irrespective of the starting and finishing instants of the control signal, there are always an equal number of switchings by TT3 and TT4 during the control signal period. This reduces the chances of output voltage spikes at the end of a control pulse since the volt-second product is always zero. A small delay is associated with the start and finish of the train of TT3 and TT4 switching pulses as illustrated in fig. 3.34. However, for SVT6001 switching information of the order of 0Hz to 200Hz and a 50kHz carrier frequency, the delay is negligible.

The transformer was designed in a similar way to those used in the hexfet inverter circuit described in Chapter 5, and for more information the reader should refer to the relevant sections of that chapter. The smoothed secondary voltage across capacitor C_s is about 13.5 volts allowing for voltage drops across TR3, TR4, D1, D2, D3, and D4. It was decided to fix the base current I_b of the SVT6001 at about 25 milliamps. Hence R_b was selected as 470Ω. One

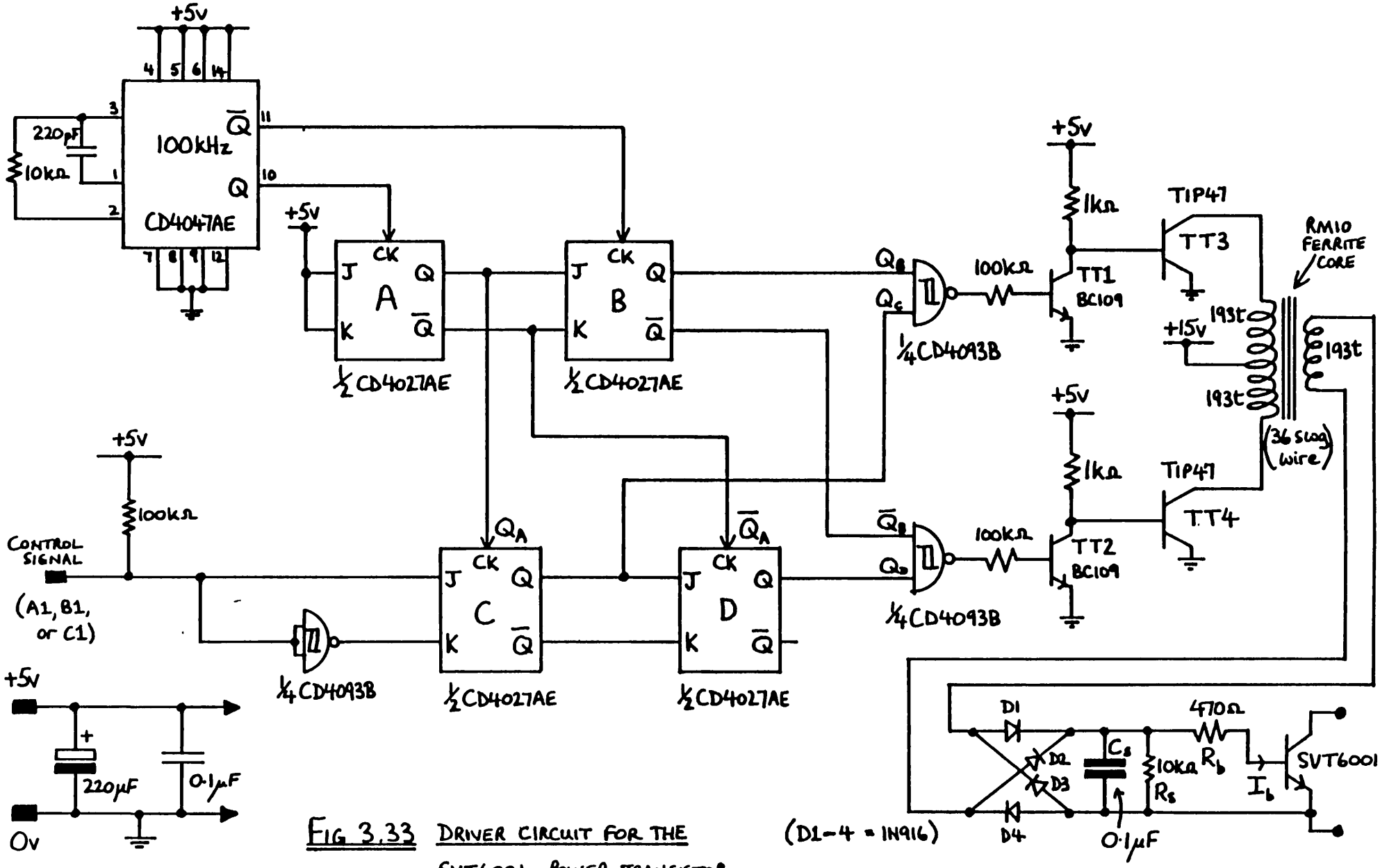


FIG 3.33 DRIVER CIRCUIT FOR THE SVT6001 POWER TRANSISTOR

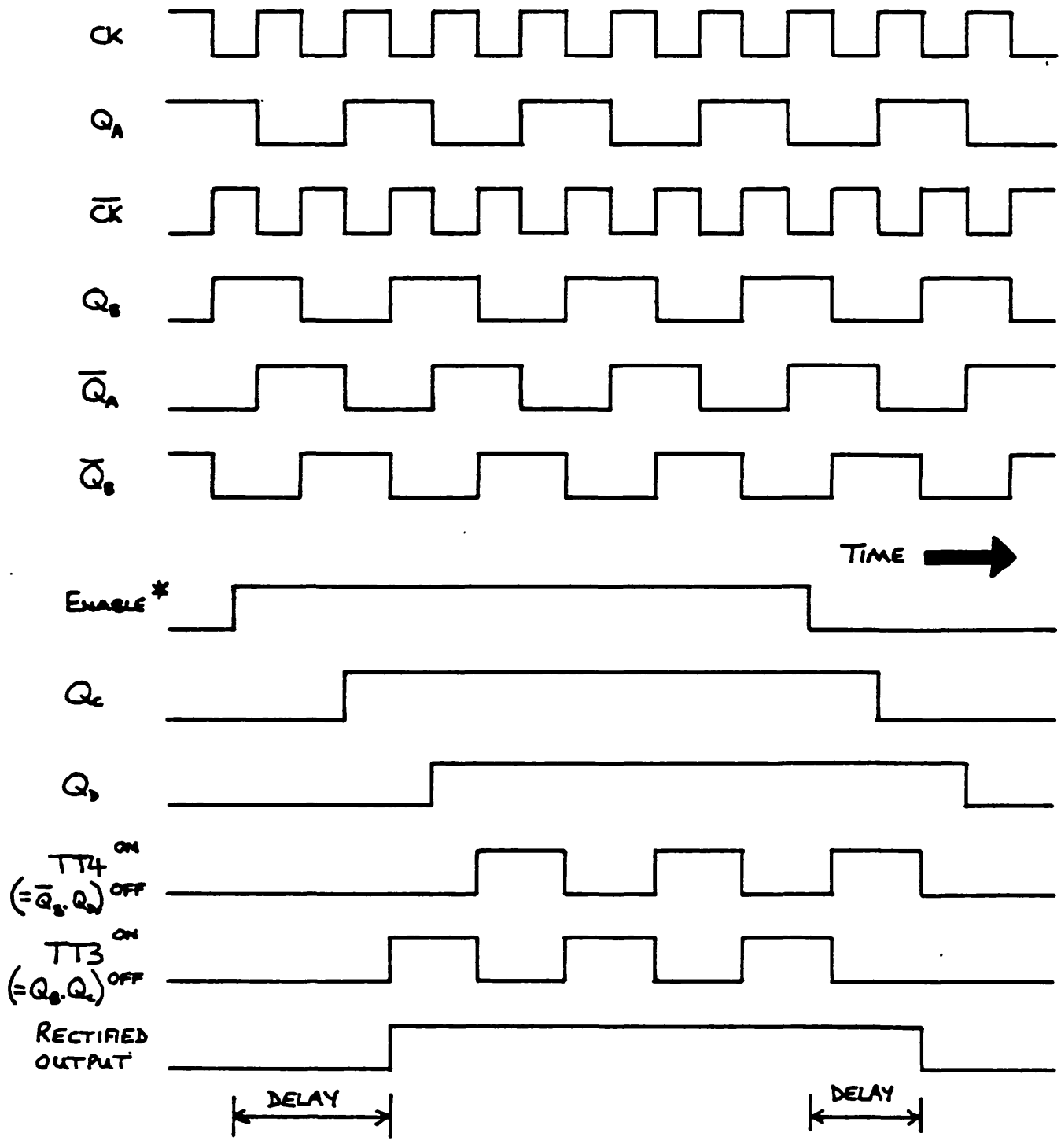


FIG 3.34 THE SWITCHING WAVEFORMS IN THE SVT6001 DRIVER CIRCUIT (FIG. 3.33).

* NOTE THE ENABLE PERIOD IS USUALLY MUCH LONGER THAN THE CK PERIOD.

point worth special mention is the choice of the time constant t_c of the smoothing capacitor C_s and the parallel combination of resistors R_s and R_b:

$$t_c = C_s.R_s.R_b./(R_s + R_b) \text{ seconds} \qquad 3.40$$

The capacitor must smooth the rectified secondary voltage sufficiently to ensure the base current of the SVT6001 is continuous. However, the voltage must decay rapidly when TT3 and TT4 are disabled. It was decided to make t_c equal to the period of three carrier cycles: i.e. $t_c = 6 \times 10^{-5}$ seconds. Then for values of $R_s = 10k\Omega$ and $R_b = 470\Omega$, equation 3.40 gives a value for C_s of 1.3×10^{-7} Farads. A value of $0.1\mu F$ was actually used.

The three transformer driver circuits required for TR1, TR3, and TR5 worked without any problems during the magslip experiments.

3.5.5.2 Negative Rail Transistor Driver Circuit

The emitters of the negative rail transistors are held at the negative power rail potential and so only move relative to the control circuitry when the supply voltage to the inverter is altered. A fully floating form of base drive circuit is not required and so the circuit shown in fig. 3.35 was developed. It was decided to use 2N3773 type devices for the negative rail power transistors (TR2, TR4, and TR6). A data sheet for the 2N3773 transistor is included in Appendix 3C. The 2N3773 can withstand a collector-emitter voltage of 140 volts and a collector current of 16 amps. It has a minimum d.c. current gain of about 15 for collector currents less than 8 amps, and it can switch in a time of about $1\mu s$ to $1.5\mu s$.

The function of the circuit shown in fig. 3.35 is to generate a constant current whenever the control signal (A2', B2', or C2') is active (i.e. at TTL logic 1 = 3.8V).

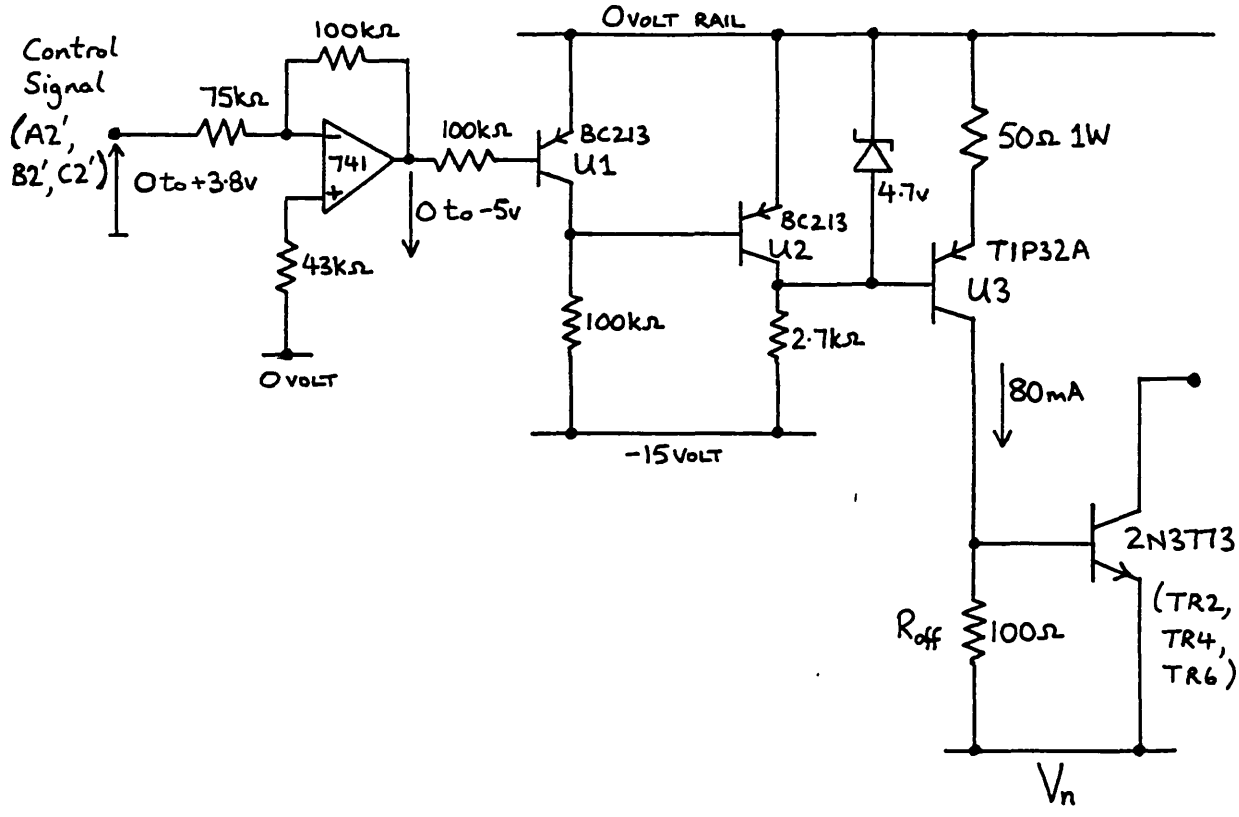


FIG. 3.35. THE DRIVER CIRCUIT FOR THE 2N3773 TRANSISTORS.

The control signal is level shifted by an inverting operational amplifier (type 741) so that it is suitable for controlling the constant current circuit. The constant current is fed to the base of the 2N3773 transistor and causes it to switch on. The constant current is essentially independent of the potential at the emitter of the 2N3773, although if the potential V_n is too small the constant current generator ceases to function. Therefore, the base current of the 2N3773 can be defined for a wide range of values of V_n up to -50 volts. If the control signal is removed (i.e. set to 0 volts) the constant current ceases and the 100Ω resistor (R_{off}) holds the 2N3773 transistor in an off state. It was found experimentally that a constant current value of 80 milliamps was more than sufficient

to maintain the 2N3773 transistors used in the experiment in a saturated state when supporting a 2 amp collector current. Hence the circuit shown in fig. 3.35 was arranged to provide an 80 milliamp current output. The TIP32A transistor (U3) was mounted on a small heatsink to dissipate the 3.6 watts that are generated when V_n = -50 volts and the constant current flows continuously.

The three constant current circuits required for TR2, TR4, and TR6 worked without any problems during the magslip experiments.

3.5.5.3 Snubber Protection for the Power Transistors

When a transistor inverter is used to drive an inductive load, it is generally necessary to provide some snubber protection to prevent damage occurring to the power transistors. A transistor switch dissipates very little power when in an on or off state, since in the former state the voltage drop across the device is very small, whilst in the latter case the current carried by the device is close to zero. However, during the time that a transistor is switching from one state to the other it carries some current and has a potential difference across it. Hence significant power can be dissipated, especially if the device switches many times per second.

A "dv/dt" snubber is connected in parallel with a transistor to limit the rise of voltage across the transistor as it turns off. This reduces the power dissipation within the device because the collector current can fall to zero before the voltage rises to any significant value. A dv/dt snubber also reduces the chances of avalanche breakdown within the transistor brought about by excessive rates of change of voltage.

A "di/dt" snubber is connected in series with a transister to limit the rate of rise of current in the trans-

ister as it turns on. This reduces the power dissipation because the voltage across the device can fall to zero before a significant current begins to flow. In addition, a di/dt snubber prevents "hot spots" in the semiconductor material which arise due to uneven current distribution as the device starts to conduct. Hot spots can lead to device breakdown.

Much has been written on the application and design of snubbers (3.10; 3.11; 3.12; 3.13) and little purpose would be served by covering the topic in any further detail here. The snubber networks connected to the magslip inverter transistors are shown in fig. 3.36 (one phase only).

The values for the snubber components are calculated in Appendix 3A. It should be noted that the snubber component values are very approximate. This is mainly because the transistor switching times used in Appendix 3A are the typical times that can be achieved with good base drive circuitry. The magslip inverter base drive circuits may possibly cause the SVT6001 and 2N3773 transistors to switch in the specified times, but it is more likely that the actual switching times are much longer. However, this was not pursued because the main aim of the work was to obtain some magslip characteristics and to try out various position sensors. In practice, the inverter was found to be very reliable.

A few comments on the snubber components are appropriate at this point. The 36μH inductors were wound on RM10 ferrite cores which do not saturate when carrying 2 amp collector currents. The diodes (D_i and D_v) were fast recovery BYW29 devices (150 volts, 7 amps, 35ns reverse recovery time). The resistors were low inductance carbon composition types. Polypropylene low loss capacitors rated at 1250 volts d.c. were used for the snubber capacitors. The connecting leads between the snubber components and the transistors were kept as short as possible to minimise

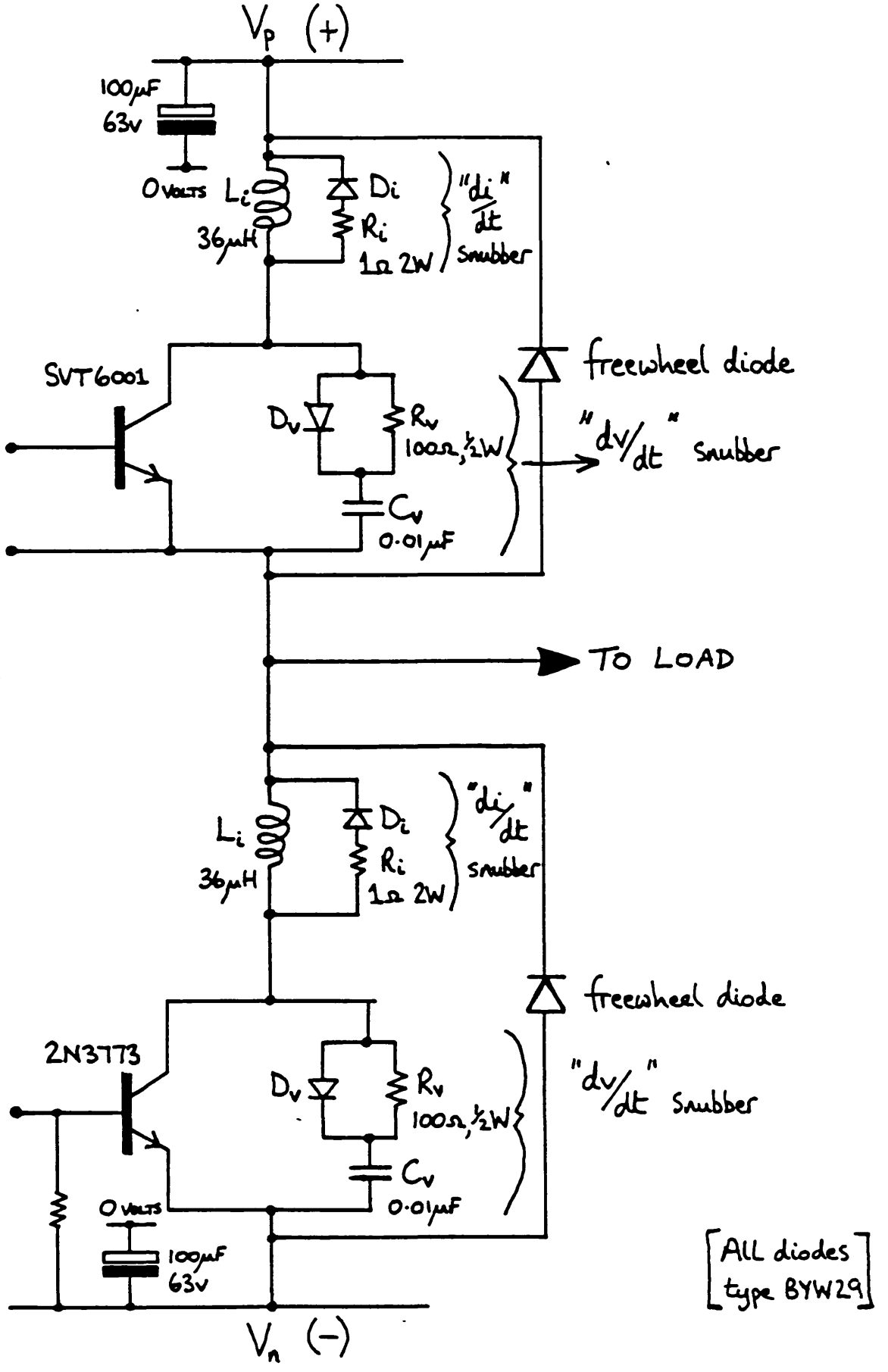


FIG. 3.36. THE TYPICAL FORM OF THE MAGSLIP INVERTER OUTPUT STAGES, (SHOWING THE SNUBBERS).

stray inductance.

3.5.5.4 Inverter Assembly

The SVT6001 and 2N3773 power transistors were mounted on individual 19° C/W TO3 heatsinks. 100μF 63 volt electrolytic capacitors were connected to the positive and negative voltage supply rails next to each power transistor as indicated on fig. 3.36. The decoupling provided by the electrolytic capacitors reduces voltage spikes on the voltage rails caused by the supply line inductance. The various circuit elements of the inverter were arranged as shown schematically in fig. 3.37(a). A photograph of the inverter is shown in fig. 3.37(b).

3.5.6 Tachometer Signal Generator (Block G)

In order to use the digital tachometer described in Chapter 8, it was necessary to derive a suitable digital speed signal from the shaft position sensor signals. The digital tachometer requires six equi-spaced pulses per revolution in order to give a correct speed reading. The circuit shown in fig. 3.38 generates the required number of pulses from the available shaft position signals. The input signals (A1T, A2T, B1T, B2T, C1T, and C2T) are supplied by the circuit shown in fig. 3.30. The rising edges of these signals occur at intervals of one-sixth of a revolution. Each time a rising edge occurs, one of the flip flops (74109 type) is set and triggers the monostable (74121 type). The monostable provides a 0.33ms output pulse to the tachometer and at the same time clears all the flip flops (even though only one needs clearing). The period of the monostable pulse was chosen so that the circuit can operate reliably up to 12000 rpm. The 7400 TTL gate and associated 10kΩ/0.47μF components ensure that when the circuit is initially switched on, all of the flip flops are cleared if necessary. The circuit was assembled on the same prototype board as that used for the optical sensor

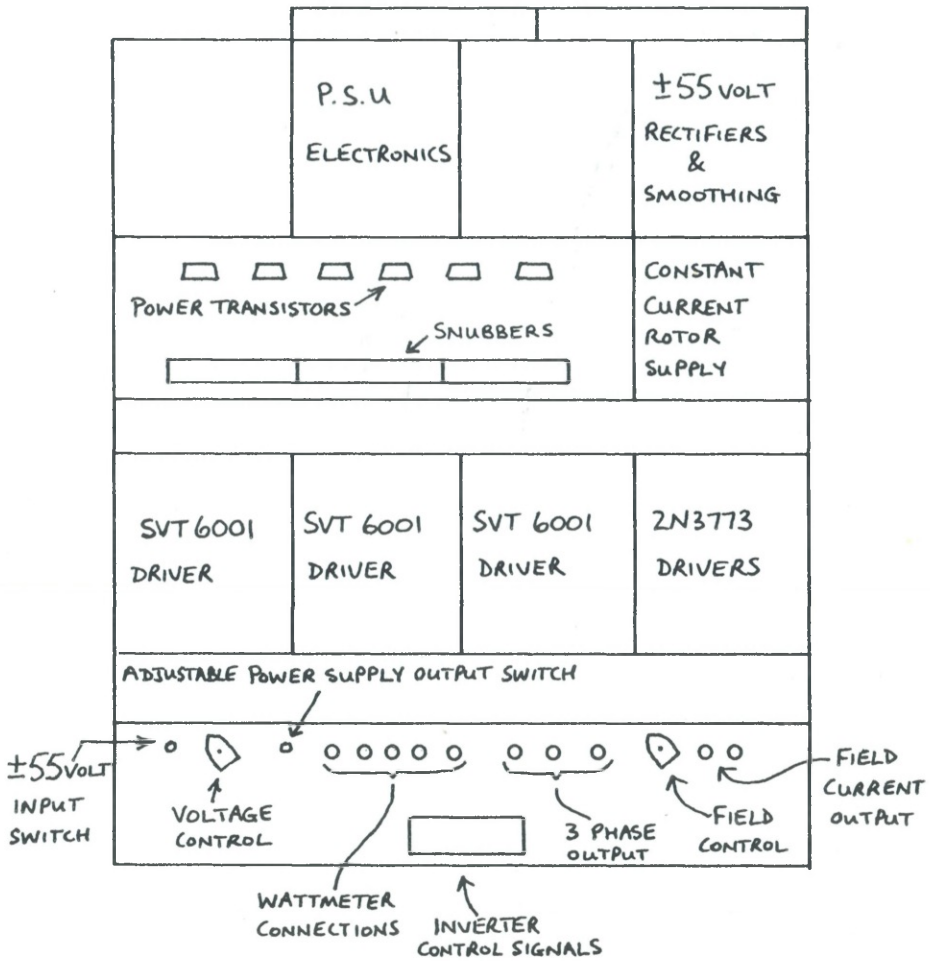


FIG. 3.37(a). THE LAYOUT OF THE MAGSLIP INVERTER COMPONENTS.

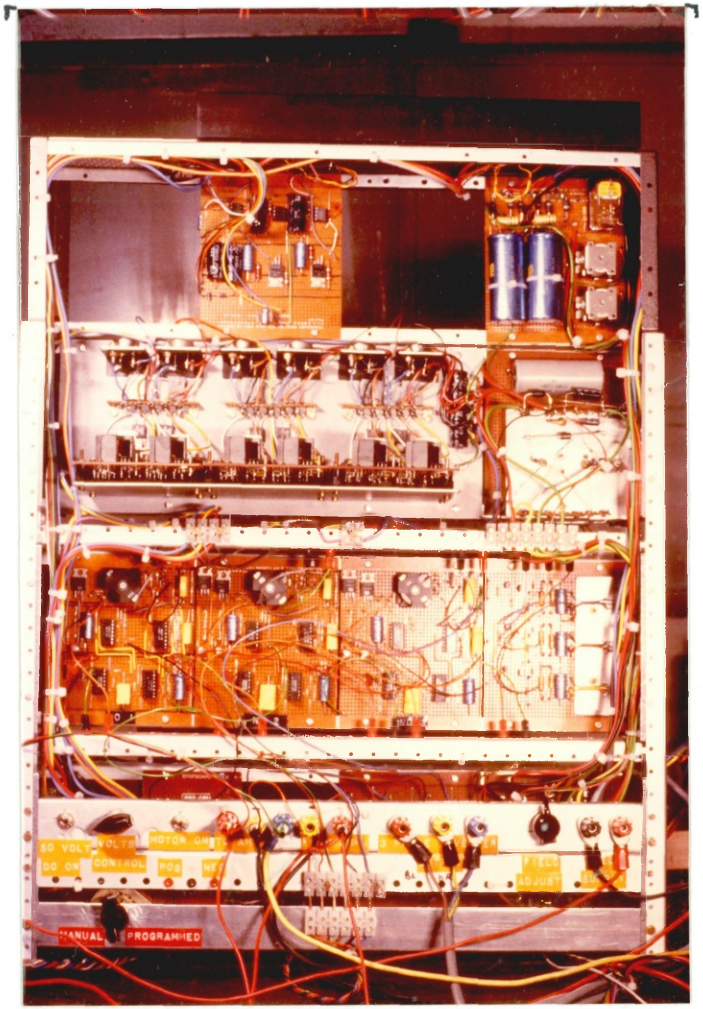


FIG. 3.37(b) PHOTOGRAPH OF THE MAGSLIP INVERTER.

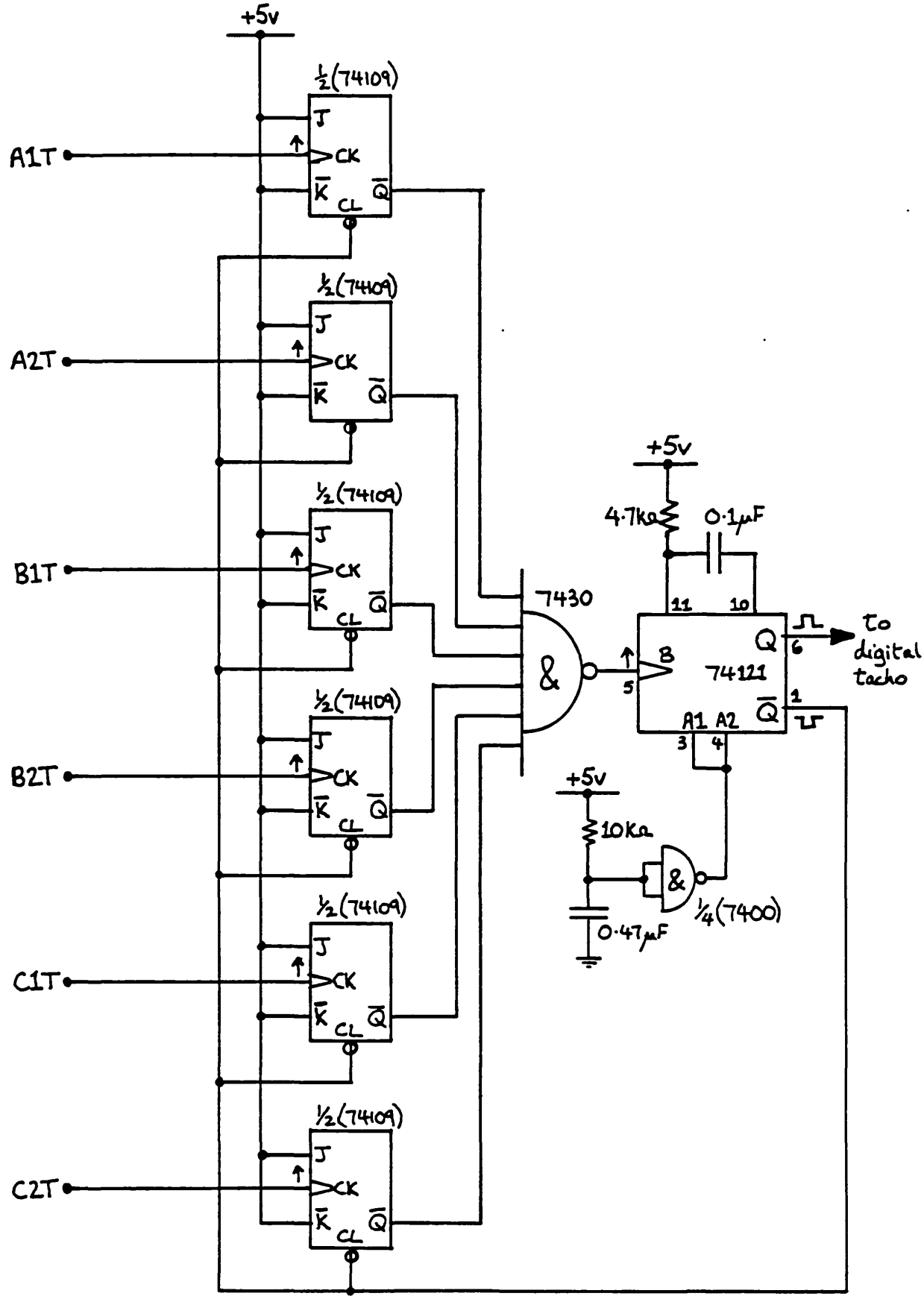


FIG. 3.38. CIRCUIT TO INTERFACE THE MAGSLIP SHAFT POSITION SENSOR TO THE DIGITAL TACHOMETER.

circuit shown in fig. 3.30.

3.5.7 Magslip Field Current Source (Block H)

The magslip field current of 0.7 amps was generated by a simple constant current circuit. A constant current source was found to be necessary to overcome the effects of induced voltages in the field winding due to stator currents. The circuit shown in fig. 3.39 allowed the field current to be varied over the range 0 to about 1.1 amps. A transistorised relay was incorporated to allow the field current to be under logic control for reversing purposes. The large electrolytic capacitor helped to damp out variations in the field current. The 4.4μF non-polarised capacitor was included to protect the relay contacts when they change states. The 2N3055 power transistor was mounted on a small heatsink to dissipate the 6.4W of power that is generated for a constant current value of 0.7 amps.

The circuit was mounted on the inverter assembly as indicated in fig. 3.37(a).

3.5.8 Operating Comments

The magslip was mounted on a dural baseplate as shown in fig. 3.40. The shaft position detector was housed in an aluminium box to ensure that if the perspex disc shattered at high speed it would not be dangerous. The magslip and shaft position detector were coupled up to a MAGTROL HD500-8 hysteresis brake dynamometer. This dynamometer can measure torques up to 850mNm and operate at speeds up to 20000 rpm.

The adjustment of load angle on the system was achieved by rotating the magslip body on its supports. The movement of the magslip with respect to the shaft position sensors causes a load angle change. (It had been decided to rotate the motor rather than the position sensors be-

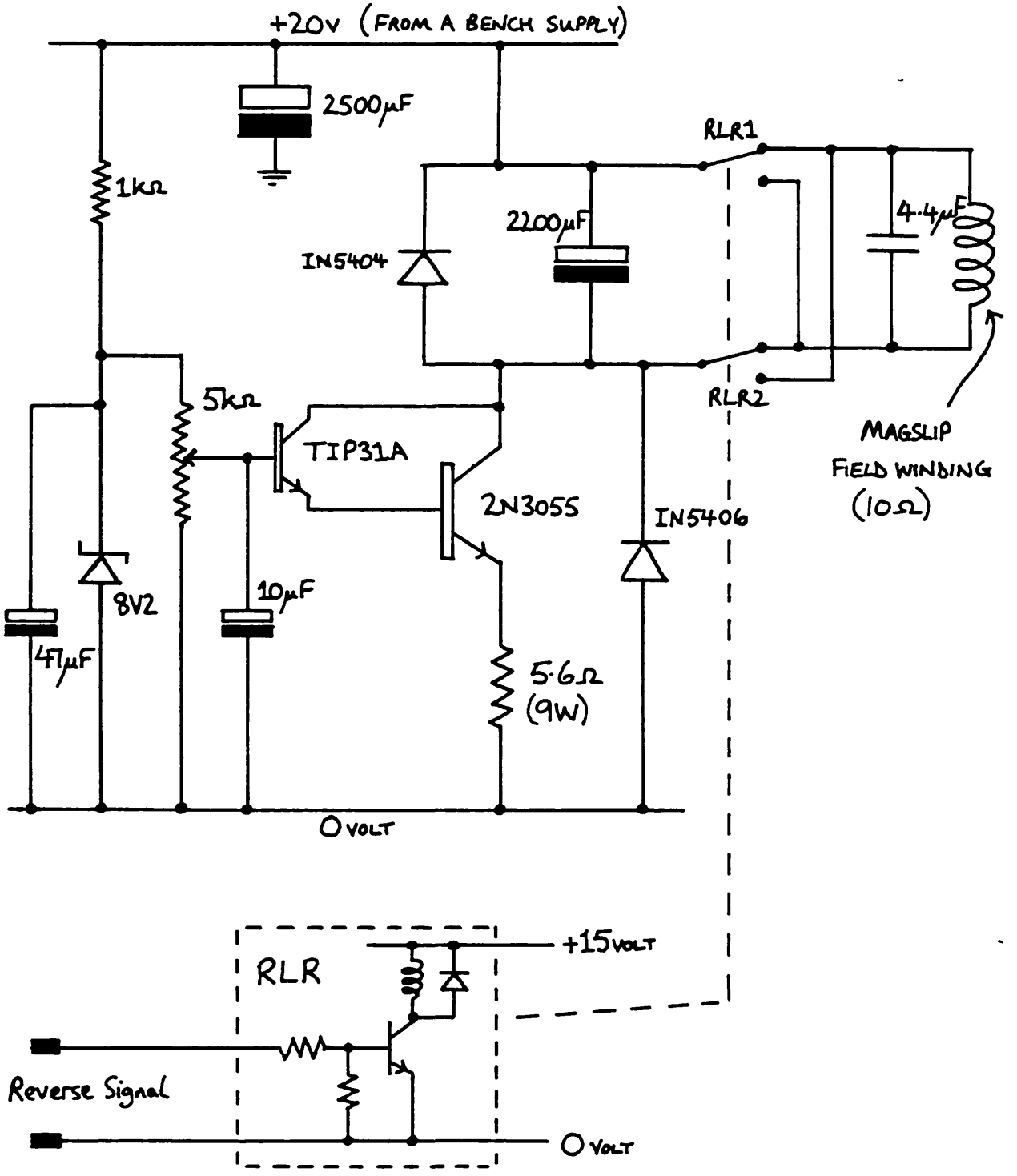


FIG. 3.39 THE MAGSLIP FIELD WINDING CURRENT SOURCE.

cause it was easier in the case of the magslip system). The motor body was marked at appropriate points so that load angles of 0^{o}, 30^{o}, 60^{o}, and 90^{o} could be set up easily during motor tests.

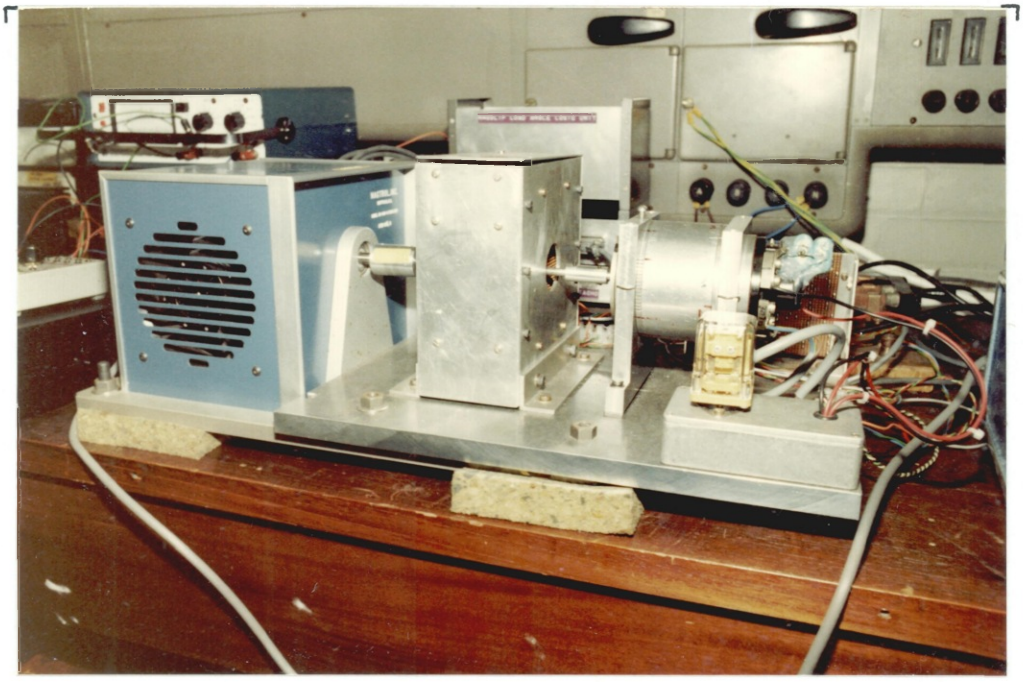


FIG. 3.40. PHOTOGRAPH OF THE MAGSLIP TEST RIG SHOWING THE MAGSLIP, THE SHAFT POSITION SENSOR, AND THE MAGTROL DYNAMOMETER.

3.6 Magslip Operating Characteristics

The magslip system was run with up to ± 50 volts d.c. supplied to the inverter. The motor was started with about ± 5 volts supplied to the motor and with a load angle of 0^{o}. The motor started reliably and would run in either direction, depending on the field current polarity. Once the motor had started the inverter supply voltage was increased up to the required test value and the load angle adjusted if necessary. The current and voltage measurements were made at the points indicated on fig. 3.41.

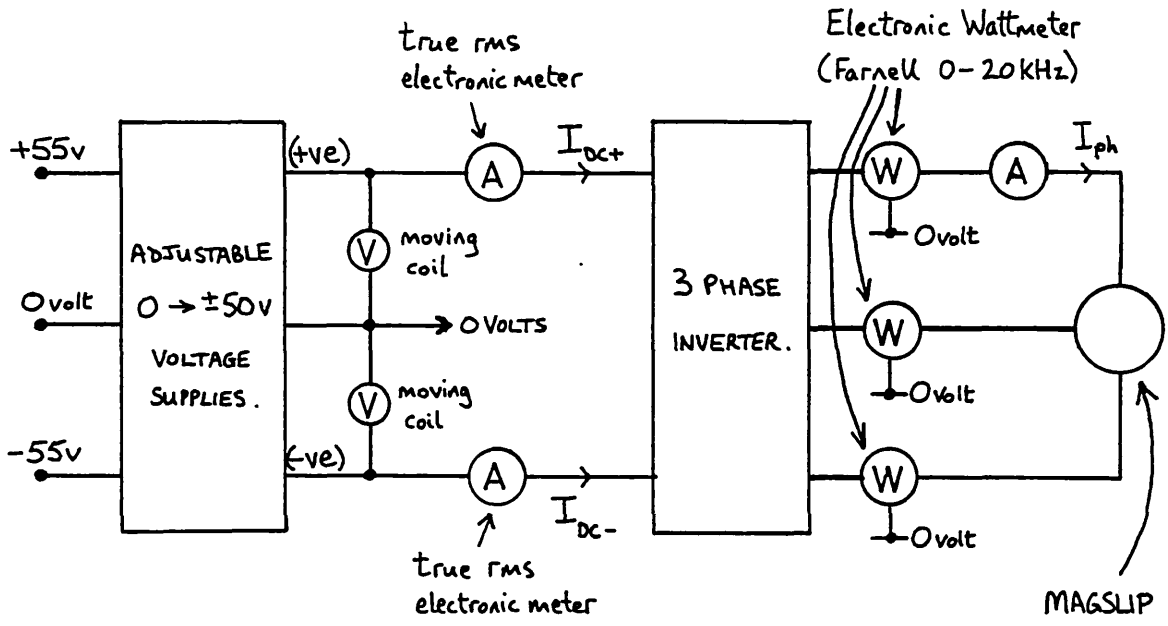


FIG. 3.41. THE MAGSLIP TEST CIRCUIT SHOWING THE INSTRUMENTATION USED.

A set of torque/speed curves were obtained for the magslip system operating from a ±50 volt d.c. supply. Fig. 3.42 shows the measured torque speed curves for load angles of 0^{o}, 30^{o}, 60^{o}, and 90^{o}. The no load speeds were 4300 rpm, 5400 rpm, 9600 rpm, and 11500 rpm respectively. This confirms the predicted characteristics in Chapter 2, which suggest that the speed of an autopiloted motor is dependent on the operating load angle.

The measured torque, phase current, and motor efficiency and the corresponding predicted values are plotted against speed for load angles of 0^{o}, 30^{o}, 60^{o}, 90^{o} in figs. 3.43, 3.44, 3.45, and 3.46 respectively. Some explanation is warranted in respect of the phase voltage used for the predictions, and the correction made to the measured torque values to take account of friction and windage.

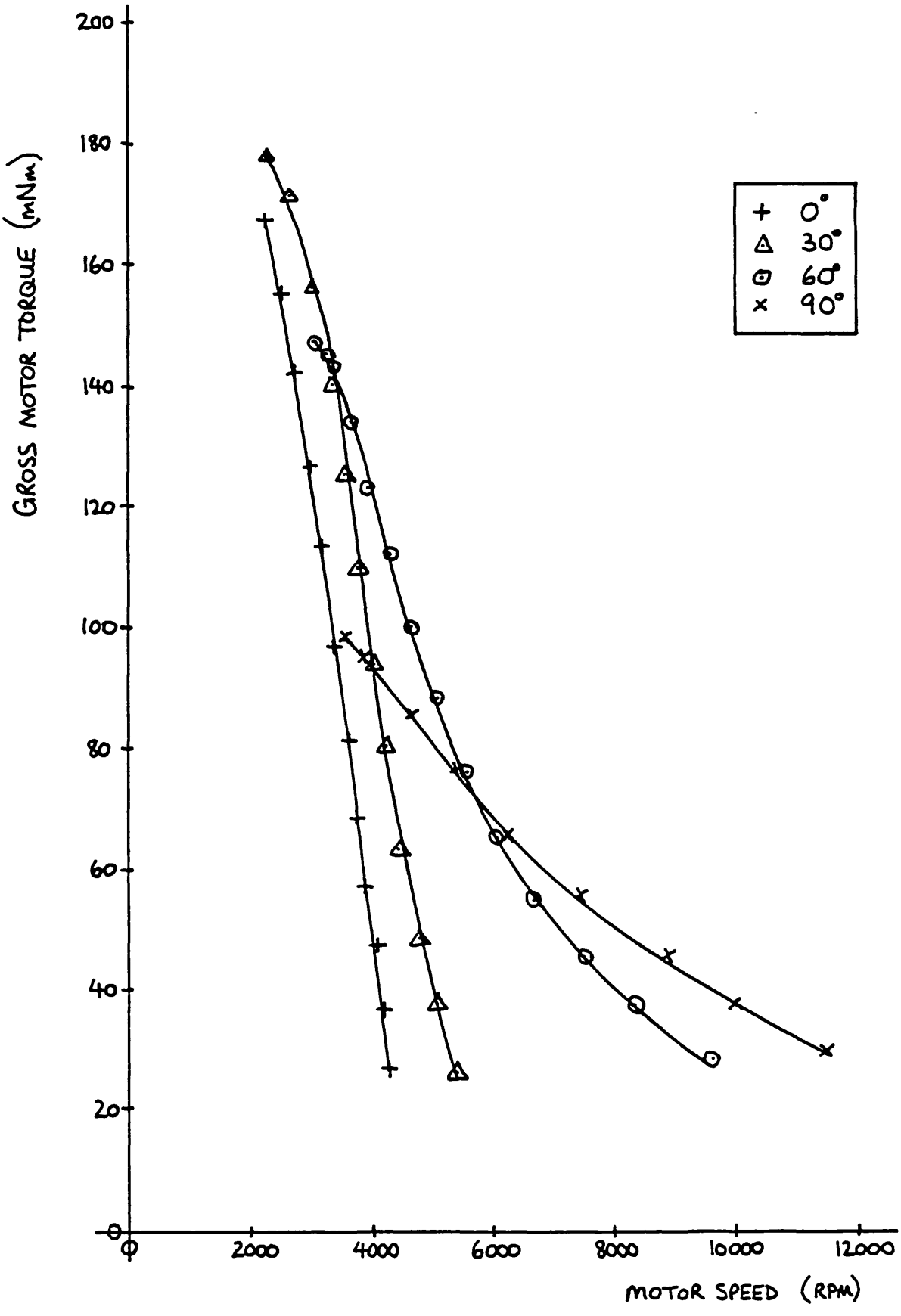


FIG. 3.42. MEASURED TORQUE / SPEED CURVES FOR THE MAGSLIP OPERATED AT ±50 VOLTS D.C. AND WITH VARIOUS LOAD ANGLES.

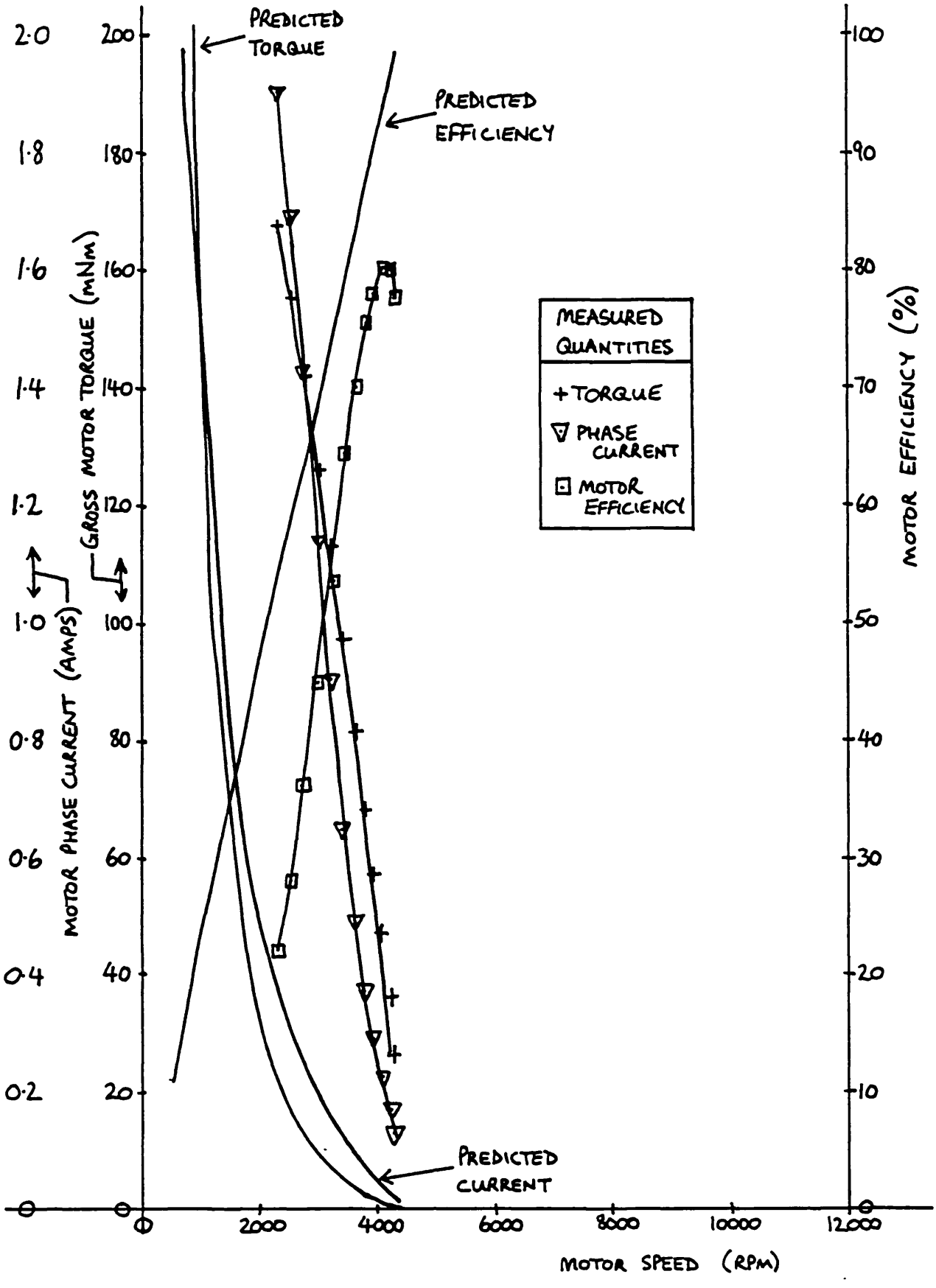


FIG. 3.43. GRAPH SHOWING PREDICTED AND MEASURED GROSS TORQUE, PHASE CURRENT AND MOTOR EFFICIENCY FOR THE MAGSLIP SYSTEM RUN FROM ±50 VOLTS D.C. AT A LOAD ANGLE OF 0°.

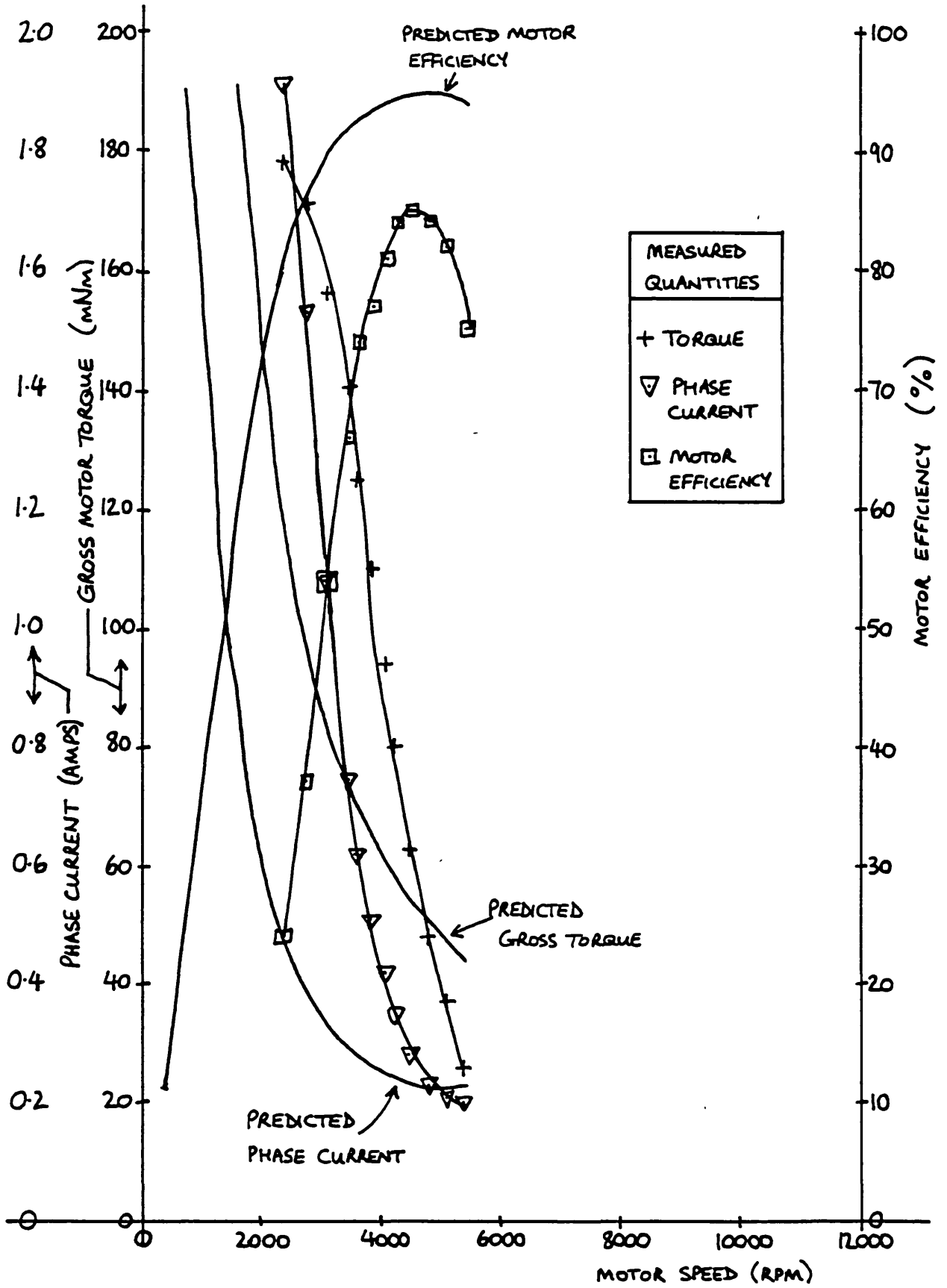


FIG. 3.44. GRAPH SHOWING PREDICTED AND MEASURED GROSS TORQUE, PHASE CURRENT AND MOTOR EFFICIENCY FOR THE MAGSLIP SYSTEM RUN FROM ±50 VOLTS D.C. AT A LOAD ANGLE OF 30°.

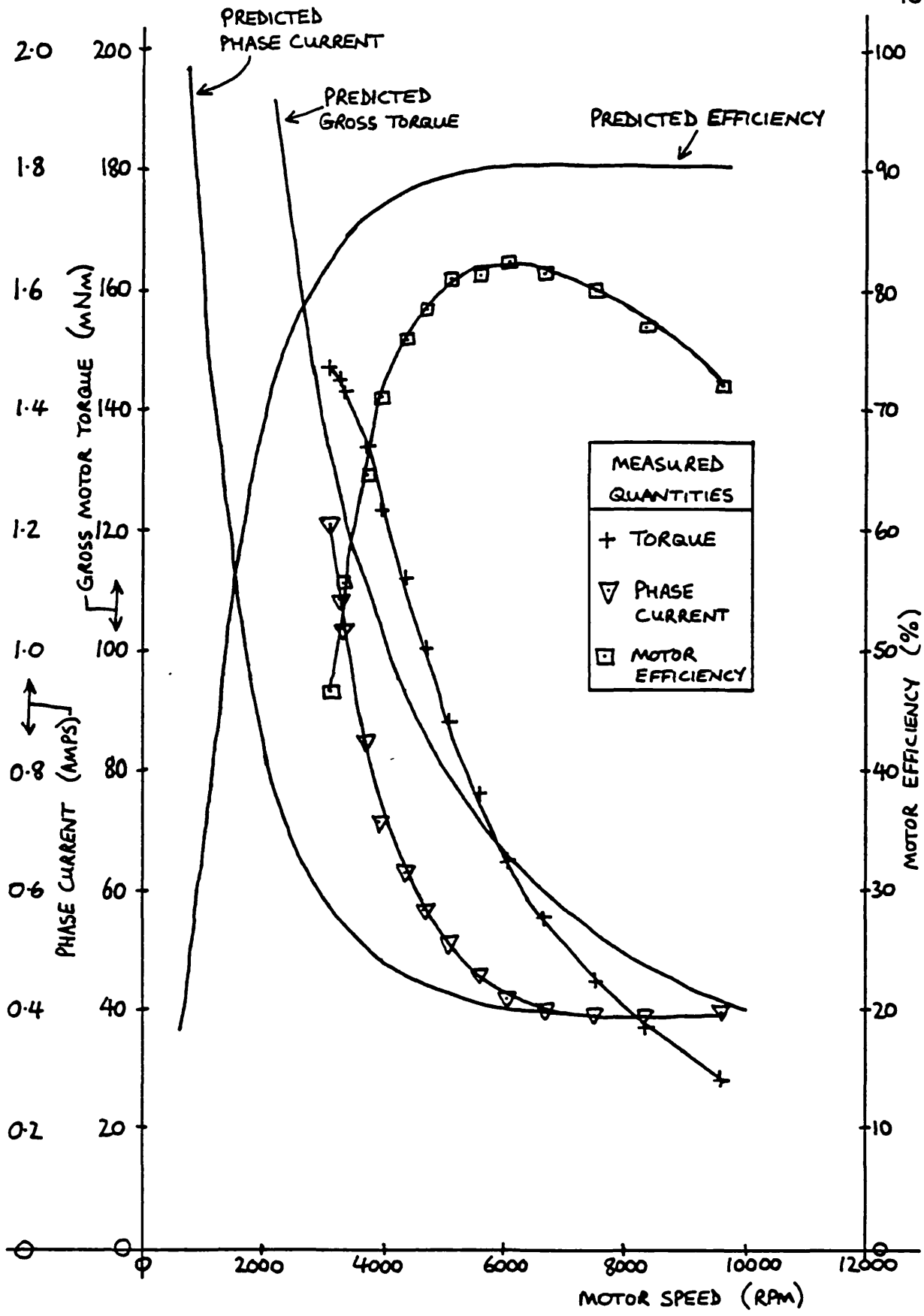


FIG. 3.45. GRAPH SHOWING PREDICTED AND MEASURED GROSS TORQUE, PHASE CURRENT AND MOTOR EFFICIENCY FOR THE MAGSLIP SYSTEM RUN FROM ±50 VOLTS D.C. AT A LOAD ANGLE OF 60°.

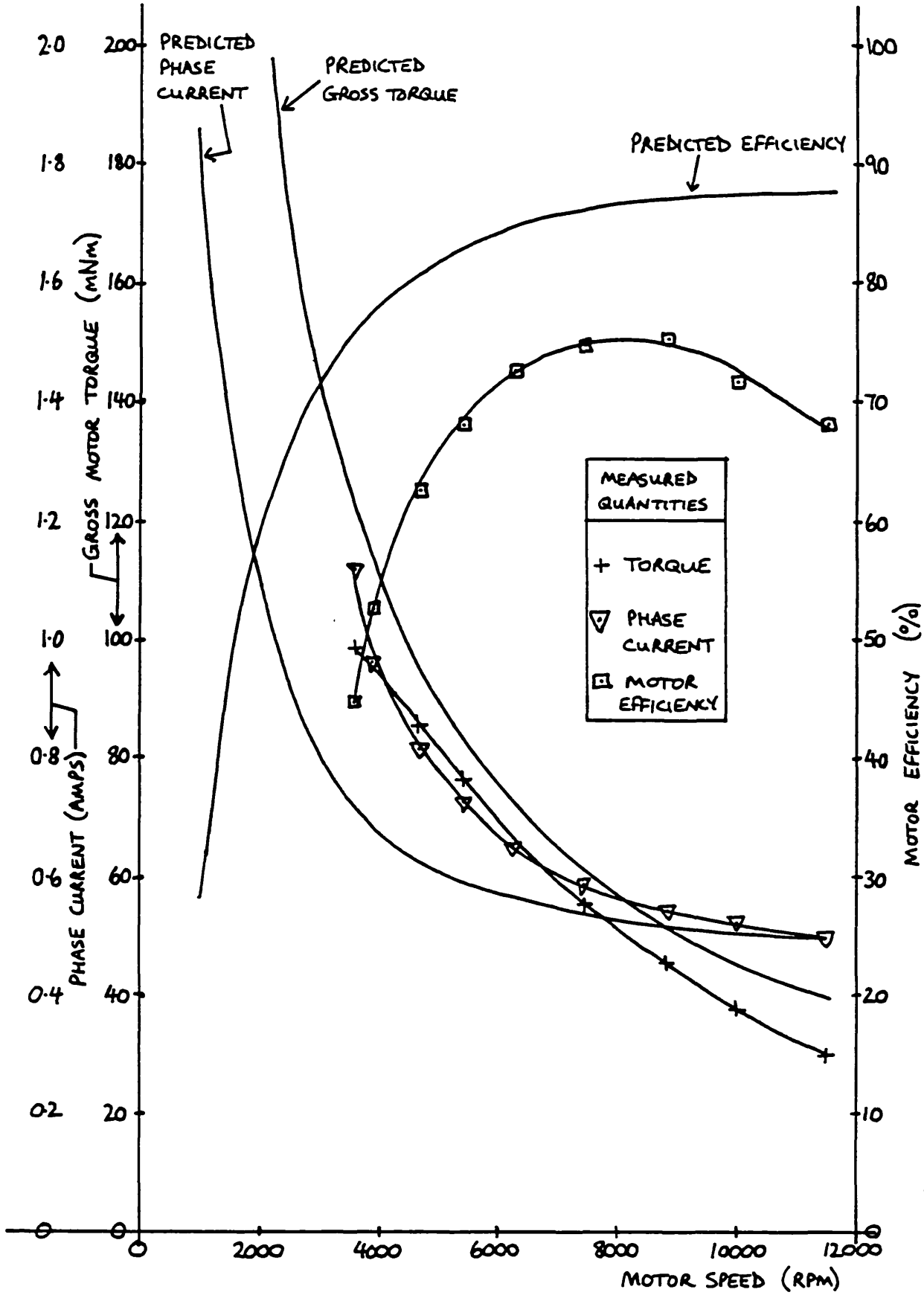


FIG. 3.46. GRAPH SHOWING PREDICTED AND MEASURED GROSS TORQUE, PHASE CURRENT AND MOTOR EFFICIENCY FOR THE MAGSLIP SYSTEM RUN FROM ±50 VOLTS D.C. AT A LOAD ANGLE OF 90°.

(a) Applied Phase Voltage Used in Predictions

By using Fourier analysis it can be shown that the fundamental component of a quasi-square 120^{o}- 60^{o} waveform has an rms magnitude V_{fund} given by:

$$V_{fund} = 0.78.V_{dc} \qquad 3.41$$

where V_{dc} is the magnitude of the quasi-square waveform. Therefore, if the supply voltage to the inverter is ±50 volts, the output voltage has an rms magnitude of 39.0 volts. This value was used in the equations given in Chapter 2 to obtain the predicted magslip results.

(b) Friction and Windage Torque Correction

The torque measured by the dynamometer does not include the friction and windage torques in the magslip or shaft position encoder. The gross magslip torque is therefore the measured torque plus the friction and windage torques. A method of determining the friction and windage torque over the motor speed range is described in Chapter 9, and it involves the measurement of a speed-time run-down characteristic. This characteristic was obtained for the magslip over a speed range from 12000 rpm down to zero. The curve was virtually linear (due to the dominating effect of the field winding brush friction) and the average slope from several experiments was -76.79 radians per second. The moment of inertia for the whole magslip rotating assembly (including position sensor and dynamometer) was found to be a total of 2.60×10^{-4} $Kg.m^{2}$ by a combination of measurements and calculations. The friction and windage torque T_{loss} required to produce a constant deceleration 'a' of a moment of inertia 'I' is given by:

$$T_{loss} = I.a \qquad 3.42$$

and hence T_{loss} is found to be 20mNm for the magslip system.

Therefore 20mNm was added to all the measured torque values and the curves shown in figs. 3.43 to 3.46 have been corrected in this manner.

The measured and predicted characteristics match remarkably well when it is considered that the predictions are for sinusoidal excitation conditions. The main differences between the measured and predicted characteristics are a direct result of the absence of iron loss and harmonic torque terms in the sinusoidal analysis given in Chapter 2. The absence of iron loss in the predicted curves can be seen clearly in the predicted efficiency curves for load angles of 60^{o} and 90^{o}. At high speeds the efficiency should fall because the iron loss rises with supply frequency, but this does not occur for 60^{o} or 90^{o}. However, the predicted and measured efficiencies are generally within about 10% of each other and so the predictions are useful.

The rather large discrepancy between the predicted and measured torque for 0^{o} is probably due to the problem of accurately setting up a 0^{o} load angle. A small error of only a few degrees can lead to a dramatic change in motor characteristic and the error seems to have greater consequences at small load angles than at large load angles.

Some researchers (3.14) have reported good agreement between measured quasi-square results and sinusoidal predictions but their experiments have been with larger machines than the magslip. It is therefore concluded that the sinusoidal prediction method is probably more accurate with "large" machine systems.

Typical applied phase voltage and phase current waveforms for load angles of 0^{o} and 60^{o} are shown in figs. 3.47, 3.48, 3.49, and 3.50. The current waveforms were obtained using a Tektronix current probe. The applied phase voltages were measured with respect to the zero volt rail of the system. Fig. 3.47 shows the waveforms for 0^{o}

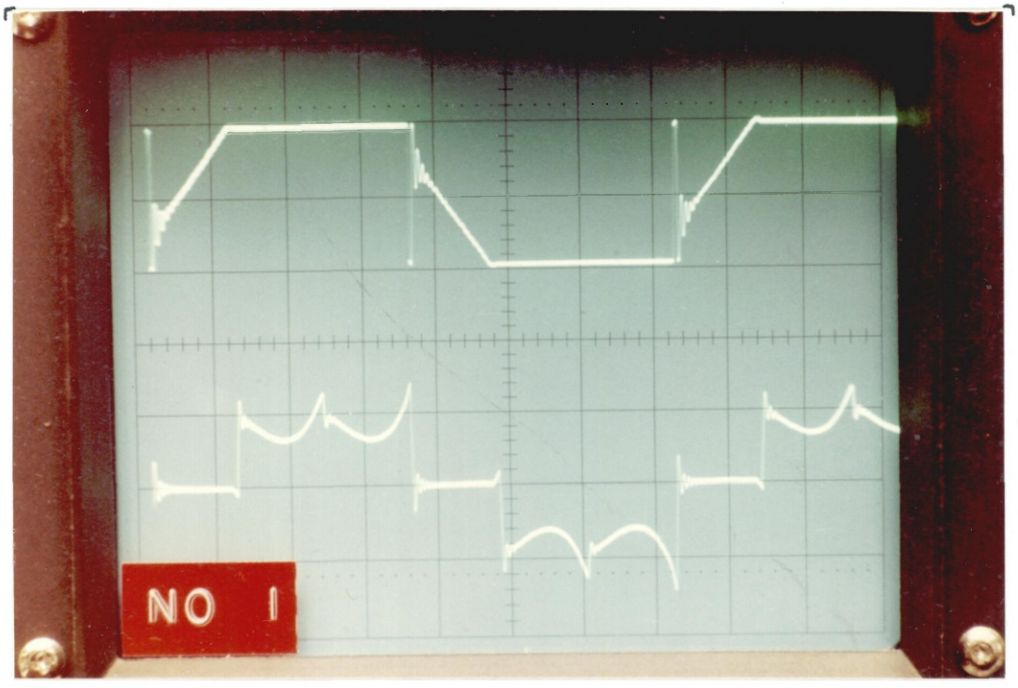


FIG. 3.47 0° LOAD ANGLE – MOTOR UNLOADED ; 4260 rpm.
(TOP ; PHASE VOLTS – 50V/cm : BOTTOM ; PHASE CURRENT – 0·2A/cm
TIMEBASE : 2ms/cm)

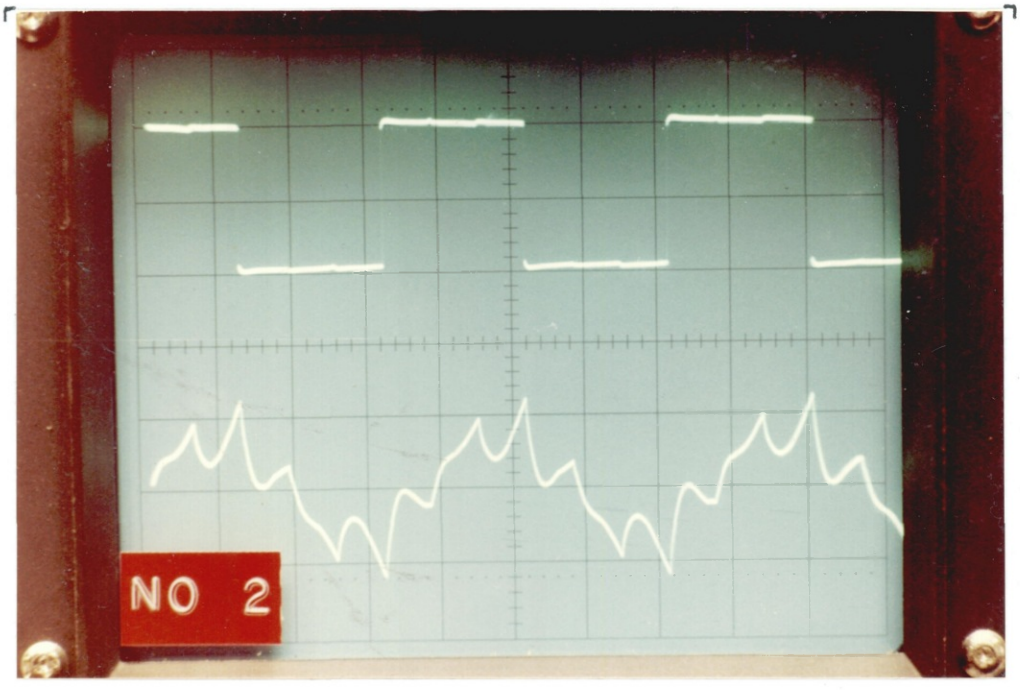


FIG. 3.48 0° LOAD ANGLE – 100mNm LOAD ; 3120 rpm.
(TOP ; PHASE VOLTS – 50V/cm : BOTTOM ; PHASE CURRENT – 2A/cm
TIMEBASE ; 5ms/cm)

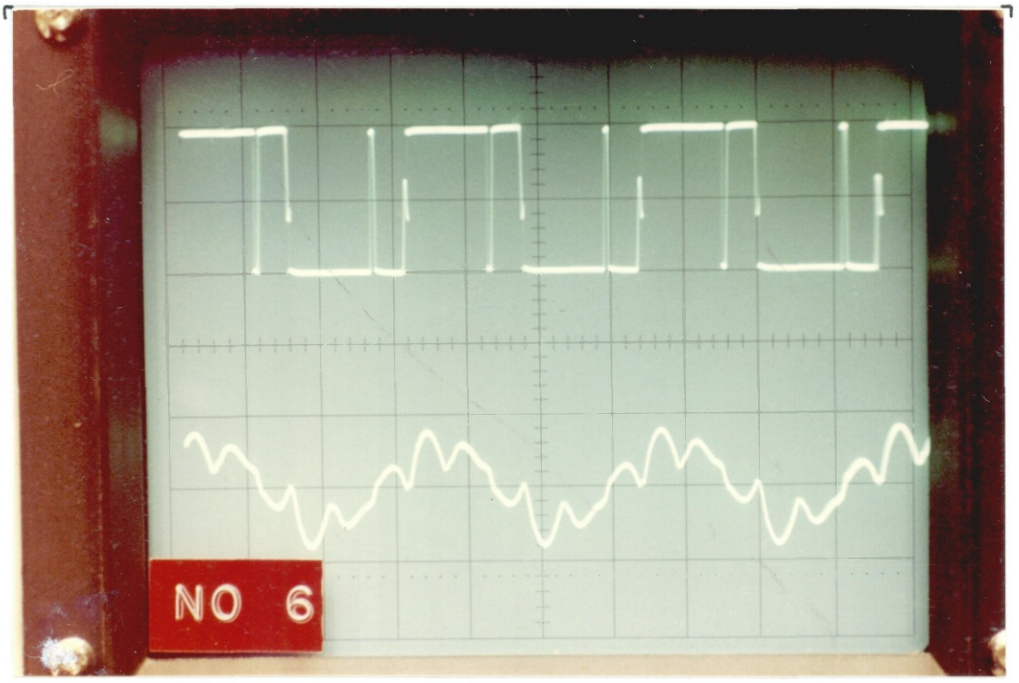


FIG. 3.49. 60° LOAD ANGLE – MOTOR UNLOADED ; 9444 rpm.
(TOP ; PHASE VOLTS – 50V/cm : BOTTOM ; PHASE CURRENT – 1·0 A/cm : TIMEBASE ; 2ms/cm)

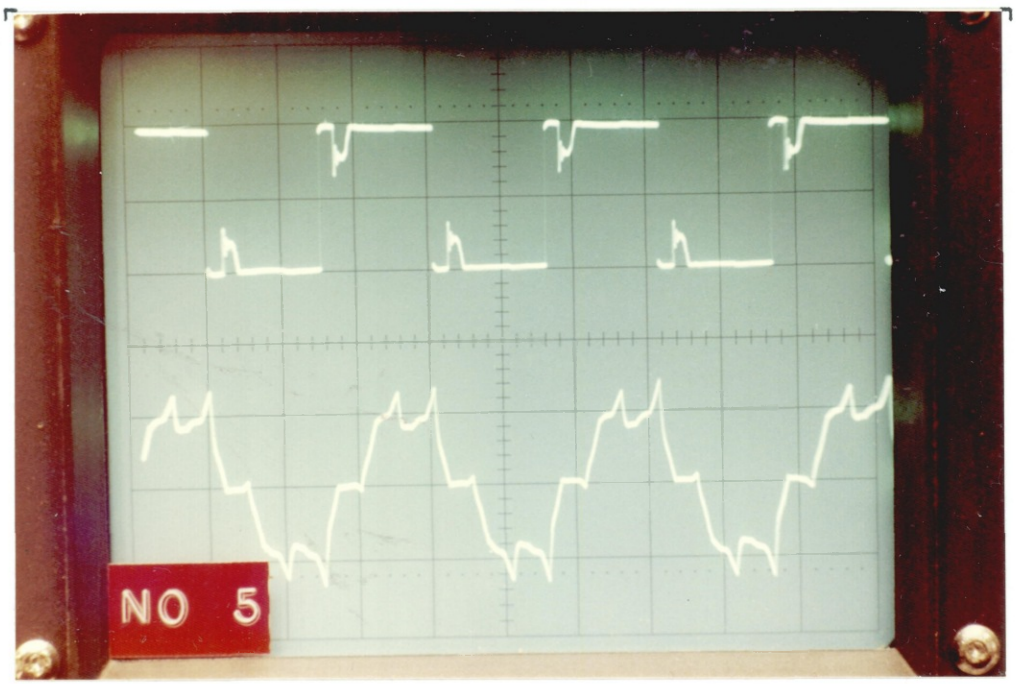


FIG. 3.50. 60° LOAD ANGLE – 100mNm LOAD ; 3960 rpm.
(TOP ; PHASE VOLTS – 50V/cm : BOTTOM ; PHASE CURRENT – 1A/cm : TIMEBASE ; 5ms/cm)

with no load torque. The applied phase voltage waveform is "clean" (free from any voltage spikes) and the freewheel periods can be clearly seen. The phase current waveform has the six step form that is a characteristic of 120°- 60° voltage source inverters and the influence of the sinusoidal back-emf can be seen. Fig. 3.48 shows the waveforms for 0° with a 100mNm load torque. The freewheel periods in the applied phase voltage fill up the 60° off sections of the waveform. The phase current still has six steps in it but it appears much more sinusoidal. Fig. 3.49 illustrates the conditions for 60° with no load torque. The applied voltage waveform is much more complex than for the 0° load angle. The phase current is again "six-step" and it contains a high sinusoidal component. Finally fig. 3.50 shows the waveforms for 60° with a 100mNm load torque. The applied phase voltage has a freewheel period lasting approximately one half of the 60° off period. The six-step current waveform is somewhat spiky but it is essentially sinusoidal as in the unloaded 60° case.

The example waveforms illustrate that the inverter was switching quickly and cleanly. The absence of overshoot voltage spikes shows that the supply decoupling was sufficient and also that the BYW29 diodes (used in the freewheel paths and the snubbers) were fast enough to cope with the rates of change of voltage present in the system.

Some simple checks were made on the magslip speed versus applied voltage characteristic at fixed load angles. The characteristic for a 0° load angle was only very approximately linear, and it was not thought to be worthwhile taking accurate measurements for comparison with the predictions in Chapter 2, especially since the "constant torque" assumption of the predictions is very difficult to achieve practically.

3.7 Electronic Load Angle Adjustment Circuits

3.7.1 'COMLOG' Based System

A load angle adjustment circuit based on the combinational logic system outlined in sub-section 3.4.2.1 and fig. 3.25, was designed and tested on the magslip system. The circuit used a 12 step Gray code pattern as shown in fig. 3.51. The diagram indicates the position of the four photodiodes needed to sense the Gray code.

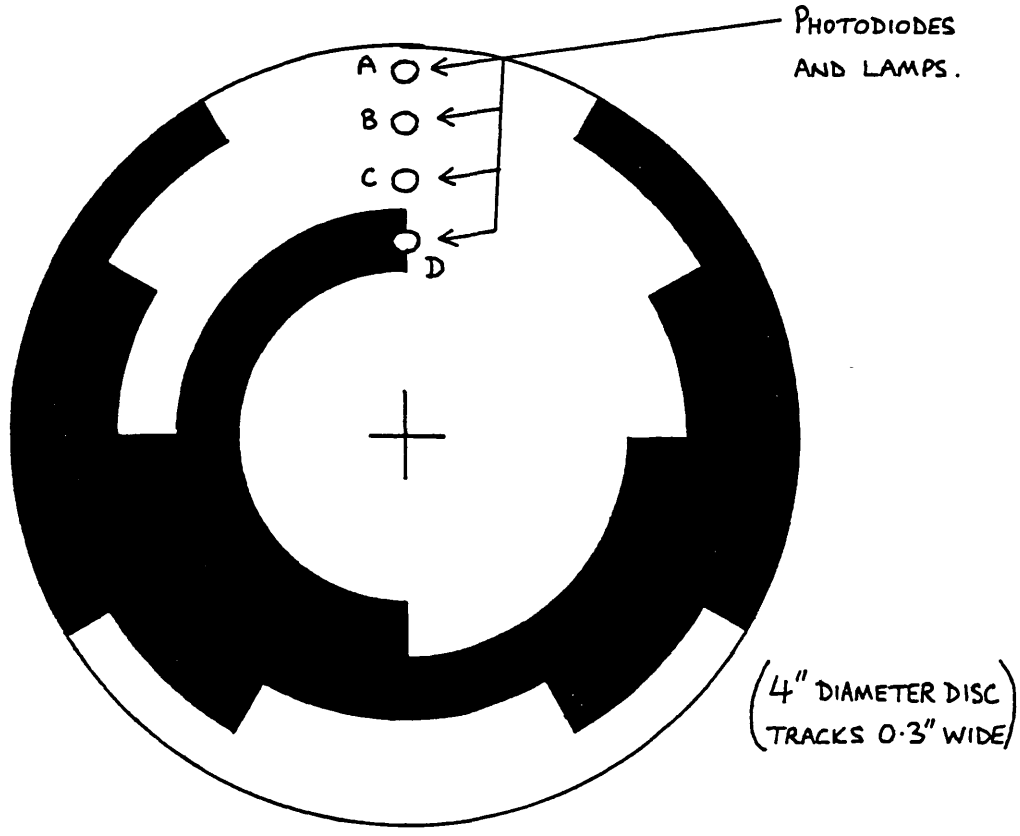


FIG. 3.51. THE 12 STEP GRAY CODE OPTICAL POSITION SENSOR USED IN THE 'COMLOG' BASED LOAD ANGLE ADJUSTMENT CIRCUIT.

The disc pattern was made by the same method as that used for the basic magslip system (see sub-section 3.5.4). The photodiode signals were amplified as shown earlier in fig. 3.29. They were then shaped by the TTL schmitt trigger arrangement shown in fig. 3.52. Fig. 3.52 also shows how a

six pulse per revolution tachometer signal was derived from the Gray code by using three 7486 exclusive or gates. This signal was used by the digital tachometer described in Chapter 8.

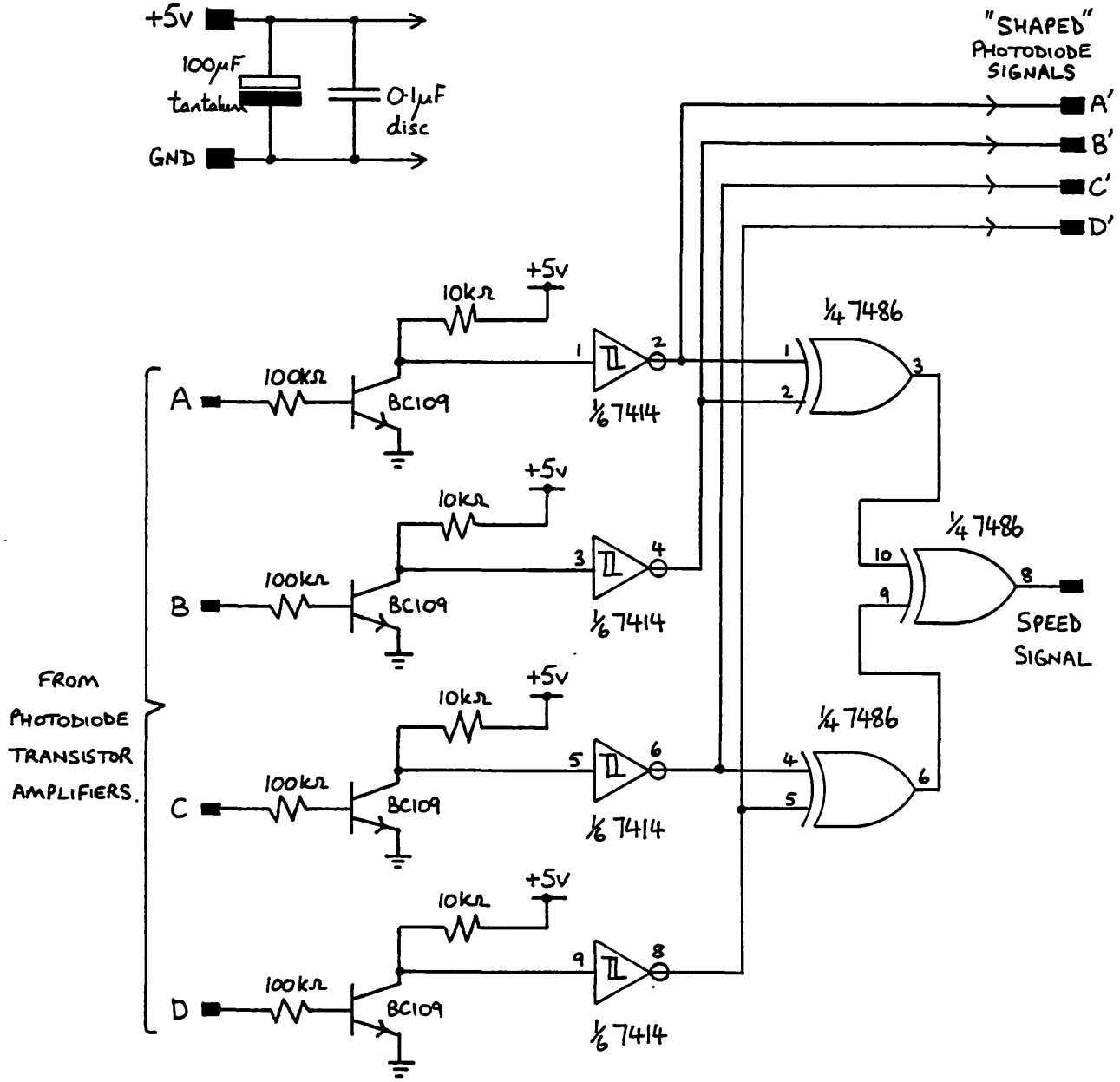


FIG. 3.52. CIRCUIT FOR SHAPING THE PHOTODIODE SIGNALS AND FOR GENERATING A SPEED SIGNAL

The circuit diagram of the combinational logic circuit is shown in fig. 3.53 and it can provide load angles of 0°, 30°, 60°, and 90°. The circuit uses TTL logic gates. The four bit Gray code $D'C'B'A'$ is buffered by 7408 AND gates (IC5(a),(b),(c),(d)). The four parallel-connected 74159 demultiplexer circuits (IC1, IC2, IC3, and IC4) perform the "first stage" Gray code to inverter state mappings for the four load angles. (They are the "A" logic blocks shown in fig. 3.25.) The required load angle is selected by putting the relevant load angle selector input to logic 0 and this enables one of the 74159 circuits. The enabled 74159 decodes the four bit input code and activates one of its sixteen outputs. Only twelve of the outputs are ever activated because the input Gray code is limited to twelve possible states.

The other three 74159 circuits are disabled and their outputs are inactive. The open collector outputs of the 74159 make it possible to directly connect the outputs of the four 74159 circuits together because only one of the circuits has control of the "inverter-state" bus at any time. The open collector feature of the 74159 outputs is very useful because it allows the function represented by logic block B in fig. 3.25 to be implemented with no extra logic gates. The arrangement of the interconnections onto the inverter state busbars (numbered 1 to 6 on fig. 3.53) determines the load angle produced by each 74159.

The final task of the circuit in fig. 3.53 is to implement the function represented by logic block C in fig. 3.25. The inverter state busbar is decoded into the corresponding inverter control signals (A1',A2',B1',B2',C1',C2') by 7400 NAND gates (IC6(a) to (d), and IC7(a),(b)). The control signal states are displayed by light emitting diodes (LED's) driven by 7406 inverters (IC8(a) to (f)).

The load angle selector code can be generated by a four way wafer switch. It is necessary to use a "break-

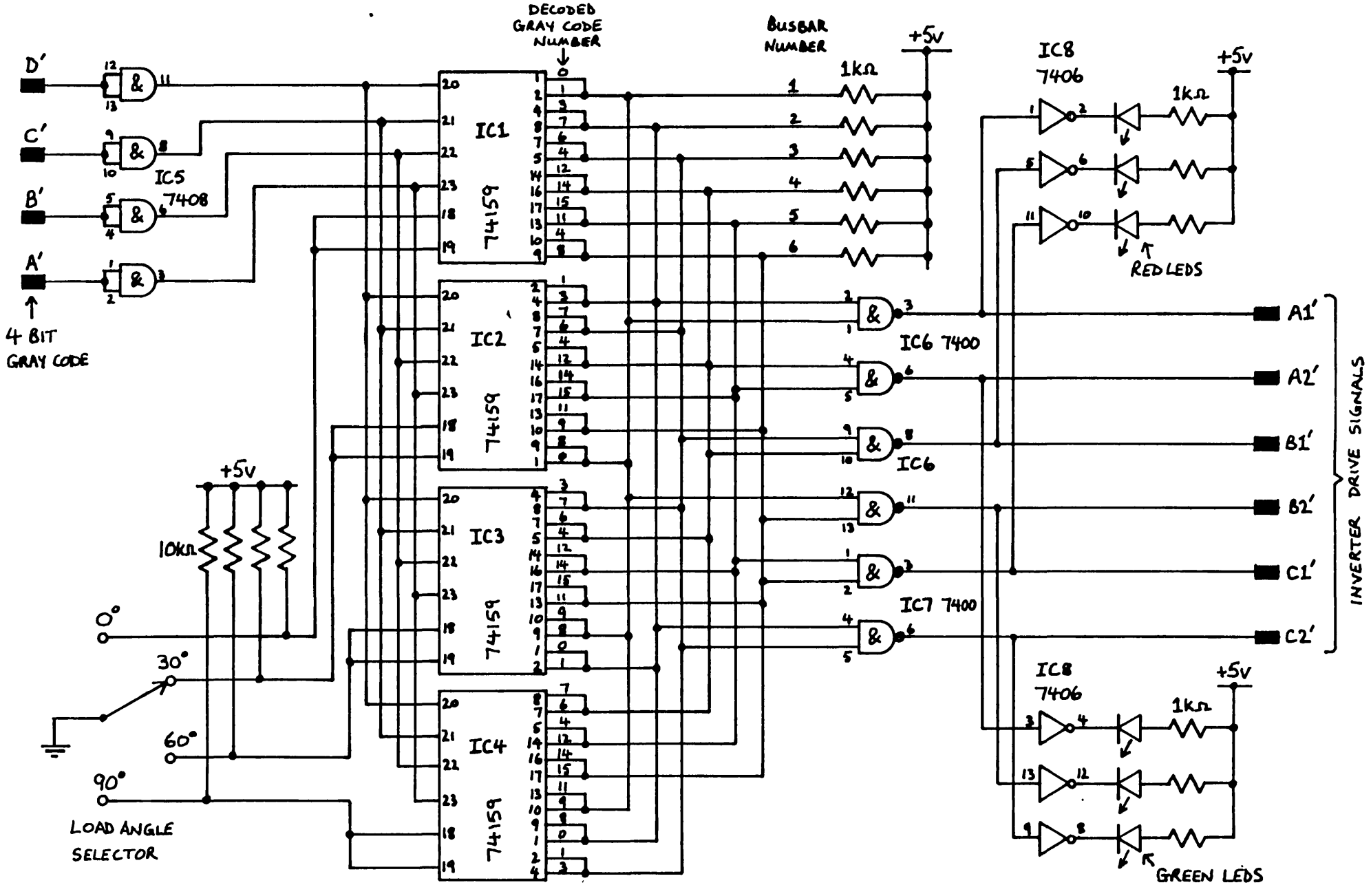


FIG. 3.53. THE GRAY-CODE-BASED COMBINATIONAL LOGIC LOAD ANGLE ADJUSTMENT CIRCUIT.

before-make" switch to prevent two 74159 circuits being enabled at any given time, since if this occurs there can be more than one "inverter-state" busbar line active. The lack of a suitable switch prompted the development of the circuit shown in fig. 3.54. This circuit checks the code generated by the load angle selector switch, and prevents erroneous codes reaching the load angle circuitry. It is not thought necessary to explain the functioning of the circuit since it is not vital to the action of the load angle circuit. The basic routine executed by the circuit is to examine the input lines coming from the selector switch and to detect whether more than one line is at logic 0. If this is the case the circuit continues to output the previous valid switch code until such time that a new permitted input code is recognised.

The load angle adjustment circuit and the switch code check circuit were constructed on separate "dual-in-line" circuit boards and the connections were made by the wire-wrap technique.

No problems were experienced with the load angle circuit in operational tests. It was possible to change load angle whilst the magslip was running and the performance characteristics were identical to those obtained from the basic magslip system.

3.7.2 'SELOG' Based System

A load angle adjustment circuit based on the sequential and arithmetic logic system outlined in sub-section 3.4.2.2 and fig. 3.26 was designed and tested on the magslip system. The circuit can provide load angles of 0^{o}, 30^{o}, 60^{o}, and 90^{o} in both directions of rotation. The motor can be reversed either by field flux reversal or stator flux reversal. The system is capable of synchronising itself from a standing start and it can operate from either a single or double edge sensing position detector.

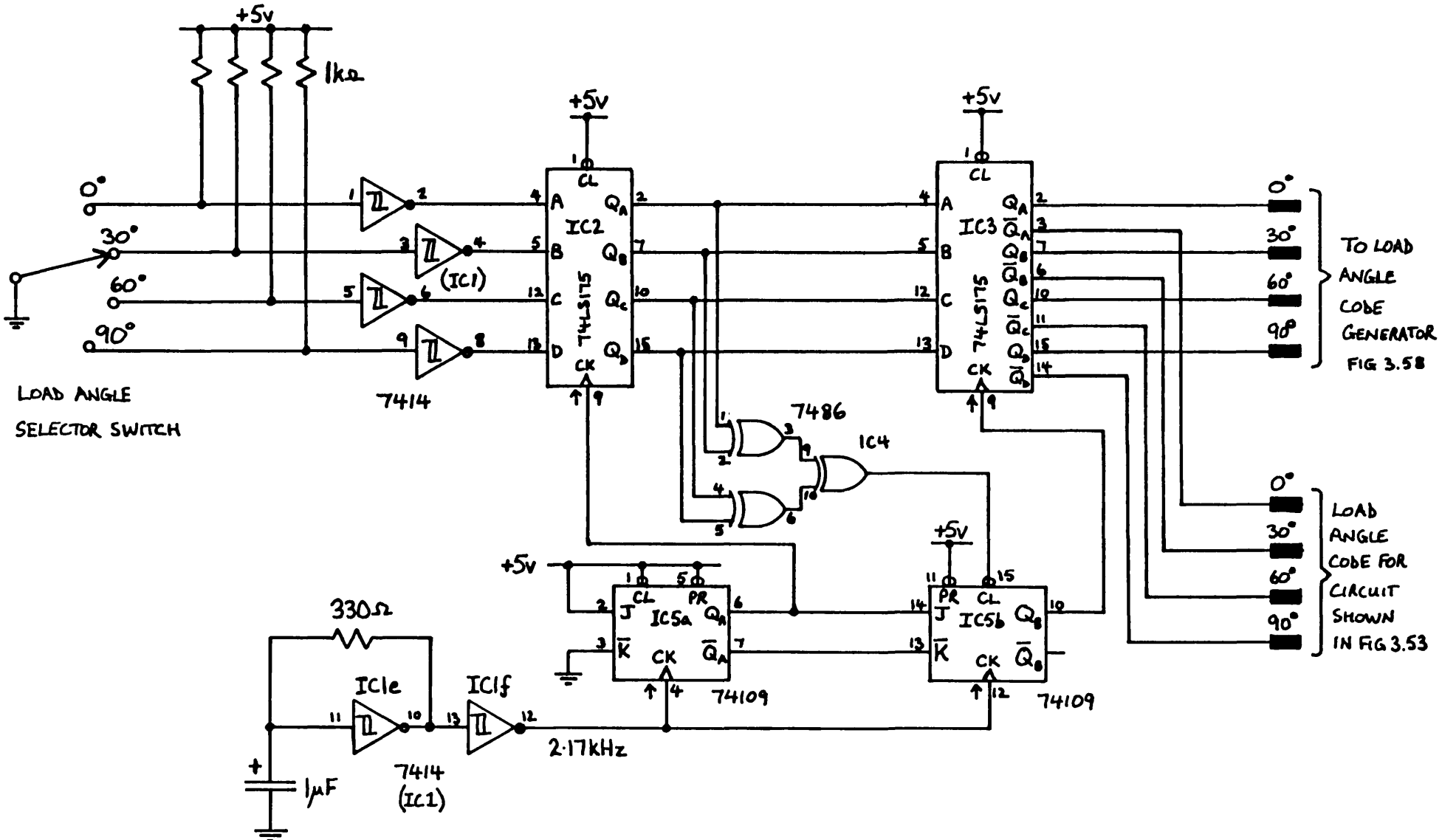


FIG. 3.54 LOAD ANGLE SELECTOR - SWITCH "DEBOUNCER AND CODE CHECKER".

The "single edge" and "double edge" disc patterns used to produce twelve SEG' pulses per revolution are shown in figs. 3.55(a) and (b) respectively. The positions of the photodiodes are indicated on the diagrams.

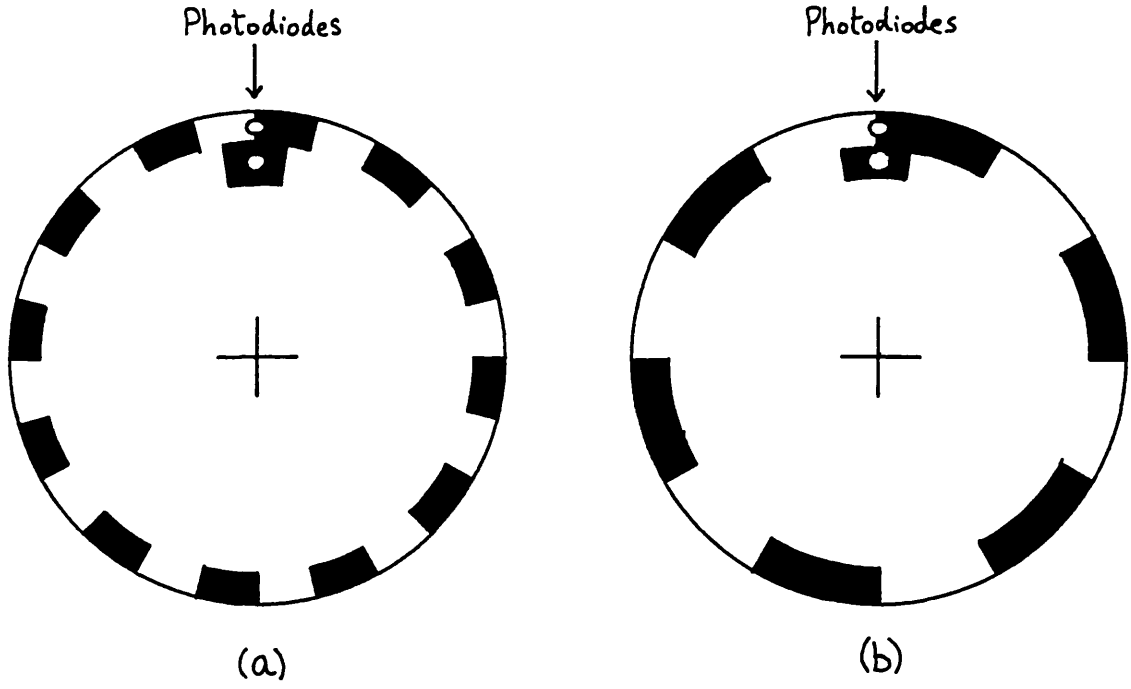


FIG. 3.55 "SINGLE EDGE" AND "DOUBLE EDGE" POSITION SENSOR DISCS PRODUCING TWELVE PULSES PER REVOLUTION.
(a) SINGLE EDGE, (b) DOUBLE EDGE.

The photodiode signals were shaped by the circuit used for the 'COMLOG' system (see fig. 3.52).

The load angle adjustment circuit is shown in figs. 3.56(a) and (b). The circuits in fig. 3.56(a) basically deal with:

(a) generating the SEG and SYNC signals from the SEG' and SYNC' information available from the optical sensors;

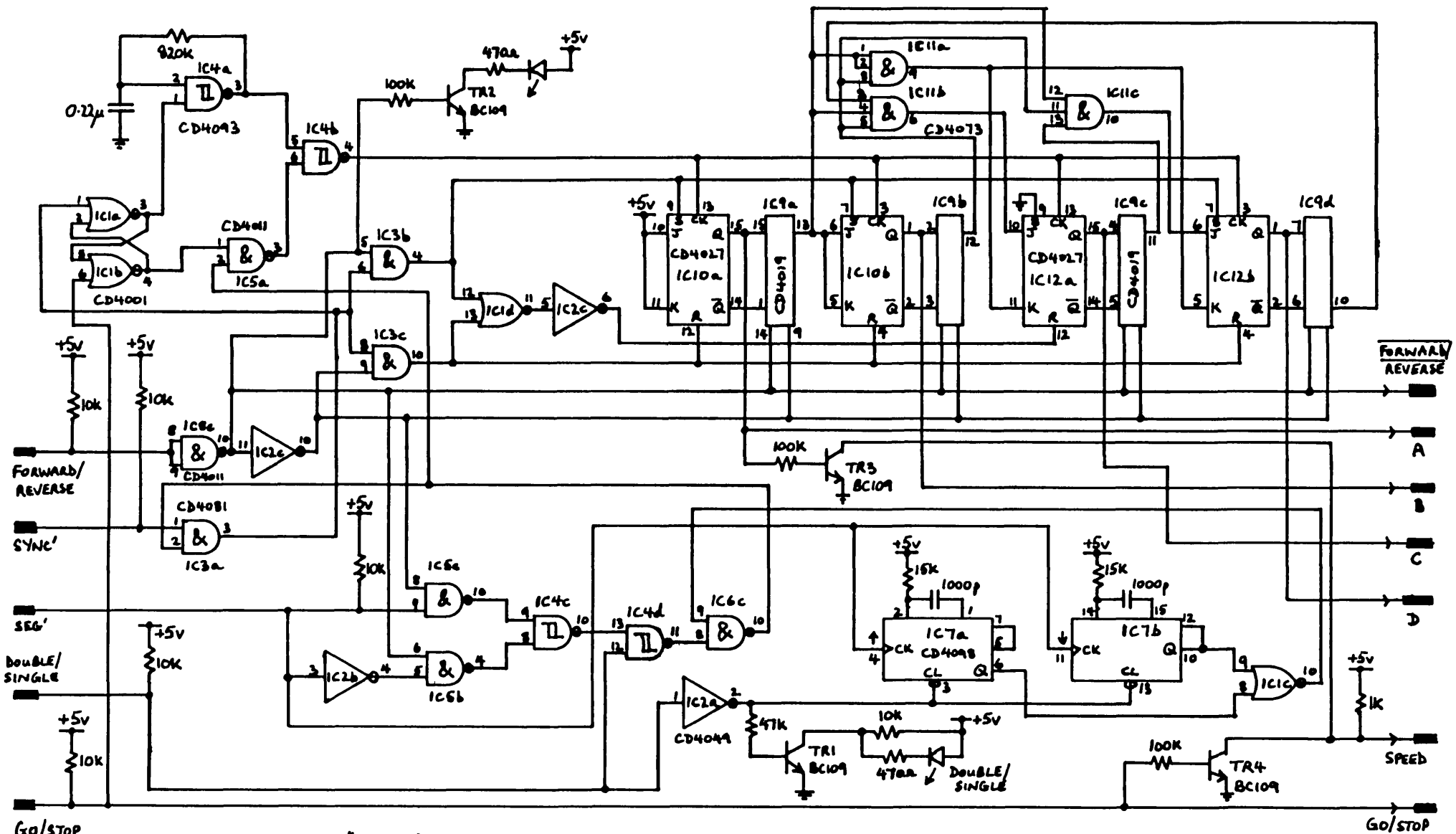


FIG. 3.56 a. "SELOG" BASED LOAD ANGLE ADJUSTMENT CIRCUIT (PART OF) [SEE FIG 3.56 b]

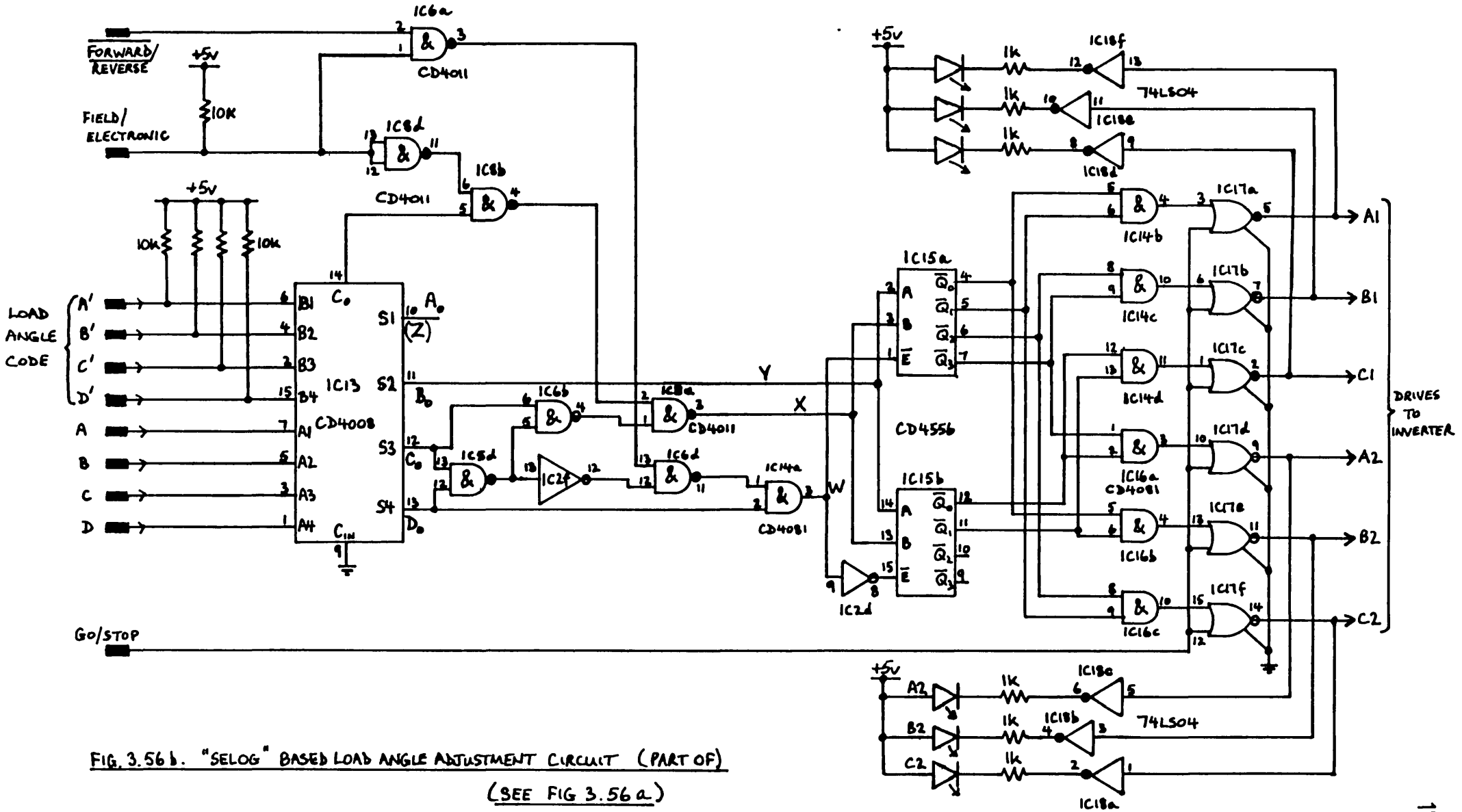


FIG. 3.56 b. "SELOG" BASED LOAD ANGLE ADJUSTMENT CIRCUIT (PART OF)

(SEE FIG 3.56 a)

(b) starting and synchronising;

(c) counting the SEG signals to generate a binary position code.

The circuits in fig. 3.56(b) basically:

(a) add the load angle shift code onto the binary position code;

(b) arrange the resultant binary code to remain within the permitted set of numbers;

(c) convert the resultant shifted position code into inverter control signals.

The basic blocks are discussed very briefly in the sub-sections below. The circuits shown in fig. 3.56(a) and (b) were assembled on a dual-in-line circuit board using wire-wrap connections. No problems were experienced with the circuits in operation. The magslip could be started reliably in either direction and the performance was identical to that obtained with the basic magslip system.

3.7.2.1 SEG and SYNC Signal Generation (fig. 3.56(a))

The Double/Single (D/S) mode input allows the circuit to be enabled for either double or single edge position sensor signals. The SEG' signal from the position sensor is fed to both the single edge logic circuits (IC2(b), IC5(b), IC5(c), and IC4(c)) and the double edge logic circuits (monostables IC7(a), IC7(b) and logic gate IC1(c)). When the D/S input is at logic 1 the single edge circuitry is enabled and when D/S is at logic 0 the double edge circuitry is enabled. The single edge circuitry receives information about the required direction of motor rotation from the FORWARD/REVERSE (F/R) input. Logic 1 on the F/R input indicates forward rotation and logic 0 indicates

reverse rotation. The outputs of the single and double edge circuits are routed via a NAND gate (IC6(c)) to produce the SEG signal. This signal is used in combination with the SYNC' input signal to generate the SYNC signal. The SEG and SYNC signals then pass on to the starting and synchronization circuit.

3.7.2.2 Starting and Synchronising Circuit (fig. 3.56(a))

The starting and synchronising circuit receives the SEG, SYNC, and GO/STOP (G/S) signals. When the G/S input is at logic 1 the system is in a stop state. The RS latch (IC1(a), IC1(b)) is held in a reset state and so the SEG signals are inhibited from passing via IC5(a) to the counter circuits. The 12Hz schmitt trigger oscillator is active and provides a train of starting SEG pulses to the counter circuits. However, the inverter control signals which are derived from the counter circuit outputs are not passed to the inverter whilst G/S is at logic 1. A start is initiated by putting G/S to logic 0. The inverter control signals are then gated to the inverter and the inverter switches through its sequence once per second. (The 12Hz clock circuit causes the counters to increment through one cycle in one second.) The magslip rotor steps around as the inverter goes through its sequence and eventually a SYNC signal is generated. The SYNC signal sets the RS latch and so causes the SEG signals to be routed to the counter circuits and the 12Hz clock circuit to be disabled. In addition, the SYNC signal causes the counter to be set to the appropriate synchronised value for the particular direction of rotation. Thus the system has started and synchronised itself. If G/S is returned to logic 1, the inverter control signals are cut off from the inverter and the RS latch is reset ready for a new start sequence.

3.7.2.3 SEG Signal Counter Circuit (fig. 3.56(a))

The counter is a synchronous circuit arrangement which

can count in an up or down direction through the set of numbers 0_{10} to 11_{10} inclusive (i.e. the cycle length is 12). The counter is based on four CD4027 JK flip flops (IC10(a), IC10(b), IC12(a), IC12(b)) with suitable extra control logic. The CD4019 AND-OR gates (IC9(a),(b),(c), and (d)) are the basic means by which the count direction is controlled by the FORWARD/REVERSE input information. The CD4073 AND gates (IC11(a),(b) and (c)) process the outputs of the AND-OR gates and control the synchronous count sequence. To enable the counter to become synchronised and to remain so during subsequent motor operation, the count value is set to a particular value whenever a SYNC pulse occurs. The logic gates IC3(b), IC3(c), IC1(d) and IC2(c) perform this task. When the motor is rotating in a forward direction the SYNC signal sets the counter to zero. In the reverse direction the SYNC signal sets the count value to 11_{10}. The outputs of the four flip flops (DCBA) make up the binary position code. A six pulse per revolution tachometer signal suitable for the digital tachometer (Chapter 8) is taken from the A output of the counter. The transistors (TR3, TR4) buffer the tachometer signal and also ensure that the tachometer signal is not active when the system is in a stop mode. (This is necessary because the counter is sequenced by the 12Hz clock when the system is in the stop mode.)

3.7.2.4 Binary Position Code Adder Circuit (fig. 3.56(b)

The counter binary output DCBA is added to the binary load angle code D'C'B'A' by a CD4008 four bit full adder circuit (IC13). The load angle codes required for forward and reverse motor operation with load angles of 0°, 30°, 60°, and 90° are summarised in fig. 3.57. The reversal of motor direction can be achieved either by field winding control (field flux reversal) or by electronic control (stator flux reversal), and fig. 3.57 includes the codes for both methods. Field winding or electronic reversal is selected by the FIELD/ELECTRONIC (F/E) input. With F/E at

logic 1 the system operates by field winding reversal and with F/E at logic 0 electronic reversing is selected.

		Binary Load Angle Code				Decimal Equivalent
	Load Angle	D'	C'	B'	A'	
Forward & Field Control F/R = 1 F/E = 1	0°	0	0	0	0	0
	30°	0	0	0	1	1
	60°	0	0	1	0	2
	90°	0	0	1	1	3
Reverse & Field Control F/R = 0 F/E = 1	0°	0	0	0	0	0
	30°	1	1	1	1	-1
	60°	1	1	1	0	-2
	90°	1	1	0	1	-3
Forward & Electronic Control F/R = 1 F/E = 0	0°	0	0	0	0	0
	30°	0	0	0	1	1
	60°	0	0	1	0	2
	90°	0	0	1	1	3
Reverse & Electronic Control F/R = 0 F/E = 0	0°	0	1	1	0	6
	30°	0	1	0	1	5
	60°	0	1	0	0	4
	90°	0	0	1	1	3

FIG. 3.57. THE BINARY LOAD ANGLE CODES USED IN THE CIRCUIT SHOWN IN FIGS 3.56(a) & (b).

The circuit shown in fig. 3.58 was designed to generate the D'C'B'A' code. A simple set of switches could have been used but it was felt that the chances of making a mistake which could damage the inverter were too large. The load angle code generator circuit received its input signals (0^{o}, 30^{o}, 60^{o}, and 90^{o}) from the switch code "checker" circuit shown earlier in fig. 3.54, and the two circuits were actually built on the same circuit board.

3.7.2.5 Resultant Binary Code Processing (fig. 3.56(b))

The resultant output number $D_oC_oB_oA_o$ from the full adder IC13 can be outside the set of numbers 0_{10} to 11_{10}. The logic gates IC8(d), IC6(a), IC6(d), IC2(f), IC5(d), IC8(b), IC6(b), IC14(a), and IC8(c) are arranged to modify the resultant number to give a final number WXYZ that is always within the required set of numbers. The gates examine the resultant number, the carry out bit (C_o), and the F/E and F/R inputs. No details have been given here of the design procedure used for the binary code processing logic because the chosen solution is just one of several possible solutions. For example, the necessary corrections could have been performed by a full adder unit. However, there were several "spare" logic gates available from other parts of the system and the logic gate solution was useful in that it allowed these gates to be utilised.

3.7.2.6 Generation of Inverter Control Signals (fig. 3.56(b))

There are six valid sets of inverter control signals (as shown in fig. 3.9). Since the resultant binary code WXYZ can have twelve states it is necessary to divide them by two. This is simply achieved by only using bits WXY (i.e. the least significant bit Z is ignored). The pair of two to four line demultiplexers (CD4556; IC15(a),(b)) decode WXY into six lines, only one of which is active at any time. The CD4081 AND gates (IC14(b),(c),(d), and IC16(a),(b),(c)) decode the selected line into the corres-

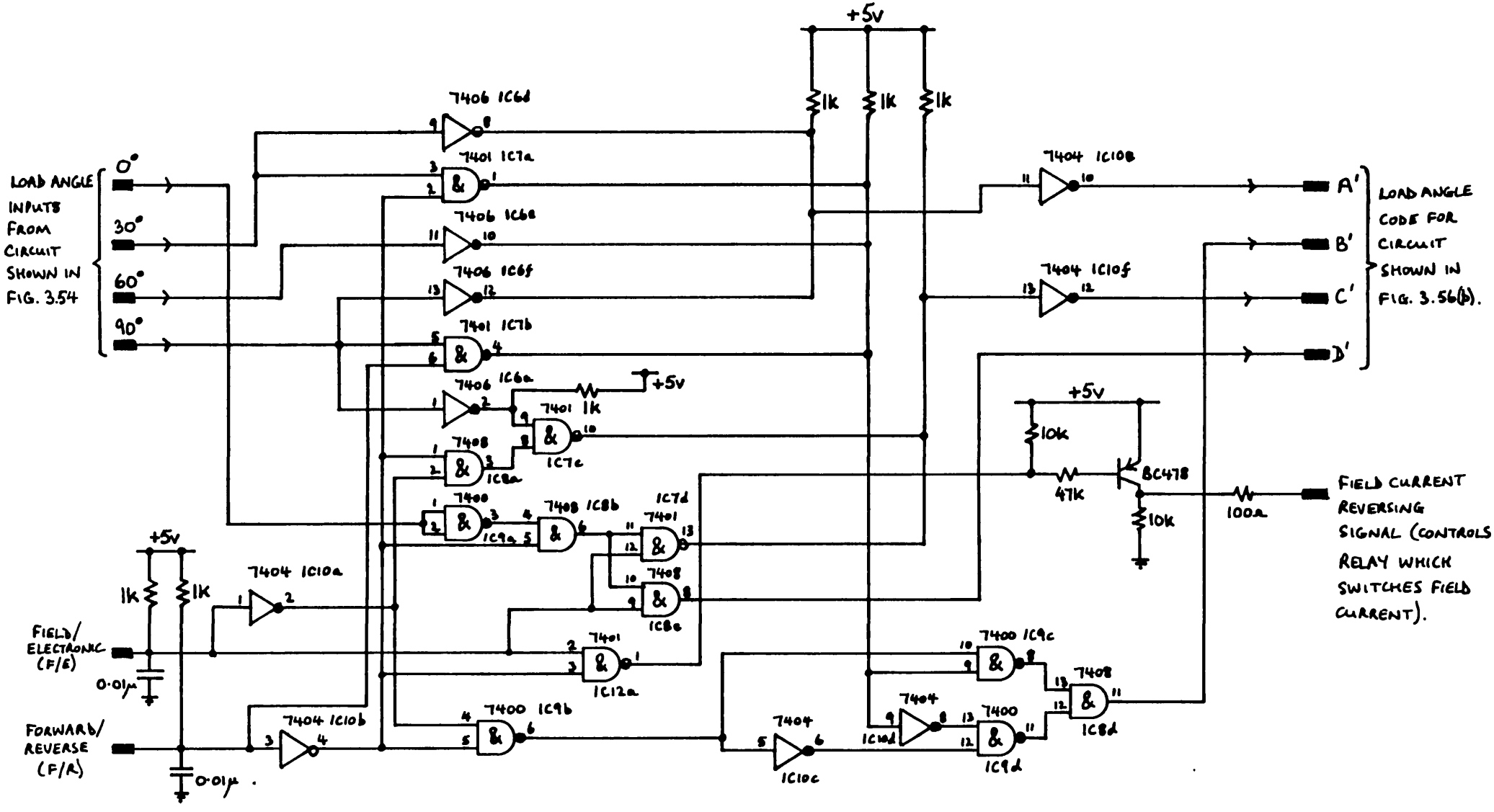


FIG. 3.58 CIRCUIT TO GENERATE THE D'C'B'A' CODE REQUIRED BY THE SELOG LOAD ANGLE VARIATION CIRCUIT SHOWN IN FIG. 3.56b.

ponding inverter control signals. The signals are gated out to the inverter via the strobed NOR gates (CD4502; IC17(a) to (f)). When the G/S input is at logic 1 (Stop selected) the NOR gates are disabled and their outputs are at logic 0, therefore causing the inverter to be inactive. The signals are allowed through to the inverter when G/S is at logic 0 (Go selected).

3.7.3 Photographs of the Load Angle Adjustment Electronics

The electronic hardware required to achieve load angle variation by the systems described in sub-sections 3.7.1 and 3.7.2 was not really very extensive. A photograph of the position sensor lamps, photodiodes, and waveform shaping circuitry is shown in fig. 3.59. The photograph in fig. 3.60 shows the three circuit boards that contain the total electronics for the two load angle adjustment systems. There is no doubt that the circuits could be easily combined into a single integrated circuit for commercial applications.

3.8 Conclusions

The work covered in this Chapter has illustrated how an autopiloted synchronous motor produces continuous torque. A quasi-square wave autopiloted motor system has been described and the test results obtained from it have been presented. It is concluded that the performance characteristics predicted by the sinusoidal methods discussed in Chapter 2 are reasonably accurate and in agreement with the practical results. Mechanical and electronic load angle adjustment has been implemented successfully. The various load angle settings available with the load angle adjustment techniques each have distinct torque/speed characteristics. The use of simple electronic load angle circuits has demonstrated that reliable "gearboxes" for autopiloted synchronous motor systems can be implemented, allowing the

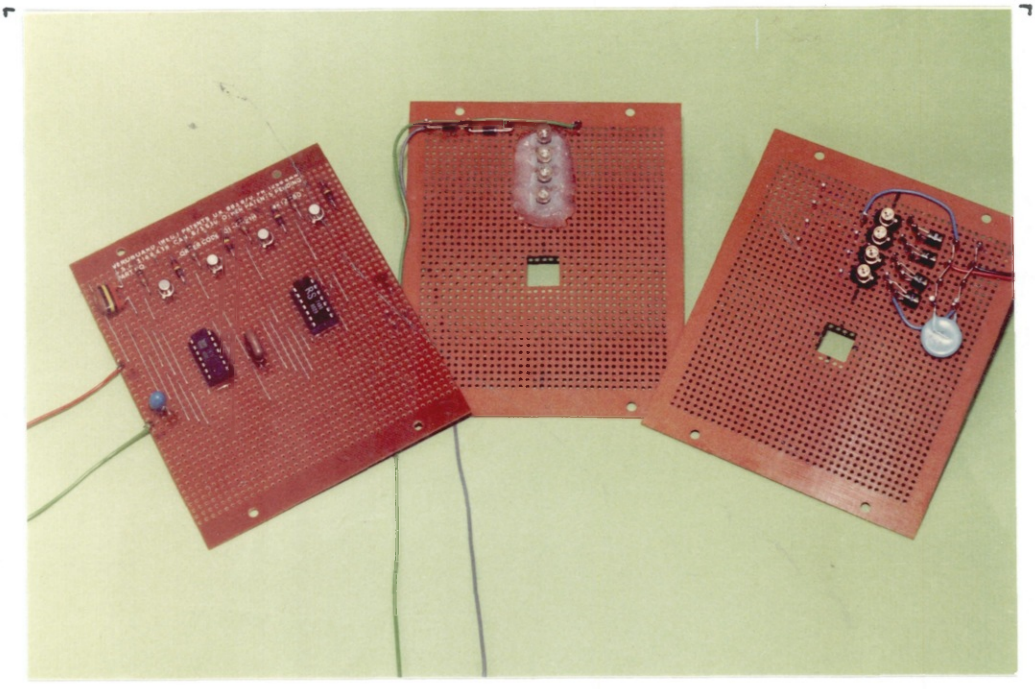


FIG. 3.59. THE POSITION SENSOR ELECTRONICS USED FOR THE ELECTRONIC LOAD ANGLE ADJUSTMENT EXPERIMENTS.

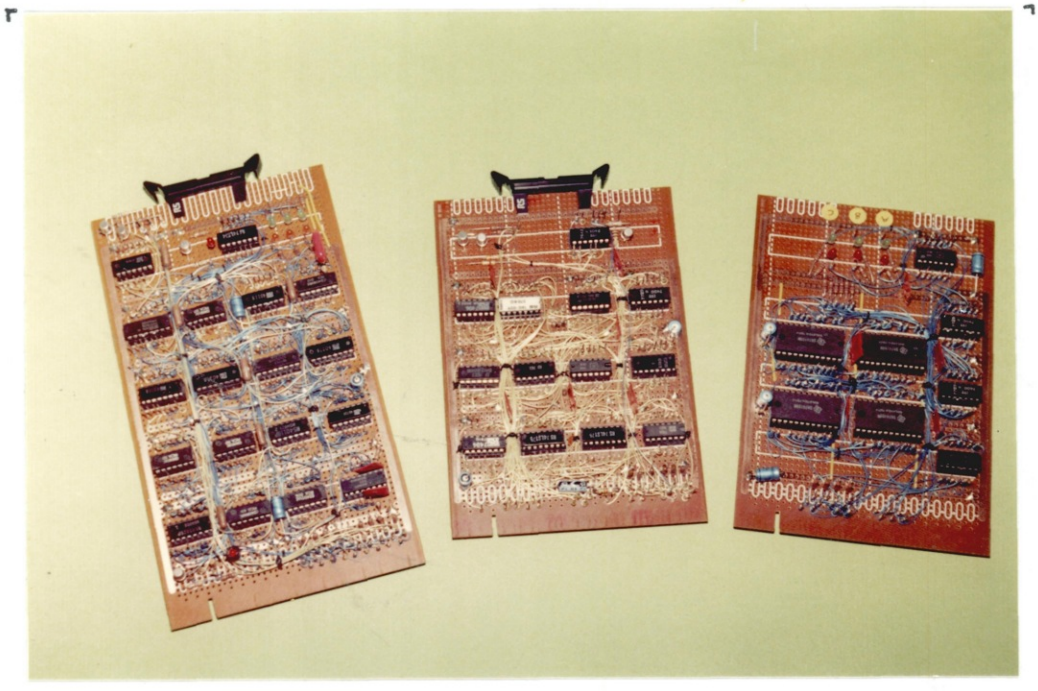


FIG. 3.60. THE CIRCUIT BOARDS FOR THE ELECTRONIC LOAD ANGLE ADJUSTMENT WORK.

selection of the most appropriate characteristic at any given load setting.

CHAPTER 4

THE USE OF A TEXAS TMS9900 MICROPROCESSOR TO AUTOPILOT HIGH SPEED SYNCHRONOUS MOTORS

4.1 Introduction

The basic aim of the project described in this thesis was to construct a high speed "square wave" synchronous motor and to autopilot it by a microprocessor system. Discrete electronic circuits can be used to autopilot synchronous motors and examples of the use of such circuits with the magslip motor are given in Chapter 3. It was intended that the functions performed by the discrete electronics should be executed as far as possible by a microprocessor with minimum additional hardware. The autopiloting electronics basically perform the task of controlling the inverter switching state in response to signals from a shaft position sensor. This process occurs in "real time" and so if a microprocessor is used in place of autopiloting electronics, it must likewise execute its software program in real time. It has been frequently claimed during recent years that a microprocessor can do anything providing it has the right software, and in general this is true, providing the microprocessor can execute the program in the available time. In motor control there is usually very little time available to perform the software routines and so the use of a microprocessor to autopilot a motor is not necessarily straightforward.

It is briefly explained in Chapter 3 how a microprocessor can perform the necessary autopiloting function. This chapter contains details of a Texas Instruments TMS9900

microprocessor system that was successfully used to autopilot the 7 phase "square wave" synchronous motor described in Chapter 6. The microprocessor system was initially tested on the established magslip motor system. This allowed the software to be debugged on a system that was known to operate reliably. Adjustments in the magslip load angle were achieved by means of software control and two position sensing systems were tried. Excellent performance was obtained up to 11500 rpm. Therefore, having proved the microprocessor system and software, the program was modified for use with the 7 phase "square wave" motor.

This chapter begins with a review of the tasks that a microprocessor can perform in a motor system and some of the problems that must be overcome are discussed. The reasons leading to the choice of the TMS9900 microprocessor are then explained and a description of the system including the necessary interfacing electronics is given. The latter parts of this chapter deal with the software developed for the magslip and 7 phase motor systems. Comments on the microprocessor system performance are given at the end of this chapter together with suggestions for system and software improvements.

4.2 The Use of Microprocessors in Autopiloted Motor Systems

Microprocessors are relatively new devices and there are many applications in which they have yet to be used. However, they are now sufficiently commonplace that descriptions of their basic structure and operation can be found in many recent electronics textbooks. Nevertheless, it is worthwhile stating that a typical microprocessor system consists of a microprocessor, a system clock, some Random Access Memory (RAM), some Read Only Memory (ROM), an Input/Output unit (I/O), and an interrupt handler. The system is interconnected by buses as shown schematically in fig. 4.1.

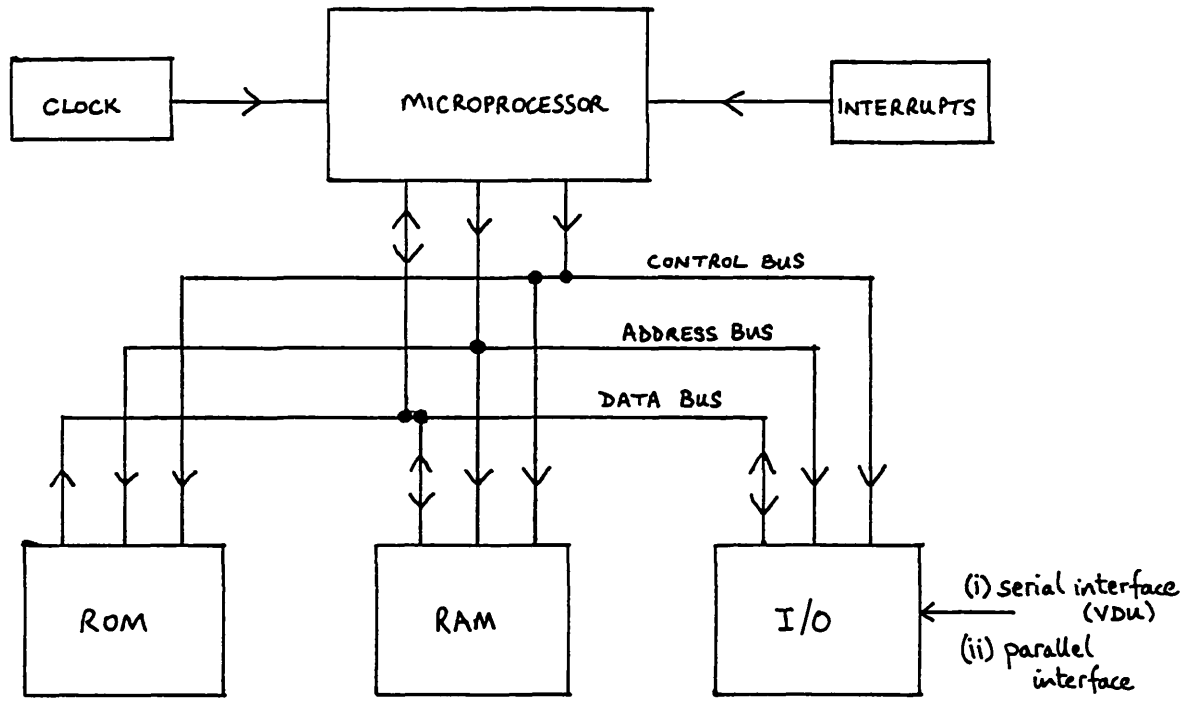


FIG. 4.1. BASIC MICROPROCESSOR SYSTEM

Data is shifted around the system on the data bus. The address and control buses ensure that the microprocessor only communicates with one location within the system at any time. The microprocessor deals with one instruction at a time and the rate at which it processes the program instructions is fixed by the system clock frequency.

A microprocessor can perform a variety of tasks in an autopiloted synchronous motor system. Firstly, it can make the basic decisions necessary to autopilot the motor. The microprocessor receives position information from the rotor position sensor and controls the inverter in such a way as to maintain the motor in synchronism. In a high speed motor there are many inverter switchings per second and so the microprocessor must be able to respond very fast to the position information to avoid loss of synchronism. Secondly, the microprocessor can monitor various variables in the motor system and attempt to regulate them within required limits. For example, it might be desired to control either

the speed, torque, or power factor in a motor system and the microprocessor can achieve this providing the necessary software is available. The response required to accurately control these motor variables is usually less than that needed to maintain synchronism. Therefore, these tasks can be given lower priorities within the program hierarchy. Thirdly, the microprocessor can deal with the communication of data between it and the machine operator and vice-versa. The methods of data entry and output are discussed in detail in Chapter 7. Fourthly, the microprocessor can be used in a supervisory role. For example, it can arrange the successful starting of the motor system and can ensure that the motor runs in the required direction. The microprocessor can check that the system is never in a dangerous state. Useful checks include the measurement of motor phase currents to detect short circuits on the inverter outputs; the monitoring of motor body temperature to prevent winding burn-outs; the monitoring of inverter switch temperatures to prevent thermal runaway; the measurement of rotor speed to detect dangerous overspeed conditions; and the sampling of the motor airgap flux to check the operating conditions of the motor.

The technical press has contained reports on the implementation of some of the above tasks by microprocessors and several references are given below. The tasks have not in the main been performed on autopiloted synchronous motor systems but they are worth referencing to illustrate the capabilities of microprocessors. Schnieder (4.1) has reported on the control of a 3 phase thyristor a.c./d.c. converter used in a d.c. drive. The microprocessor controlled the thyristors as well as the speed and current control loops. Sen et al (4.2) have reported on the microprocessor control of an induction motor with flux regulation. Harashima et al (4.3) have published details of a microprocessor based digital phase-locked loop speed control system. Rajashekara et al (4.4) have reported on a microprocessor controlled sinusoidal pulse-width modulation

inverter. The microprocessor uses direct memory access (DMA) in order to allow more frequent changes of the inverter state. Le-Huy et al (4.5) have published a paper on a microprocessor autopiloted synchronous motor drive. The motor is inverter-fed and the paper outlines the microprocessor-controlled starting scheme. Rieke (4.6) has given details of the microprocessor control of a four phase reluctance motor, in which changes of operating conditions were achieved via a VDU connected to the microprocessor. Finally, Fornel et al (4.7) have published details of the numerical speed control of a current-fed asynchronous machine in which the TMS9900 16 bit microprocessor was used to achieve the required computational accuracy.

The references illustrate the principal advantage of microprocessors when compared with discrete logic circuits. A microprocessor is a very "flexible" device and it can be adapted into many applications. A motor control system based on a microprocessor is much more flexible than a similar system using a discrete logic controller. If changes in the operation of the motor system are required it is usually a simple matter of modifying the microprocessor software (that is, no hardware changes are necessary). Unfortunately, the flexibility is not achieved without one significant disadvantage. The microprocessor is a sequential logic device which performs <u>one</u> instruction at a time, with each instruction taking a certain number of clock cycles to complete. Since even relatively simple tasks usually require several (or even many) instructions, it follows that a microprocessor can take a significant number of clock cycles (amounting to perhaps tens of microseconds) to complete one task. Hence there is a limit on the number of times that a task can be performed per second, and this limit is essentially fixed by the clock cycle period which is typically of the order of 1μs in many common microprocessors. The execution of a task can be speeded up either by reducing the clock cycle period (e.g. by choosing a different microprocessor) and/or by reducing the number of

software instructions in a given task (i.e. optimising the software; this can be achieved by clever programming or by choosing a microprocessor with an instruction set that is very relevant to the required tasks). In contrast a discrete logic circuit is much faster in its execution of a task, since the main limitations on its speed are the propagation delays through the logic gates. (Such propagation delays are of the order of tens to hundreds of nanoseconds.)

The versatility of a microprocessor allows it to simulate much electronic hardware by software. Hence ideally, providing the microprocessor is fast enough, it can replace virtually all the system hardware. In a motor system there are several tasks that require processing at the same time and so the microprocessor must divide its time appropriately amongst the various tasks. When a microprocessor is used to autopilot a synchronous motor, the system is arranged in the form shown in fig. 4.2. The position sensor is arranged as an input peripheral whilst the inverter and adjustable d.c. power supply are configured as output peripherals. If the motor rotates at high speed it is necessary for the microprocessor to act on the position sensor information very frequently. It is possible that under certain circumstances the microprocessor can be fully occupied by the autopiloting task (the maintenance of synchronism) and so cannot deal with any of the lower priority tasks such as speed regulation. Indeed, above a particular speed the position information changes at a rate faster than the microprocessor can deal with and so the microprocessor limits the maximum motor speed that can be achieved. It is possible to think of the microprocessor as a "bottleneck" through which the required tasks will not go quickly enough. This problem is very relevant when considering the application of a microprocessor to a high speed synchronous motor such as the one described in this thesis.

It is relevant at this point to consider the form of the rotor position sensor signal that can be supplied to

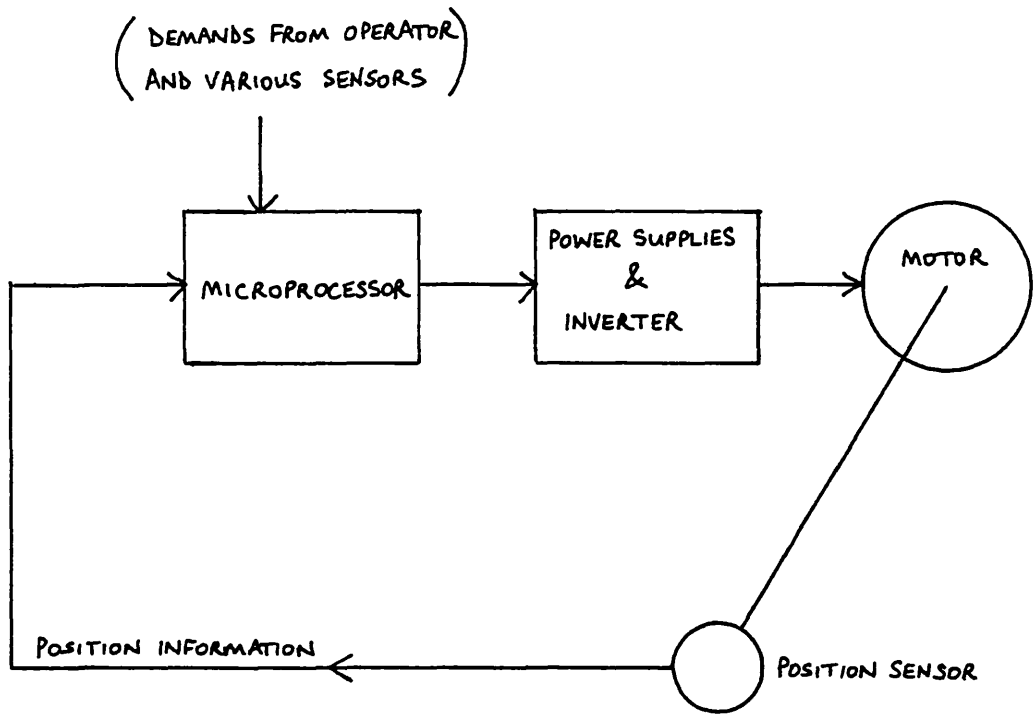


FIG. 4.2. MICROPROCESSOR USED TO AUTOPILOT A SYNCHRONOUS MOTOR.

the microprocessor. The types of position sensor signals are covered in some detail in Chapter 3 and they may be summarised in general either as absolute codes (e.g. Gray, binary, binary coded decimal), in which the position information is complete, or as incremental codes in which a signal is generated each time the rotor moves through a fixed number of degrees. An incremental position sensor is the simplest type. Such a sensor can produce an incremental signal (SEG') every few degrees plus a marker signal (SYNC') once every revolution. The SEG' and SYNC' signals are used to generate interrupts to the microprocessor and each time it receives an interrupt it calculates the new position of the rotor and controls the inverter accordingly. When an absolute position sensor is used, the microprocessor samples the position code at regular intervals and hence controls the inverter. The samples must be made sufficiently often so that synchronism is ensured. In practice, therefore, the microprocessor samples the absolute code as often as it can whilst performing other system tasks. Hence the absolute code may be sampled far too often at low motor speeds leading to a waste of microprocessor time. This problem does not occur with the inter-

rupt based solution because the microprocessor only deals with the position sensor when it is interrupted. At all other times the microprocessor can process other system tasks.

Unfortunately, the interrupt solution suffers from one drawback which becomes significant at high interrupt rates (i.e. at high motor speeds). Each time an interrupt occurs there are a fixed number of clock cycles during which the microprocessor leaves the task that it is performing and transfers to the interrupt service routine. At the end of the service routine there are several more clock cycles taken up by the return to the previous task. The time taken for the transfers can be significant and it produces a delay in the execution of the interrupt which is proportionately more significant at high interrupt rates.

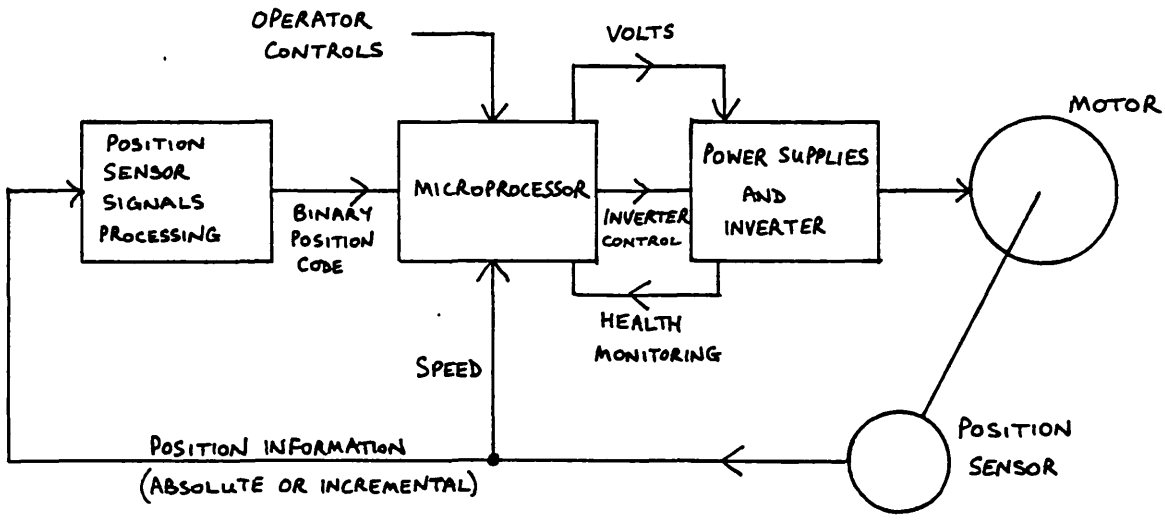
However, in general, there is little to choose between the use of absolute or incremental position encoders with microprocessors. A neat combination of an incremental position sensor with a sampling microprocessor strategy can be arranged by using a binary counter to count the SEG' signals with the SYNC' signal resetting the counter every revolution. The microprocessor samples the counter whenever possible. This method allows the use of a simple position sensor as well as avoiding the interrupt service transfer time delays.

It can be argued that a microprocessor which is capable of making relatively complex "decisions" is wasted when it is used to perform simple repetitive tasks such as autopiloting. In Chapter 3 it is explained how load angle control in an autopiloted system can be achieved by using a position sensor whose resolution is finer than that required for the basic autopiloting task. The resulting sensor signals on a high speed motor require frequent attention from the microprocessor. Since the main limitation of a microprocessor in such an application is the speed at

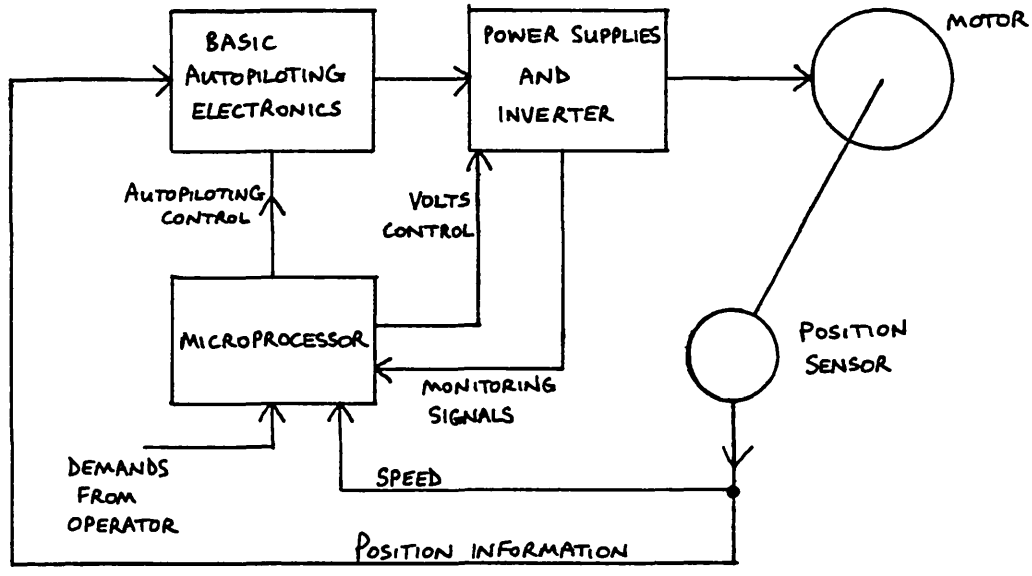
which it can process incoming signals, it is sensible to consider using the microprocessor in alternative ways.

Microprocessor time spent on the autopiloting task can be minimised or eliminated completely by a suitable combination of a microprocessor and discrete logic. In this way it is possible to combine the "flexibility" of a microprocessor with the speed of discrete circuitry. For example, at any speed there is an optimum load angle at which a synchronous motor can operate to maximise its torque output as discussed in Chapters 2 and 3. A microprocessor could be used to monitor the motor speed and calculate the optimum load angle, and then "instruct" a discrete logic circuit to implement the required angle.

The microprocessor can be retained in the signal path or can be removed to a purely management role as shown in figures 4.3(a) and 4.3(b) respectively. In the system shown in fig. 4.3(a) the microprocessor maps the processed position code to the appropriate inverter state. It is useful to retain the microprocessor in the signal path because it allows direct software control on the inverter states, thus allowing simple reversing for example. In the system shown in fig. 4.3(b) the microprocessor's time is concentrated on such tasks as the selection of load angle, speed control, system health monitoring, starting procedures, and communication with the machine operator. A refinement in this approach is to use more than one microprocessor in the system as shown in fig. 4.4. This allows all tasks to be achieved by microprocessors with no timing problems and yet minimises the discrete logic required. One microprocessor is arranged to deal exclusively with the autopiloting task (i.e. it is not interrupted from this at all) and one or more other microprocessors perform the management and monitoring functions in the system. The falling cost of microprocessors makes the use of such multi-processor solutions more likely in the future.



(a) MICROPROCESSOR IN THE MAIN AUTOPILOTING LOOP.



(b) MICROPROCESSOR EMPLOYED IN A "MANAGEMENT" MODE

FIG. 4.3. ALTERNATIVE WAYS OF USING A MICROPROCESSOR IN AN AUTOPILOTED SYNCHRONOUS MOTOR SYSTEM.

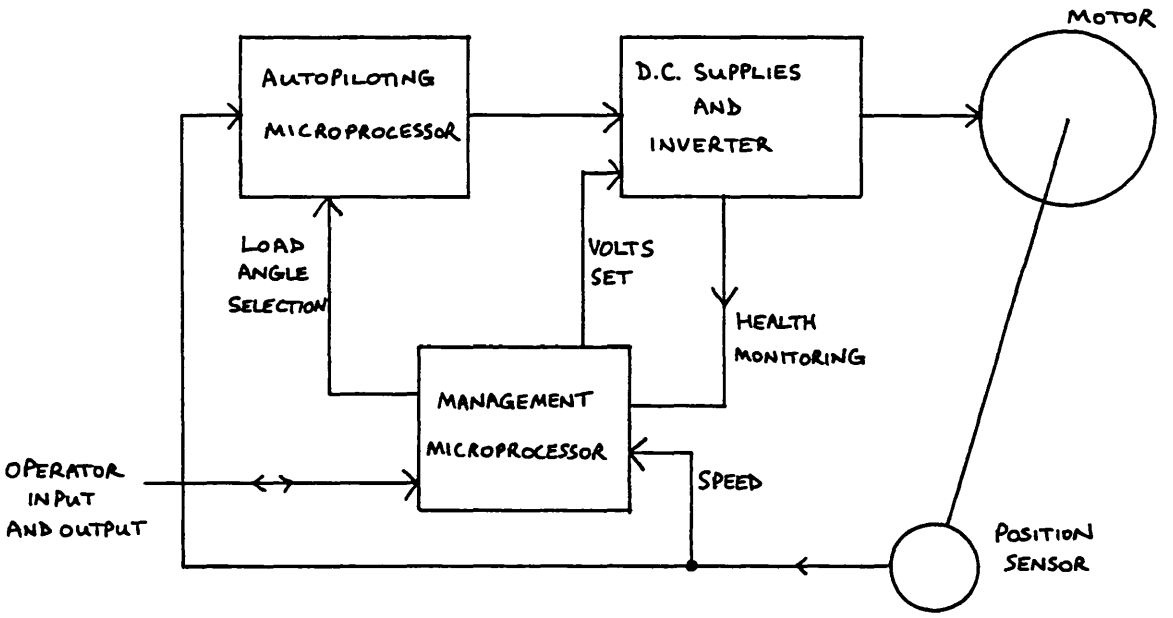


FIG.4.4 A MULTI-PROCESSOR AUTOPILOTED MOTOR SYSTEM

4.3 Points to Consider in the Implementation of a Microprocessor Autopiloted Synchronous Motor System

The use of a single microprocessor to autopilot a "square wave" synchronous motor is not particularly difficult but there are several practical points that deserve consideration during the initial planning stage. They are as follows:

(a) the system must be arranged to be as failsafe as possible. For example, if during an emergency the microprocessor is reset to halt the program execution, it is desirable that the peripherals (such as the inverter) should be set into a safe state and not remain in the active state they were in at the time of the reset.

(b) The system should be arranged to be insensitive to electrical interference. This can be achieved to some extent by the use of filters on power supply lines and some

circuit shielding. In addition the use of optical isolation in signal lines reduces the transmission of noise around the system.

(c) It is essential that communication between the microprocessor and the human operator is possible at ALL times. The microprocessor must be able to input data and commands whilst it is simultaneously autopiloting the motor. It is especially important that the stop instruction is always rapidly obeyed.

(d) The microprocessor controlled motor system should appear as a normal piece of equipment to the operator. The microprocessor should be invisible to the operator: that is, the system should not require complicated machine-code driving instructions. The control peripheral can be arranged with easily understood function buttons such as stop, go, forward, reverse, etcetera. In addition, any initialisation prior to a motor run should be simple and not require reference to a "driving manual".

(e) The interfacing between the microprocessor and the various system peripherals, should be arranged to reduce the possibility of damage to the microprocessor in the event of a failure in any peripheral. This precaution is especially worthwhile between the microprocessor and the power electronics of the inverter. Good isolation can be achieved very cheaply by the use of opto-couplers.

(f) The microprocessor is a digital device. In order to reduce the number of analogue to digital conversions (and vice versa) within the system, it is sensible to implement as many of the functions as possible in digital form, providing this does not lead to undue complexity. The measurement of speed is one function that can be easily performed digitally.

(g) The processing speed of the microprocessor is governed

by its clock rate and its instruction set. The time taken to execute a software routine can be minimised by efficient program writing. For example, trigonometric functions can be rapidly determined by using "look-up tables" and fast multiplication can be performed by special purpose arithmetic peripherals.

(h) Some interlocking in the system is desirable to protect the inverter from being inadvertently damaged. Interlocking ensures that particular parts of the system are only active when they should be. The peripherals can be arranged so that they are inactive unless they receive the correct enable word, and they only remain active as long as the word is present.

(i) The debugging of faults in a microprocessor controlled system is especially difficult because the faults can be due to hardware and/or software. Careful design of the system can ensure that individual parts can be checked for correct operation and this helps to narrow down where the fault is.

4.4 The Choice of a Suitable Microprocessor for the High Speed "Square Wave" Synchronous Motor System

Microprocessors can be selected on the basis of a variety of features including:

(a) the size of the data bus. Microprocessors are currently available with 4, 8, and 16 bit data buses.

(b) The size of memory that can be accessed.

(c) The instruction times (determined by the clock frequency).

(d) The instruction set available. Some microprocessors have a very limited instruction set whilst others possess powerful instructions such as multiply or divide.

(e) The power consumption. This is related to the clock frequency used and it rises as the clock frequency is increased. The lowest power consumption microprocessors are based on CMOS technology.

It was decided that the main thrust of the microprocessor work should be concentrated on the basic problem of autopiloting the high speed "square wave" motor. This is basically a simple logical task and so a complicated instruction set is not necessary. Several microprocessors were considered for use in the motor system. They included the Motorola M6800 (8 bit); the Motorola M68000 (16 bit); the Intel 8080 (8 bit); the Zilog Z80 (8 bit); the Zilog Z8000 (16 bit); the RCA CDP1802 (8 bit CMOS); the Texas Instruments TMS9900 (16 bit); and the Ferranti F100-L (16 bit). The Z8000 and M68000 were new and difficult to obtain whilst little information was available about the F100-L. Therefore, the choice amongst the 16 bit machines was effectively restricted to the TMS9900. The choice amongst the 8 bit machines was not so restricted and a deliberate decision was made to just consider two; namely the M6800 and the CDP1802. These microprocessors were readily available in cheap development kits. Personnel with experience in using the TMS9900, the M6800, and the CDP1802 were available for discussion, and it was established that they were all equally suited to the required autopiloting application.

It was decided that the number of bits needed to control the system inverter had to be established so that a choice of microprocessor could be made. Therefore, the basic designs for the square wave motor and the inverter

were carried out at this point, with the microprocessor control aspect kept very much in mind at all times. Two bits per phase are required to control the MOSFET inverter described in Chapter 5. The inverter was designed to drive the "square wave" synchronous motor described in Chapter 6. The motor phase number was selected as seven for the reasons given in section 6.4. (Principally on the one hand a high ODD phase number was desired to achieve the square wave objective, whilst on the other hand, the need to achieve a fast operating speed meant that each inverter state should be fully defined by no more than a single 16 bit parallel control word transfer.) The 7 phase MOSFET inverter requires a 14 bit control word, and this can be provided in a single parallel data transfer by the 16 bit TMS9900 microprocessor.

4.5 The TMS9900 Microprocessor System

4.5.1 The "Architecture" of the TMS9900 Microprocessor

The TMS9900 microprocessor is a single-chip 16 bit central processing unit (CPU). It uses a 3MHz clock, and the instruction execution times range from about 3μs for a jump, up to about 46μs for the longest divide operation. Instructions typically take between 3μs to 15μs. It has a relatively good instruction set considering that it was designed long before the newest generation of 16 bit microprocessors. The TMS9900 can address a maximum of 32768 16 bit words in memory (32K words). The CPU can work on either words (16 bits) or bytes (8 bits); therefore the memory addresses refer to individual bytes, and the addresses of words are always even (0000_{16}, 0002_{16}, etcetera). The TMS9900 is unique in that it does not possess any programmable registers (accumulators), on which most other microprocessors are based. The TMS9900 employs a memory-to-memory architecture in which blocks of RAM are designated as workspace, and so internal hardware registers are

replaced by memory based registers.

Within the TMS9900 there are just three 16 bit programmable registers: a program counter (PC), a workspace pointer (WP), and a status register (ST). The program counter contains the address of the next instruction to be executed, and the status register stores various status decisions that are made during the execution of some instructions. The workspace pointer is unique to the TMS9900. It is a very powerful register since it identifies the first of sixteen 16 bit RAM locations which act as general purpose accumulators (called registers 0 to 15). All of the 16 registers can be operated on by the full instruction set. In addition, some registers (11 to 15 inclusive) serve special pointer storage functions during the microprocessor operation. For example, when a branch and link instruction is executed, the microprocessor jumps to a new point in the program. The program counter contents existing just before the jump are stored in workspace register 11, so that a return to the previous point in the program can be made if desired. Hence a "link" is achieved via register 11.

Any 16 contiguous words of RAM may serve as the current 16 general purpose registers for the TMS9900. There can be as many sets of 16 bit registers as required, limited only by the size of implemented memory. If more than one set of 16 bit registers are used in a program, only one set can be selected at any time. The workspace pointer register identifies the first of the 16 contiguous memory locations serving as the current 16 general purpose registers. Each general purpose register may be used as an accumulator, an operand register, an address register, or an index register.

Having 16 general purpose registers in RAM, rather than in the CPU, is very useful in the servicing of interrupts. When an interrupt is acknowledged it is not neces-

sary to save the contents of the general purpose registers on a stack; the TMS9900 interrupt handling logic simply switches to a new workspace (i.e. a new set of 16 general purpose registers). The old PC, WP, and ST register contents are stored in registers 13, 14, and 15 respectively of the new workspace, to enable the microprocessor to return to its previous task at the completion of the interrupt routine. The TMS9900 is therefore well suited to interrupt based programs.

The TMS9900 can handle 16 interrupts and it assesses the priority of an interrupt when it occurs. Thus low priority interrupts can be interrupted by higher priority interrupts. The highest priority interrupt (Int 0) is reserved for the microprocessor reset and so only 15 interrupts are available for general use. The vectoring information (indicating the new WP and PC values) for the 16 interrupts is stored in memory between addresses 0_{16} and $3E_{16}$.

The ability to change workspaces makes the TMS9900 also very suitable for the use of subroutines in which independent registers, separate from the main program, are needed. The microprocessor can branch to a new workspace under software control (BLWP instruction) and return at the end of the subroutine. The return PC, WP, and ST information are stored in registers 13, 14, and 15 respectively of the subroutine workspace. An additional useful subroutine facility is available in the instruction set. Up to 16 subroutines can be defined as extended operations (XOP's 0 to 15). The vectoring information for the XOP's is stored in memory between addresses 40_{16} and $7E_{16}$. XOP's allow a parameter to be passed to the subroutine by placing a linking address value (defined in the XOP instruction) in register 11 of the subroutine workspace.

The change of workspace initiated by interrupt or software is referred to by Texas Instruments as a "context

switch". The disadvantage of using registers located in memory is that the TMS9900 cannot run any program without being configured with some RAM. More important, however, is the loss of the speed associated with hardware register-to-register operations because every register access involves a memory access. The speed of the TMS9900 is therefore limited very much by the read and write times of the RAM connected to it.

The TMS9900 also possesses unusual input/output (I/O) logic. In addition to the standard method of addressing I/O devices as memory locations, commonly called "memory mapped" I/O, it is also possible to communicate via a field of up to 4096 bits. This field is referred to as the "Communications Register Unit" (CRU). The addition of some hardware allows the CRU to be employed as a versatile command-driven I/O interface. Input and output bits can be addressed individually or in fields of from 1 to 16 bits. Five TMS9900 instructions are available to control the CRU. They can set (SBO), reset (SBZ), or test (TB), any bit in the CRU array, or move data between memory and CRU data fields (LDCR and STCR). The TMS9900 obtains the CRU bit address for any CRU operation by adding a base address stored in the workspace register 12 to a signed displacement specified in the instruction.

A programmable systems interface (TMS9901) and an asynchronous communications controller (TMS9902) can be connected to a TMS9900 to provide parallel and serial data transfer respectively. Both units are designed to communicate with the TMS9900 via the CRU. The TMS9901 contains a programmable interval timer and the necessary hardware to interface the external interrupt lines to the microprocessor. The TMS9902 also contains a programmable interval timer.

It is not thought necessary to cover the architecture of the TMS9900 in any greater detail in this thesis. Fur-

ther information on the TMS9900, the TMS9901, and the TMS9902 can be found in Texas Instruments data books; references (4.8), (4.9), and (4.10) respectively. Reference (4.8) provides information on the instruction set, the instruction times, and the addressing modes of the TMS9900. Details of the implementation of typical TMS9900 microprocessor systems are available in other Texas Instrument publications such as the "9900 Family Systems Design and Data Book" (4.11), and the "TMS9900 Family System Development Manual" (4.12).

4.5.2 Basic Microprocessor System Implementation

Having chosen the TMS9900 microprocessor, a choice had to be made whether to construct a special purpose system using a TMS9900, some RAM, and the necessary interfacing circuitry, or to purchase a ready-made and tested development kit. The special purpose system approach is cheap and the system can be exactly tailored to the requirements of the motor drive, but the approach can involve very difficult debugging problems because both the hardware and the software can be faulty. The development kit solution is more expensive in initial capital costs, but it does reduce debugging problems since the only hardware required (if any) is some interfacing between the kit and the motor system (inverter, position sensor, etcetera). It was therefore decided to use a Texas Instruments TM990/100M-1 microcomputer board. This comprises a TMS9900 CPU; a TIM9904 system clock; 256 x 16 bits of RAM (expandable to 512 x 16 bits); 1k x 16 bits of TMS2708 eraseable programmable read-only memory (EPROM, expandable to 2k x 16 bits) containing a TIBUG monitor program; a TMS9901 parallel I/O port providing 16 bits of programmable I/O and 6 dedicated interrupts (the full microprocessor 15 external interrupts capability is available by suitable programming of the I/O ports); and a TMS9902 programmable serial I/O port which provides, under the TIBUG control, an RS-232-C interface for the connection of a terminal or visual display unit.

Two programmable interval timers are available on the board incorporated in the TMS9901 and TMS9902 circuits. The bus structure on the microcomputer board is such that it can be interconnected with additional memory boards, interface circuitry, and so forth. The system is supplied with 450ns access time RAM and so the microprocessor has to execute one wait state per memory access: this increases the instruction execution times. All timings in this thesis assume "1 wait state per memory access". Full details of the TM990/100M-1 microcomputer board are given in the "TM990/-100M Microcomputer User's Guide" (4.13). A listing of the TIBUG software is given in the "TM990/401-1 TIBUG Monitor Listing" (4.14).

The requirement for fast interrupt service routines, in order to achieve high speed autopiloted operation, led to a decision to write the microprocessor programs in assembly language, rather than use a high level language (such as Basic) and then compile it into TMS9900 machine code. The Texas Instruments 9900 Cross Assembler was available on the college computer. The Cross Assembler could have been used to develop the assembly language programs and to generate the corresponding machine code. However, there was not a computer line conveniently close to the laboratory area, and so downloading of the assembled machine code from the college computer to the TM990/100M-1 board posed a problem. This was overcome by choosing to use the TM990/302-1 Software Development Board. This board interconnects with the microcomputer board and it replaces the monitor functions performed by the TIBUG software. The TM990/302-1 provides a complete independent software development system, and it includes an audio cassette tape recorder interface which allows program storage to be easily achieved. Using the TM990/302-1 it is possible to text edit, assemble, debug, and execute assembly language programs. In addition, it can be used to program a variety of EPROM's. The text editor enables an assembly language "source" file to be created, edited, and stored on cassette tape. The two-

pass symbolic assembler is then used to create the machine code "object" file, which is also stored on tape. The object code is then loaded into the microcomputer memory for execution and debugging. The symbolic assembler is not as powerful as the full Texas Instruments Cross Assembler but it is adequate for the development of "small" machine code programs. The size of program that can be conveniently handled by the TM990/302-1 board depends on the amount of RAM connected to the system. Full details of the TM990/302-1 board are given in the "TM990/302 Software Development Board User's Guide" (4.15) and the "TM990/-302 Hardware User's Guide" (4.16).

In order to use the TM990/302-1 board with the TM990/-100M-1 microcomputer, it is necessary to expand the RAM on the microcomputer board to its maximum size of 512 x 16 bits. This RAM (located between addresses 0_{16} and $3FF_{16}$) is used by the TM990/302-1 to run the monitor programs in, and is also reserved for interrupt and XOP vectors. The 2k x 16 bits of RAM provided on the TM990/302-1 is available for general program use and it is located between addresses 1000_{16} to $1FFE_{16}$. The software development programs are located in EPROMs located between addresses $E000_{16}$ to $FFFF_{16}$.

Having based the microprocessor system on the TM990/-100M-1 and TM990/302-1 boards, it was also thought to be desirable to expand the system memory capability to insure against unforeseen future requirements. A TM990/201-41 memory expansion board was obtained and this was populated with 2k x 16 bits of RAM (TMS4045) and 6k x 16 bits of EPROM (TMS2716). The EPROM was intended for storage of the motor control programs once they had been fully debugged. Full details of the TM990/201-41 board are given in the "TM990/201 and TM990/206 Expansion Memory Boards" handbook (4.17).

A TM990/512 prototyping board was also obtained, so

that the interfacing logic necessary between the microcomputer board and the motor system, could be mounted on a system compatible board. The four system boards (TM990/-100M-1; 302-1; 201-41; and 512) were housed in a TM990/510 four slot card cage. This provides the necessary system bus interconnections for the boards to operate together. Details of the prototyping board and the card cage are given in "TM990/512 TM990/513 Universal Prototyping Boards" (4.18) and "TM990/510 TM990/520 Card Cages" (4.19) respectively.

The TM990 system boards require three voltage supply rails; +5 volts, +12 volts, and -12 volts. The Texas Instruments power supply (TM990/518) was considered to be poor value for money and so a suitable power supply was constructed for the system. The current demands for the Texas Instruments manufactured boards are shown in fig. 4.5.

	CURRENT DEMAND		
	+5VOLT	+12VOLT	-12VOLT
TM990/100M	1·4A	0·2A	0·1A
TM990/302	1·5A	0·05A	0·05A
TM990/201*	1·2A	0·2A	0·1A
TOTALS ⟶	4·1A	0·45A	0·25A

* suitable estimates made for extra memory added.

FIG. 4.5. POWER SUPPLY REQUIREMENTS OF THE TEXAS INSTRUMENTS MICROPROCESSOR SYSTEM

The power supply was designed to supply a maximum of 5.0 amps at +5 volts, 1.0 amp at +12 volts, and 1.0 amp at -12 volts. This allowed a maximum possible consumption by the prototype interface board of 0.9 amps at +5 volts, 0.55 amps at +12 volts, and 0.75 amps at -12 volts.

The circuit diagram of the power supply is shown in fig. 4.6. The circuit was based on the standard integrated regulators that are readily available. A mains filter was included to reduce supply borne interference, and a cooling fan was added to increase the reliability of the components. The transformer for the +5 volt supply was made from a kit. This was so that a 10 volt secondary voltage could be obtained; this reduces the power dissipation in the +5 volt regulator. Simple crowbar circuits were designed for each regulated supply rail to prevent over voltages reaching the microprocessor. As a further safety precaution the three supply rails were relay interlocked. This ensures that the microprocessor system only receives power if all supplies are present. The voltage rails were brought out of the power supply via a 9 way D socket; several pins were paralleled so that the +5 volt and ground rails had the necessary current rating.

The only problem experienced with the power supply was that occasionally the -12 volt regulator tended to produce a momentary over voltage condition when suddenly loaded: that is, when the output toggle switch was closed (it was concluded that the regulator disliked having to charge the -12 volt decoupling capacitors on the system boards). The over voltage caused the crowbar protection thyristor to trigger. However, the situation could always be cleared by switching off the mains supply temporarily to release the crowbar. The power supply has functioned reliably for many hours over a two year period.

The basic microprocessor system was completed by the addition of a Perkin Elmer 550B VDU. This terminal can

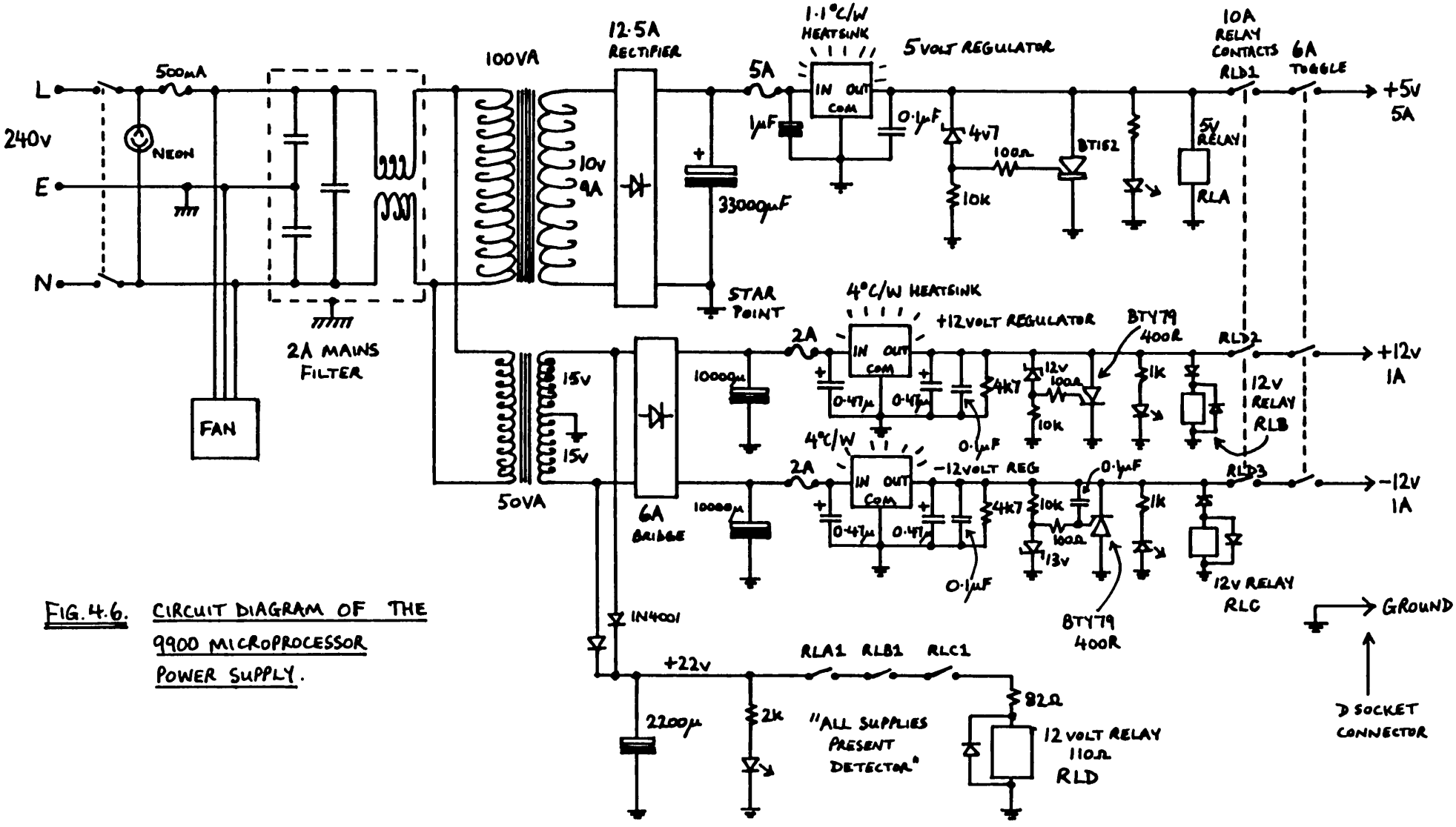


FIG. 4.6. CIRCUIT DIAGRAM OF THE 9900 MICROPROCESSOR POWER SUPPLY.

operate at a baud rate of 9600 and possesses some useful cursor control features, which were utilised in the initialisation routines for the motor control programs. Full details of the VDU are given in its user manual (4.20). Hard copy listings of the source and object programs were obtained on a Teletype 43 KSR teleprinter. This could operate at 300 baud and signals were routed to it via the "print" port of the VDU.

It is worth mentioning the additional cassette recorder control unit that was added to the system. The control unit was inserted into the connecting leads between the TM990/302-1 board and the cassette recorder as shown in fig. 4.7.

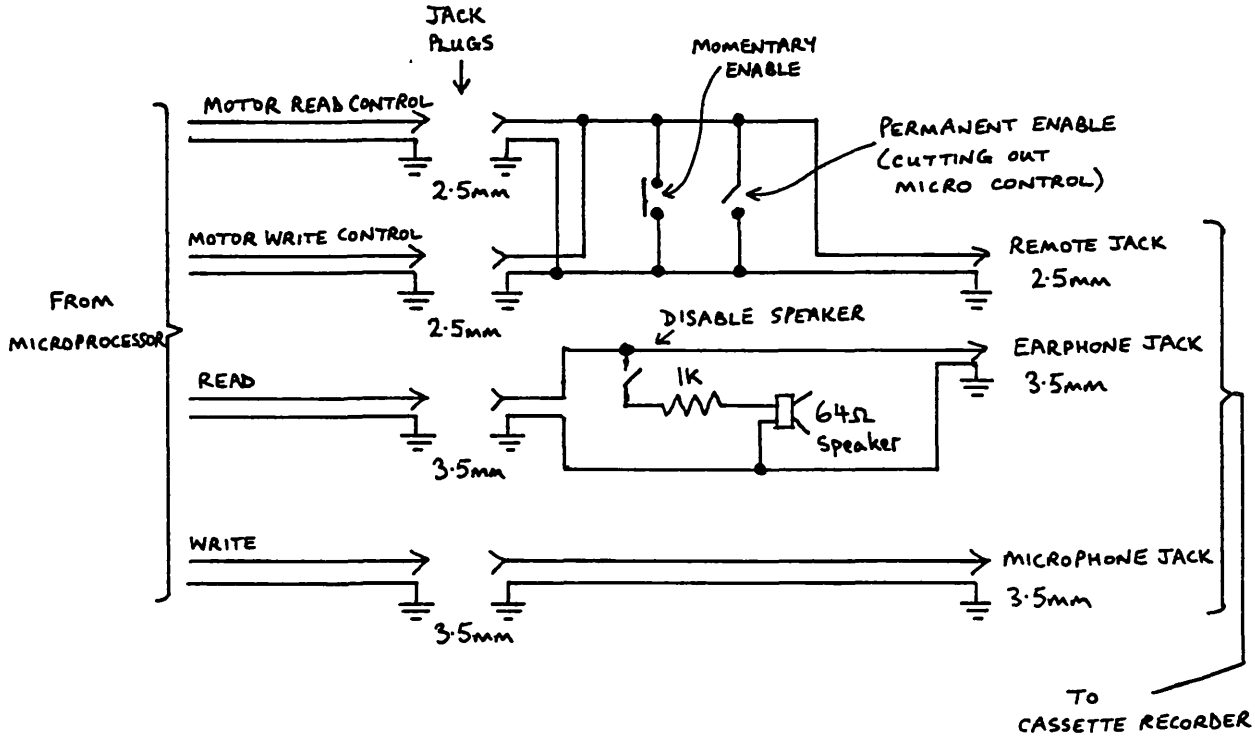


FIG. 4.7. CASSETTE RECORDER CONTROL UNIT

The switches allowed the cassette recorder to be operated independently of microprocessor control without needing to unplug the remote control jack plug from the recorder.

This facilitated the positioning of tapes using the fast forward and rewind functions. The loudspeaker permitted the monitoring of the replay signals going from the cassette recorder to the TM990/302-1 board (due to the peculiarities of the Sanyo M2511G cassette recorder used for the work, it was also possible to hear the signals as they were recorded). A photograph of the control unit connected to the cassette recorder is shown in fig. 4.8.

Finally, a photograph of the power supply unit with the basic card cage (on its side) is shown in fig. 4.9.

4.6 The Microprocessor System Interfacing Logic - Basic Interfacing Philosophy

Before the microprocessor interface logic could be designed, it was necessary to decide how many units within the motor system required communication to or from the microprocessor. Therefore, the first task was to decide on the basic system arrangement. The primary aim of the work was to implement microprocessor autopiloting of a synchronous motor at high speeds. The microprocessor can devote much more of its time to this principal task if the other tasks in the system are made short and simple. This can be achieved by the careful design of hardware circuitry to allow the simplification of the software.

It was therefore decided that the calculation of speed would be performed by a hardware 16 bit digital tachometer and full details of its design are given in Chapter 8. Communication of control commands and data during motor operation were simplified by the design of a "keypad" and data display unit. Full details of the keypad, its control logic, and the data display unit are given in Chapter 7. It should be noted that the "data available" and "data recall" interrupts that are discussed in Chapter 7 are referred to respectively as the "DATA" and "DISPLAY" inter-

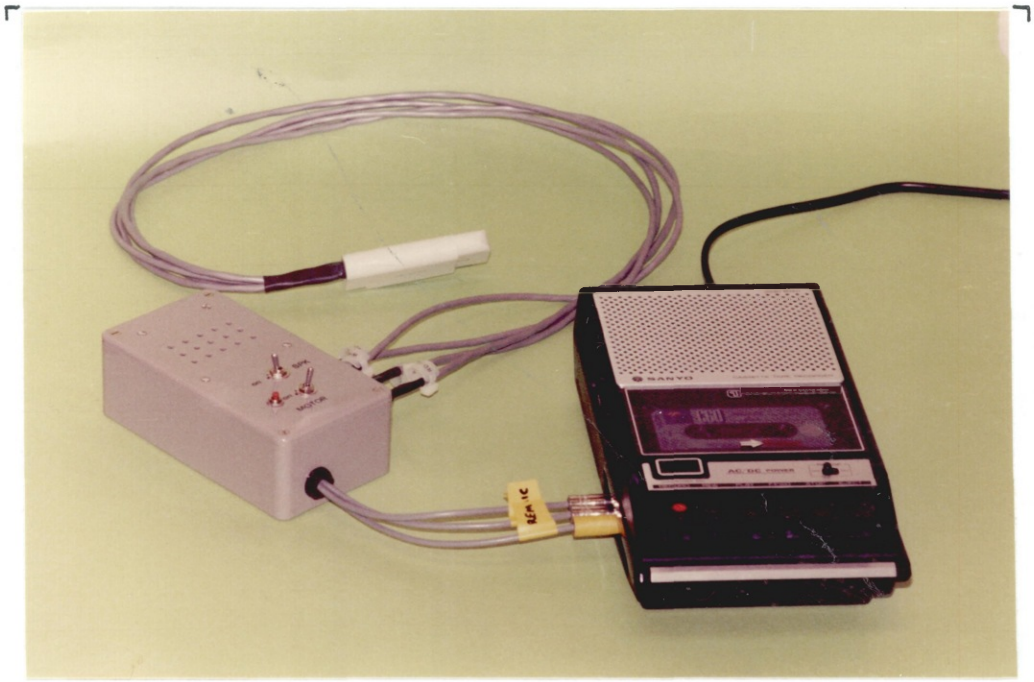


FIG. 4.8. THE CASSETTE RECORDER AND ITS CONTROL UNIT USED FOR PROGRAM STORAGE ON THE TMS9900 SYSTEM

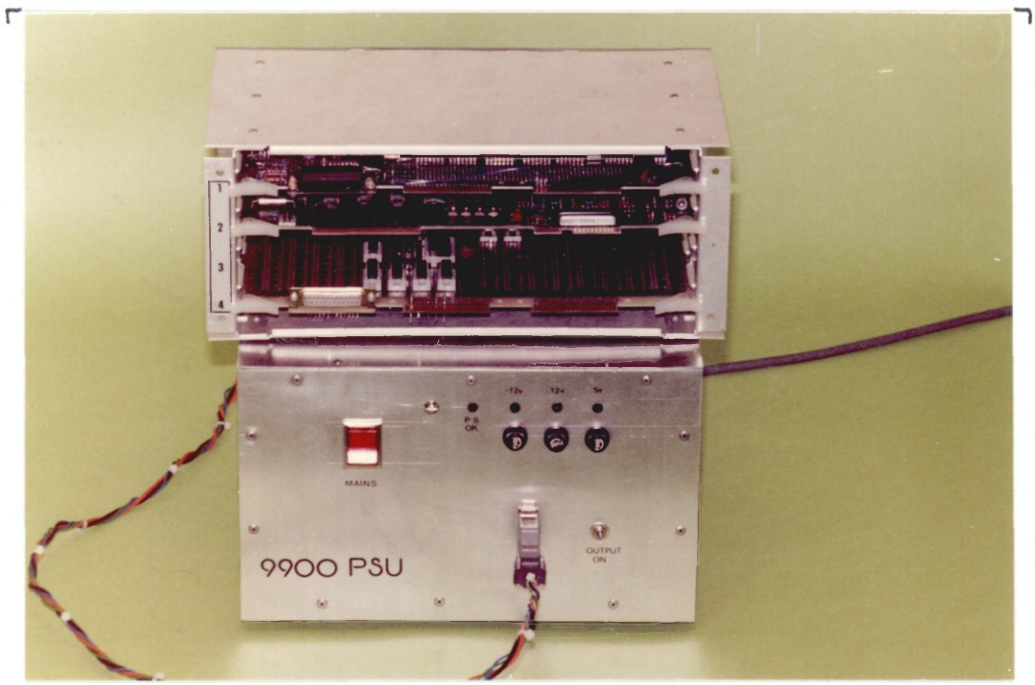


FIG. 4.9. THE MICROPROCESSOR CARD CAGE AND POWER SUPPLY UNIT (INTERFACE CIRCUIT BOARD IN SLOT 4 IS IN INCOMPLETE STATE)

rupts in this chapter. The shortened names for the interrupts do not describe the interrupt functions so accurately, but they have been used in this chapter for economy of words since they are referred to repeatedly. The MOSFET inverter and the digital to analogue control circuit for the ±60 volt inverter power supplies were both arranged to appear as simple digitally controlled peripherals to the microprocessor, as is explained in Chapter 5. Finally, the position sensor circuitry described in Chapter 6 was arranged to give incremental information in digital form.

The various system peripherals were designed to interconnect as shown in fig. 4.10. The microprocessor can only communicate with one peripheral at a time. Hence it is necessary to store the outgoing data to a particular peripheral on an interface latch so that the peripheral can act on the information continuously until the next microprocessor communication to it occurs. Likewise, data transmitted from peripherals to the microprocessor must be latched on the interface board and held until the microprocessor has an opportunity to examine the data.

A small amount of interfacing is necessary to handle interrupt signals. The TM990/100M-1 microcomputer board includes one TMS9901 parallel interface chip (configured between CRU software addresses 0100_{16} to $013E_{16}$). A TMS-9901 has 22 interrupt and I/O lines of which 6 are dedicated interrupt inputs, 9 are programmable as interrupt inputs or I/O ports, and 7 are dedicated I/O ports. A control word can be transmitted from the microprocessor to the TMS9901 via the CRU to enable the interrupt lines that are required. The TMS9901 assesses the priority of any interrupt signals present on the enabled interrupt lines, and it informs the TMS9900 of the highest priority interrupt that is pending. The TMS9901 is therefore very useful. However, external logic must be arranged to maintain an interrupt request on an interrupt line until the interrupt is acknowledged (i.e. executed). This is necessary because

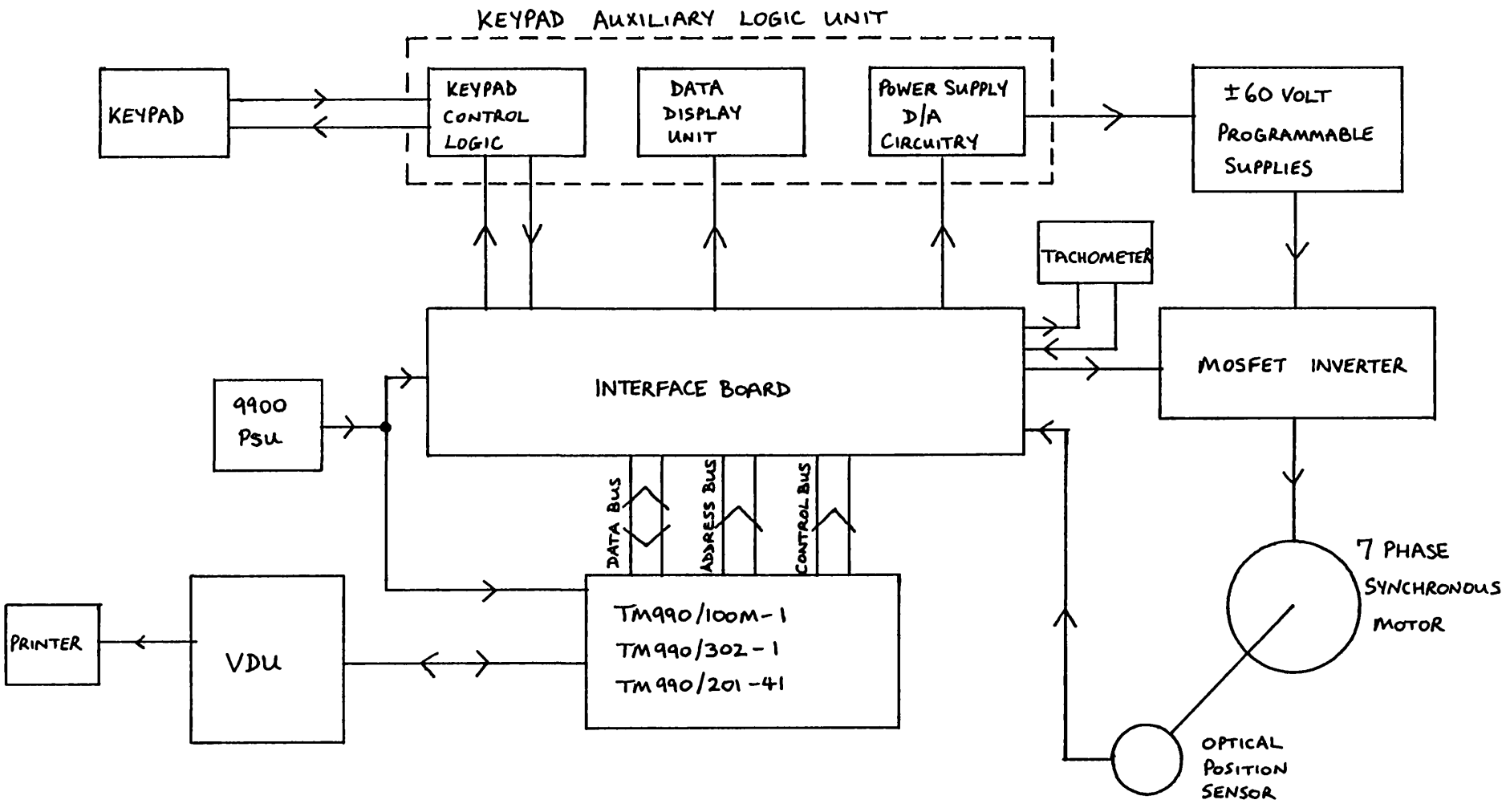


FIG. 4.10. SCHEMATIC OF THE MOTOR CONTROL SYSTEM.

an interrupt request may be denied for a long time while higher priority interrupts are being serviced: if the interrupt signal is in the form of a pulse it may therefore have disappeared by the time that the microprocessor can sample it. Hence, to ensure that even a momentary interrupt signal is correctly registered and serviced, it is necessary to latch the interrupt when it occurs and to clear the latch under microprocessor control when the interrupt is actually serviced.

In order to implement the basic autopiloting system shown in fig. 4.10 it is necessary to interface the microprocessor to the keypad, the data display unit, the ±60 volt power supply D/A circuitry, the MOSFET inverter, and the optical position sensor (the interfacing between the VDU and the CPU is handled on the microcomputer board). The minimum interfacing requirements (parallel data ports, miscellaneous control lines, and interrupt latches) for each peripheral are listed below.

(a) MOSFET Inverter

Output ports to inverter:

1 x 14 bit parallel "Inverter Control" port;
1 x 15 bit parallel "Inverter Enable" port;
1 x 1 bit "System Initialise Line".

(b) ±60 volt Programmable Power Supply Unit

Output ports to D/A circuitry:

1 x 15 bit parallel "Combined Control and Enable" port;
1 x 1 bit "System Initialise Line".

(c) Optical Position Sensor Unit

Input ports to microprocessor:

1 x 6 bit parallel "Rotor Position Code" port (for input of an absolute position code if required).

Interrupt handling logic:

2 interrupt latches to register SYNC and SEG pulses (for implementation of an incremental position code system).

Miscellaneous logic:

Suitable circuits to process the raw SYNC' and SEG' signals from the position sensor in a manner depending on the Forward/Reverse command routed from the keypad; the resulting SYNC and SEG signals are suitable for the desired direction of motor rotation. In addition, the circuits convert the SYNC and SEG signals into an appropriate binary code, thus allowing an incremental position sensor to be combined with a "sampling" microprocessor autopiloting strategy.

(d) Keypad

Input ports to microprocessor:

1 x 16 bit parallel "Keypad Data" port (to receive variable data from keypad);
1 x 4 bit parallel "Key Code" port (to identify either received variable data or requested display data);
1 x 1 bit "Forward/Reverse Command";
1 x 1 bit "Stop/Start Status".

Interrupt handling logic:

3 interrupt latches: the first to register changes in the Stop/Start Status line, the second to register the DATA interrupt (which occurs when new variable data is presented), and the

third to register the DISPLAY interrupt (which occurs when display of data is requested).
N.B. As already mentioned at the beginning of section 4.6, the DATA and DISPLAY interrupts are referred to more accurately in Chapter 7 as the data available and data recall interrupts respectively.

Output port to keypad:

1 x 1 bit "System Initialise Line" (referred to in Chapter 7 as the Keypad Initialisation Line).

(e) Data Display Unit

Output ports to display unit:

1 x 16 bit parallel "Combined Data and Key Code" port (16 bit data word and 4 bit key code are transmitted separately).
2 x 1 bit identification signals: one to indicate to the display unit that the "Combined Data and Key Code" port is presenting 16 bit data; the other to indicate that a 4 bit Key Code is being output.

(f) Tachometer

Input port to microprocessor:

1 x 16 bit parallel "Speed Reading" port.

Output port to tachometer:

1 x 16 bit parallel "Estimate of Speed" port (for possible implementation of a software based Torok Valis tachometer, discussed very briefly in Chapter 8).

Interrupt handling logic:

1 interrupt latch to register binary rate multiplier output pulses, BRMOUT (this is also for possible implementation of a software based Torok Valis tachometer).

Miscellaneous signal lines to tachometer:

1 x 1 bit "Speed Signal" derived from SEG pulses by appropriate divider logic.

The interfacing requirements listed in (a) to (f) above are not trivial, although one signal (System Initialise Line) is common to several peripherals. A decision had to be made about the best way to implement the I/O interfacing. As mentioned earlier in section 4.5.1, it is possible to achieve I/O on the TMS9900 either by using the unique CRU facility or by using memory mapped ports.

The use of several TMS9901 parallel interface circuits configured in the CRU bit field is a relatively cheap and easy way of implementing all the parallel I/O requirements but there is a significant drawback: communication between a TMS9900 and a TMS9901 is by serial data transfer. This considerably reduces the necessary interconnections but the transfer time of a word of data is relatively long. Parallel data words of up to 16 bits are transferred from the TMS9901 to memory by using the STCR instruction, and in the opposite direction by using the LDCR instruction. The time taken to execute the instructions depends on the number of bits transferred. For a 16 bit transfer the times are a maximum of 24.6μs for STCR and 21.6μs for LDCR (these times are with 450ns memory requiring 1 wait state per access). The main operation of the microprocessor in an autopiloted motor system is to transfer control data to the inverter at frequent intervals. To perform this with the CRU providing I/O would involve the use of the LDCR instruction. Since this is a relatively slow instruction it was thought that it might place a limit on the speed which the 7 phase motor could achieve. Therefore consider-

ation was given to memory mapped I/O.

Memory mapped I/O ports are connected directly to the data bus of the microprocessor system. When the microprocessor wishes to transfer data from memory to an output port, it moves the data onto the data bus and sets up the appropriate address on the address bus. Decode logic associated with the output port is arranged to decode the address bus and control bus signals and at the appropriate time enable the output port so that it samples the information on the data bus. At all other times the output port ignores the information present on the data bus. Input of information to the microprocessor is slightly more complicated because only one device at a time may drive the data bus. Therefore, an input port must only drive the data bus when the microprocessor wishes to receive data from it; at all other times the port must be disconnected. Decode logic connected to the address and control buses determines when the input port is required to drive the data bus with its information. Data is moved between memory locations in the 9900 system by means of the MOV instruction. The maximum execution time for the MOV instruction is about 12.7μs (system set up for 450ns memory; 1 wait state per access).

Transfer of I/O data in a TMS9900 system is therefore significantly quicker if the memory mapped method is used rather than the CRU method. Against this must be set the slightly more complex logic needed to implement a memory mapped solution (decoded CRU address signals are present on the TM990/100M-1 board which permit the easy addition of up to 5 TMS9901 circuits). However, since the primary aim of the work was to achieve fast motor speeds, it was decided that the memory mapped method was the most suitable for the input and output of parallel data words.

Having chosen memory mapped I/O it was then necessary to consider the interrupt logic required. The TMS9901 on

the TM990/100M-1 microcomputer board can handle more than enough interrupts to implement the motor system proposed in fig. 4.10, but latches are necessary on the interrupt lines to ensure correct interrupt servicing (as explained earlier in this section). It was decided to latch each interrupt request using an edge triggered D type flip flop as shown in fig. 4.11.

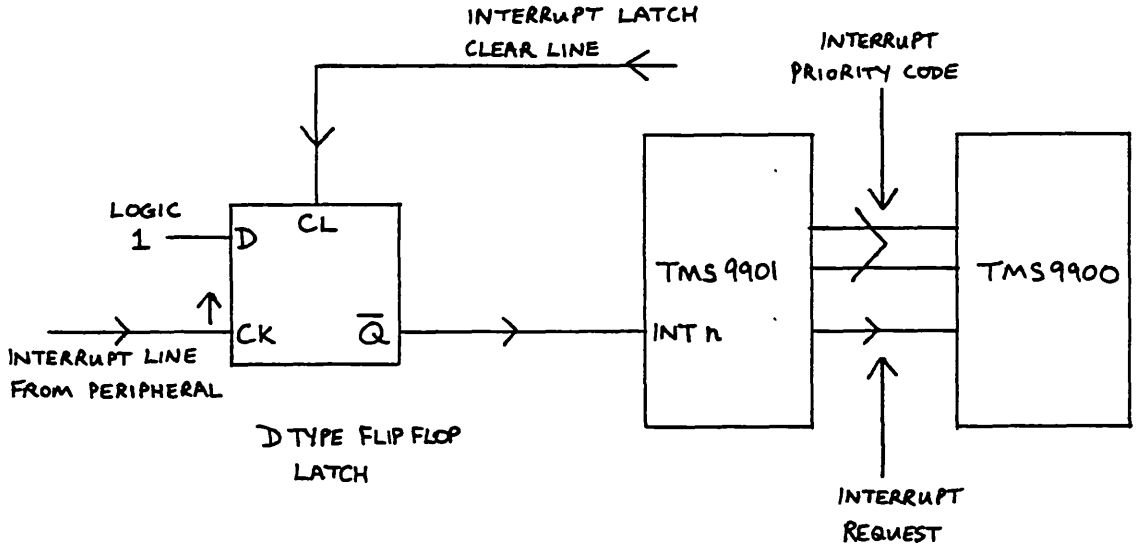


FIG. 4.11 BASIC INTERRUPT LATCH METHOD

The interrupt inputs of a TMS9901 respond to active low signals. Therefore, the $\overline{Q}$ output of the D type flip flop is connected to the desired TMS9901 interrupt line (INT'n' say). The peripheral requests an interrupt by taking the D type flip flop clock input from logic 0 to logic 1, thus causing the $\overline{Q}$ output to go to logic 0. At some time during the subsequent servicing of the interrupt the microprocessor must acknowledge that it has received the request. It does this by generating a signal to clear the flip flop (i.e. to restore $\overline{Q}$ to logic 1). The flip flop is then ready to latch another interrupt. Unfortunately, the TMS-9900 has no interrupt acknowledge signals. Suitable interrupt acknowledge signals can be generated in three ways:

(1) By using CRU bit instructions (SBO, SBZ) to control

I/O lines in a TMS9901. Each I/O line can be configured to control the clear input of a D type flip flop.

(2) By decoding appropriate addresses on the address bus. This is memory mapped interrupt acknowledgement. Each flip flop clear line is associated with an individual memory location. Any microprocessor instruction in which the address of a particular flip flop clear line is specified causes the clear line to be activated.

(3) By using some of the external instructions available on the TMS9900. There are five external instructions (CKON, CKOF, RSET, IDLE, and LREX) that allow user-defined external functions to be initiated under program control. The majority of these instructions are already used on the TM990/100M-1 microcomputer board but CKON and CKOF are available for use, thus providing two ready made software controlled interrupt acknowledge lines.

The choice of method for the generation of interrupt acknowledge signals was made on the basis of the time required to complete the clearing of a flip flop, since a short clearing time helps to minimise the execution times of the interrupt service routines. A flip flop is cleared by taking the clear input to its active level (logic 0 for TTL flip flops) momentarily. If a CRU approach is used with I/O line 'n' of a TMS9901 connected to the flip flop clear input, the software needed to perform a clear is basically:

```
SBZ  n    (set line 'n' to logic 0)

SBO  n    (set line 'n' to logic 1)
```

The time taken to execute the two instructions is a total of 9.324μs. This assumes that the necessary CRU software base address is already present in workspace register 12. If the CRU base address is not correct it has to be set up

before the clear operation can be performed, thus increasing the interrupt clearing time.

If a memory mapped approach is used for interrupt acknowledgement, it is necessary to use an instruction which can access the desired memory location and hence provide signals on the address and control buses that can be decoded into a "clear" signal. Some instructions take longer to execute than others; a suitable minimum duration instruction is CLR. The procedure to clear an interrupt latch located at memory address $ABCD_{16}$ would be:

```
CLR  @>ABCD     (clear memory location ABCD16)
```

The time taken to execute this instruction is 7.326μs.

If either of the available external instructions (CKON or CKOF) are used for interrupt acknowledge signals the typical software is simply:

```
CKON     (generate CKON pulse)
```

or

```
CKOF     (generate CKOF pulse)
```

The time taken for either of these instructions is 4.329μs.

It can be seen from the above software examples that the external instructions provide the fastest method of interrupt clearing. It was therefore decided to use CKON and CKOF for the most frequently occurring interrupts, which in the motor system happen to be SYNC and SEG, produced by an incremental position sensor unit. It was then decided to use memory mapped interrupt acknowledgement for all the remaining interrupts in the system, thus helping to minimise the execution times of their respective service routines.

The final choice to be made about the system inter-

facing was concerned with the method of implementing miscellaneous control and status lines. For example, a means was required to input the forward/reverse information from the keypad. In addition, a "System Initialise Line" was needed which could be set under software control to indicate to the peripherals that the microprocessor was initialised, but which would be immediately reset (disabling the peripherals) in the event of the microprocessor being reset. The I/O lines of the TMS9901 on the TM990/100M-1 board were ideal for these types of tasks. If data is present on a TMS9901 I/O line it can be sampled using the CRU "test bit" (TB) instruction, and subsequent series of instructions executed by the microprocessor can be made dependent on the logic level sampled. This is therefore an ideal way of testing flag lines such as the forward/-reverse input. The TMS9901 on the microcomputer board also permits direct implementation of a "System Initialise Line". The I/O lines are all programmed as input ports whenever the TM990/100M-1 system reset is activated. All the 22 interrupt and I/O lines of the TMS9901 on the microcomputer board are connected to the +5 volt rail via 4.7kΩ pull-up resistors. Hence when programmed as inputs, the ports are held at logic 1 unless external signals pull them to logic 0. An individual I/O line is programmed as an output port by writing data to it from the microprocessor (using for example the SBO and SBZ instructions). Once programmed as an output, a port remains in the output mode until the TMS9901 device is reset. This can be initiated by either software or hardware generated reset signals. A software reset is achieved either by writing a zero to the $\overline{\text{RST2}}$ bit when the TMS9901 is in clock mode (see reference 4.9), or by executing the RSET instruction. A hardware reset is obtained simply by operating the reset switch on the microcomputer board. The TMS9901 I/O ports are therefore ideal for the implementation of a "System Initialisation Line". One I/O port is connected to the peripherals as shown in fig. 4.12. When the system is in a reset state the I/O line is an input and the pull-up resistor holds the

line at +5 volts. When the microprocessor wishes to initialise the system it writes a logic 0 to the I/O line. This programs the line as an output and pulls the "System Initialisation Line" to logic 0. This line allows the keypad, ±60 volt power supplies, and the inverter to operate. If a reset of the microprocessor occurs the I/O line is immediately reconfigured as an input, and hence the "System Initialisation Line" returns to its inactive logic 1 state, thus safeguarding the peripherals.

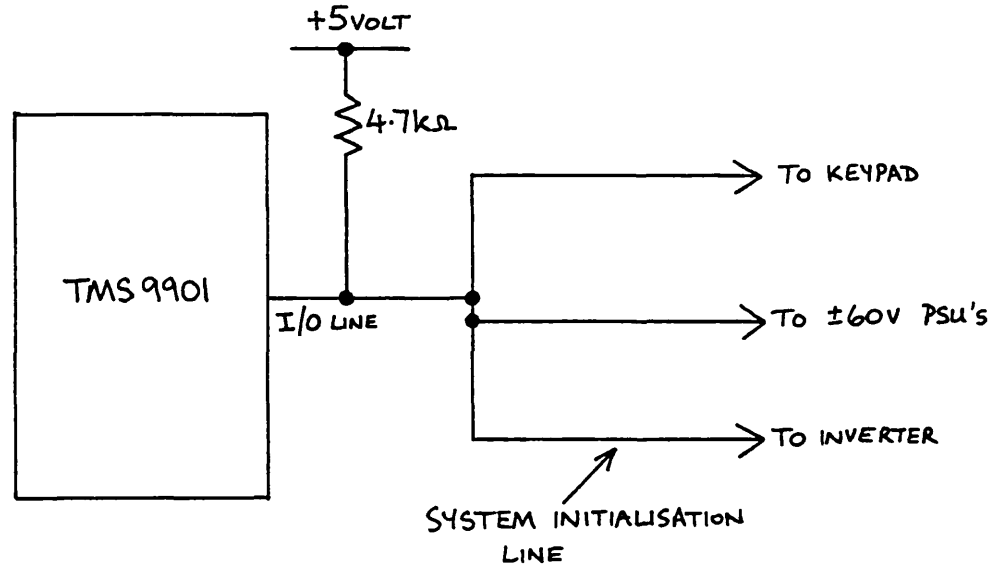


FIG. 4.12 SYSTEM INITIALISE LINE ARRANGEMENT

4.7 Practical Implementation of System Interfacing

Having outlined the chosen interfacing methods it is now possible to discuss the practical implementation of the interfacing. Virtually all of the interfacing circuitry was mounted on the TM990/512 universal prototyping board. All signals between the microcomputer board and the prototyping board, with the exception of CKON and CKOF, were routed via the Card Cage back plane and connector P1 on the

interface board. The TM990 microprocessor system boards are intended to be interfaced with peripherals via low power schottky TTL logic (LSTTL). Therefore, wherever possible the interface circuitry was implemented in LSTTL rather than in standard TTL. This decision was actually vital in order to keep the power consumption of the interfacing within the available limits: the maximum currents available for the interfacing circuits from the +5 volt, +12 volt, and -12 volt power supplies were 0.9 amps, 0.55 amps, and 0.75 amps respectively (as shown in section 4.5.2). Details of the LSTTL chips used in the interfacing are given in "The TTL Data Book for Design Engineers" (4.21). The limited current sourcing capability of the microprocessor power supply was a major reason for the decision to give independent power supplies to some of the peripherals in the motor system (e.g. the inverter and the keypad).

The chances of damage occurring to the inverter were reduced by using active low signals between the interface board and the control port of the inverter. The TTL inputs in the inverter assume logic 1 states if the interconnecting leads are broken. Hence any phase that has its control signals disconnected goes into an inactive state. The "System Initialisation Line" was similarly arranged to be active-low to protect against broken connections.

The total numbers of input ports, output ports, and interrupt latches required by the basic motor system (as listed in section 4.6) are six, eight, and six respectively. A total of four input ports and five output ports are multi-bit and so are best dealt with by memory mapped interface ports, whilst two input ports (forward/reverse and stop/start) and three output ports (the system initialise line and the combined data and key code port identification signals) are single bit and can, if necessary, be dealt with by the TMS9901 I/O ports. Two of the interrupt ports can be cleared by means of the CKON and CKOF instructions and so there is a minimum requirement for four memory mapped

interrupt acknowledge signals. The foregoing assessment of I/O requirements led to the conclusion that a minimum of thirteen memory mapped I/O port control signals had to be generated.

4.7.1 Memory Mapped Interface Control Circuit

The "Interface Control" circuit shown in fig. 4.13 provides sixteen memory mapped port control signals. It is based around a 74154 TTL 4 line-to-16 line decoder (IC12). Prior to the design of the Interface Control circuit it had been decided to use 74LS374 Octal D-type edge triggered flip flops for the multi-bit input ports, 74LS373 Octal D-type transparent latches for the multi-bit output ports, and 74LS74 Dual D-type positive edge triggered flip flops for the interrupt latches. The Interface Control circuit was designed to provide suitable signals to control five 74LS374 input ports, five 74LS373 output ports, and six 74LS74 interrupt latches. To make the port addresses more meaningful the circuit was arranged to be active for memory addresses $C000_{16}$ to $C01E_{16}$ inclusive; the "C" part of the address representing "Communication". The Interface Control circuit was arranged so that the inputs connected to the address and control buses were buffered to limit the loading of the buses.

The basic operation of the Interface Control circuit is very simple. The decoding logic (IC24(a), IC12, and IC11) examines the fifteen most significant lines of the address bus (A0.B to A14.B inclusive; the most significant bit being A0.B) and the memory enable line ($\overline{\text{MEMEN}}$.B) to determine if one of the sixteen memory word locations from $C000_{16}$ to $C01E_{16}$ has been selected. In addition, the Interface Control circuit examines the DBIN.B and $\overline{\text{WE}}$.B signals on the control bus. The DBIN.B signal is active when the microprocessor expects to receive input data from the data bus; hence it is active whenever an instruction requires a memory read during its execution. The $\overline{\text{WE}}$.B signal is active

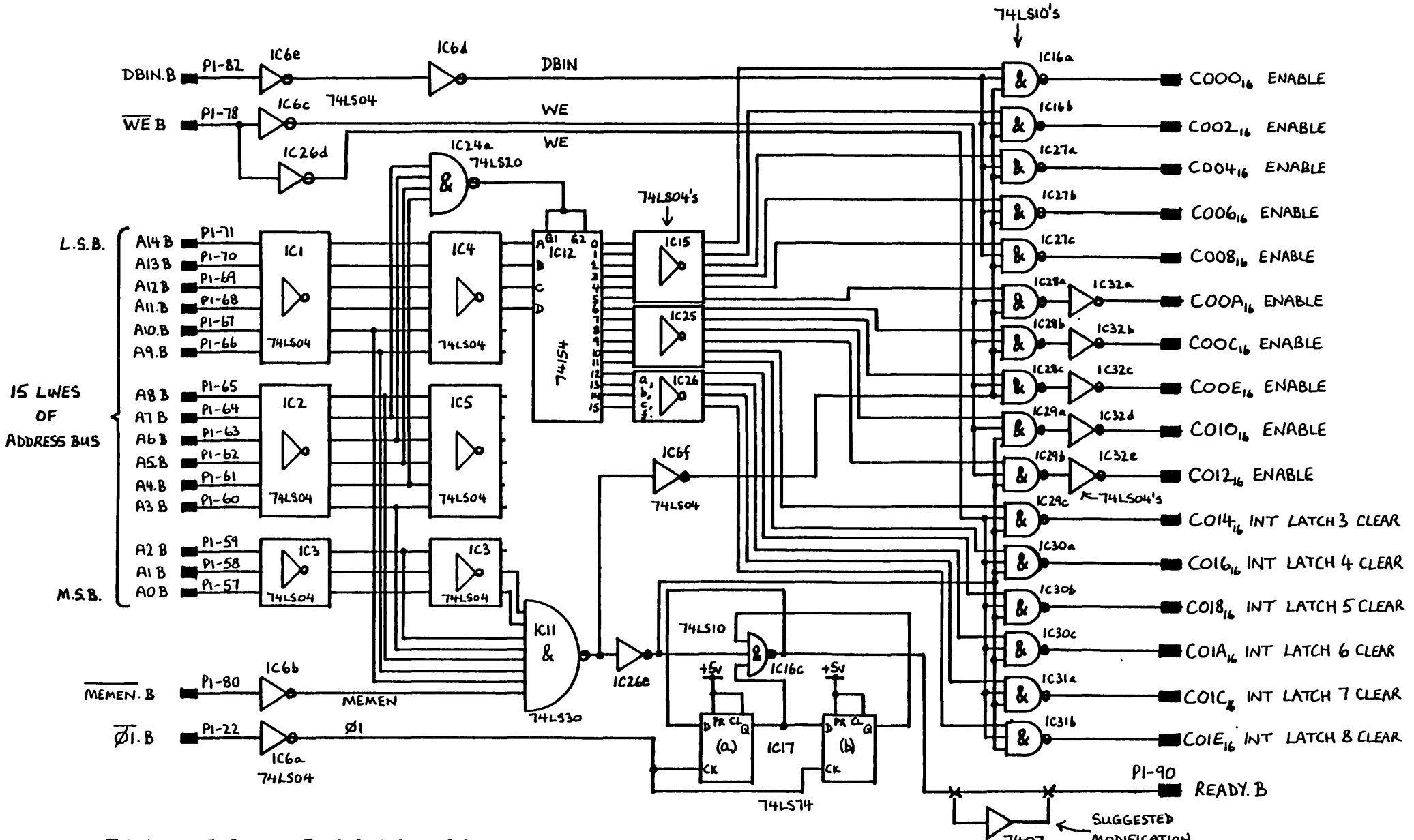


FIG. 4.13. THE INTERFACE CONTROL CIRCUIT.

when the microprocessor wants to output data onto the data bus for transfer into memory. Therefore, $\overline{WE}$.B is active whenever an instruction requires a memory write during its execution. There are sixteen 74LS10 triple input nand gates forming the output stages of the Interface Control circuit (IC16(a),(b); IC27(a),(b),(c); IC28(a),(b),(c); IC29(a),(b),(c); IC30(a),(b),(c); IC31(a),(b)). The three inputs of each nand gate are respectively connected to the buffered output of IC11, a separate buffered output of IC12, and either the buffered DBIN or the buffered WE lines. When the microprocessor sets up a valid address on the address bus and enables $\overline{MEMEN}$.B, only one of the sixteen nand gates has two inputs at logic 1 (all the rest have only one input at logic 1). The nand gate is fully enabled when its third input is taken to logic 1 by either DBIN.B or $\overline{WE}$.B going active during the memory cycle. The nand gates that are required to control the input ports must be enabled by the DBIN.B signal and an instruction capable of activating DBIN.B must therefore be used to select such gates. In a similar way the nand gates that are required to control the output ports must be enabled by the $\overline{WE}$.B signal and so an instruction capable of activating $\overline{WE}$.B must be used to select these gates. The nand gates that generate interrupt latch clear signals can be operated by either the DBIN.B or $\overline{WE}$.B signal: it was decided to configure them as outputs and so they were connected to the buffered WE line.

Nand gates IC16(a), IC16(b), IC27(a), IC27(b), and IC27(c) are activated by DBIN.B and so are intended for the control of input ports: they are selected by addresses $C000_{16}$, $C002_{16}$, $C004_{16}$, $C006_{16}$, and $C008_{16}$ respectively. Nand gates IC28(a), IC28(b), IC28(c), IC29(a), and IC29(b) are activated by $\overline{WE}$.B and so are intended for the control of output ports: they are selected by addresses $C00A_{16}$, $C00C_{16}$, $C00E_{16}$, $C010_{16}$, and $C012_{16}$ respectively. Nand gates IC29(c), IC30(a), IC30(b), IC30(c), IC31(a), and IC31(b) are also activated by $\overline{WE}$.B and they are intended

for interrupt latch clearing: they are selected by addresses $C014_{16}$, $C016_{16}$, $C018_{16}$, $C01A_{16}$, $C01C_{16}$ and $C01E_{16}$ respectively.

Finally some explanation is necessary regarding the function of the 74LS74 D-type flip flops (IC17(a), IC17(b)) and the 74LS10 nand gate IC16(c). The Texas Instrument microcomputer boards (TM990/100M-1; 302-1; 201-41) used for the project were supplied with "slow" (450ns access time) RAM and EPROM as standard. For successful memory access it is therefore necessary for the TMS9900 to enter 1 wait state during each memory access. The I/O interfacing logic does not suffer from such access delay times and so the TMS9900 could communicate with the memory mapped I/O ports without the need for a wait state on each access. This would actually speed up the execution of input or output routines very slightly. However, it was decided to be cautious and arrange the I/O ports to cause 1 wait state on each access, thus making the I/O appear to the TMS9900 as standard 450ns memory. Therefore, IC17(a), IC17(b), and IC16(c) were configured to generate 1 wait state whenever one of the I/O ports was selected. The D-type flip flops (IC17(a), IC17(b)) are clocked by the microprocessor clock signal $\overline{\emptyset 1}$.B and the circuit forces the READY.B line of the TM990/100M-1 board to go to logic 0 whenever a memory location between $C000_{16}$ and $C01E_{16}$ is selected, thus requesting a wait state from the TMS9900. Further details of wait states can be found in reference (4.11) pp 4-15 to 4-20 inclusive. The wait state logic incorporated in the Interface Control circuit does work but an important oversight was made at the time of its design. Both the TM990/-302-1 and TM990/201-41 boards contain their own wait state logic because they are populated with 450ns access time memory. The READY.B lines from these boards are intended to be wire-ored together to form a composite READY.B signal for the TM990/100M-1 board. To achieve this the READY.B signals on both boards are produced by <u>open collector</u> logic gates, and there is a 220Ω pull-up resistor on the READY.B

input of the TM990/100M-1 board to complete the OR function. Therefore, the choice of a 74LS10 nand gate to generate the Interface Control READY.B signal was wrong on two counts. Firstly, the signal should have been buffered onto the READY.B line of the system control bus via an open collector buffer. The consequence of not doing this is that each time the system READY.B line is taken to logic 0 by either the TM990/302-1 board or the TM990/201-41 board, the totem-pole output of IC16(c) is also pulled to logic 0 even though it wishes to stay at logic 1. Secondly, the 220Ω pull-up resistor on the TM990/100M-1 READY.B input line was selected by Texas Instruments on the basis that it would be driven by 74S22 open collector nand gates (as used on the TM990/-302-1 and TM990/201-41 boards to generate their READY.B signals). The 74LS10 nand gate (IC16(c)) is not guaranteed to be able to sink the 20mA current that flows when it attempts to pull the READY.B line to logic 0, whereas the 74S22 gates can do so. Therefore, it is possible that the Interface Control circuit wait state logic could never actually request a wait state if it was unable to pull READY.B to logic 0. This in itself would not be a problem since the interfacing can respond fast enough to manage without a wait state. However, the main cause for concern is the fact that the 74S22 gates on either the TM990/302-1 or TM990/201-41 board can pull down the output of IC16(c) to logic 0 and hence could damage or destroy its totem pole output. Since the design mistake was only spotted during the writing of this thesis, it can only be concluded that either: (a) IC16(c) survived the numerous abuses on its output (possible because the output is not pulled down for long periods) and so was able to request wait states for the I/O interface circuitry when required; or (b) it was damaged and no wait states were requested by the Interface Control circuit. However, it can be stated categorically that the system as a whole operated with no problems and so the requests for wait states by the other boards in the system were obviously still being received by the micro-computer board. For future use the wait state circuit in

fig. 4.13 should either be changed to the form used by Texas Instruments (for example see page 3-6 of reference 4.16) or it could be modified by the addition of a 7407 open-collector buffer as indicated on fig. 4.13.

The memory map for the complete microprocessor system with the "Communication" ports located between $C000_{16}$ and $C01E_{16}$ is shown in fig. 4.14.

The Interface Control circuit signals were allocated to the various I/O ports and interrupt latches as follows:

Memory Mapped Input Addresses (Input Ports)

$C000_{16}$ - Tachometer 16 bit "Speed Reading" input port.

$C002_{16}$ - Keypad 16 bit "Keypad Data" port.

$C004_{16}$ - Keypad 4 bit "Key Code" port.

$C006_{16}$ - Data Display Unit reset line (used to allow generation of suitable identification signals for information carried on "Combined Data and Key Code" bus; see Display port diagram for further details).

$C008_{16}$ - Optical Position Sensor Unit 6 bit "Rotor Position Code" port.

Memory Mapped Output Addresses (Output Ports)

$C00A_{16}$ - Tachometer 16 bit "Estimate of Speed" port (for software tachometer implementation - never actually used).

$C00C_{16}$ - Inverter 14 bit "Inverter Control" port.

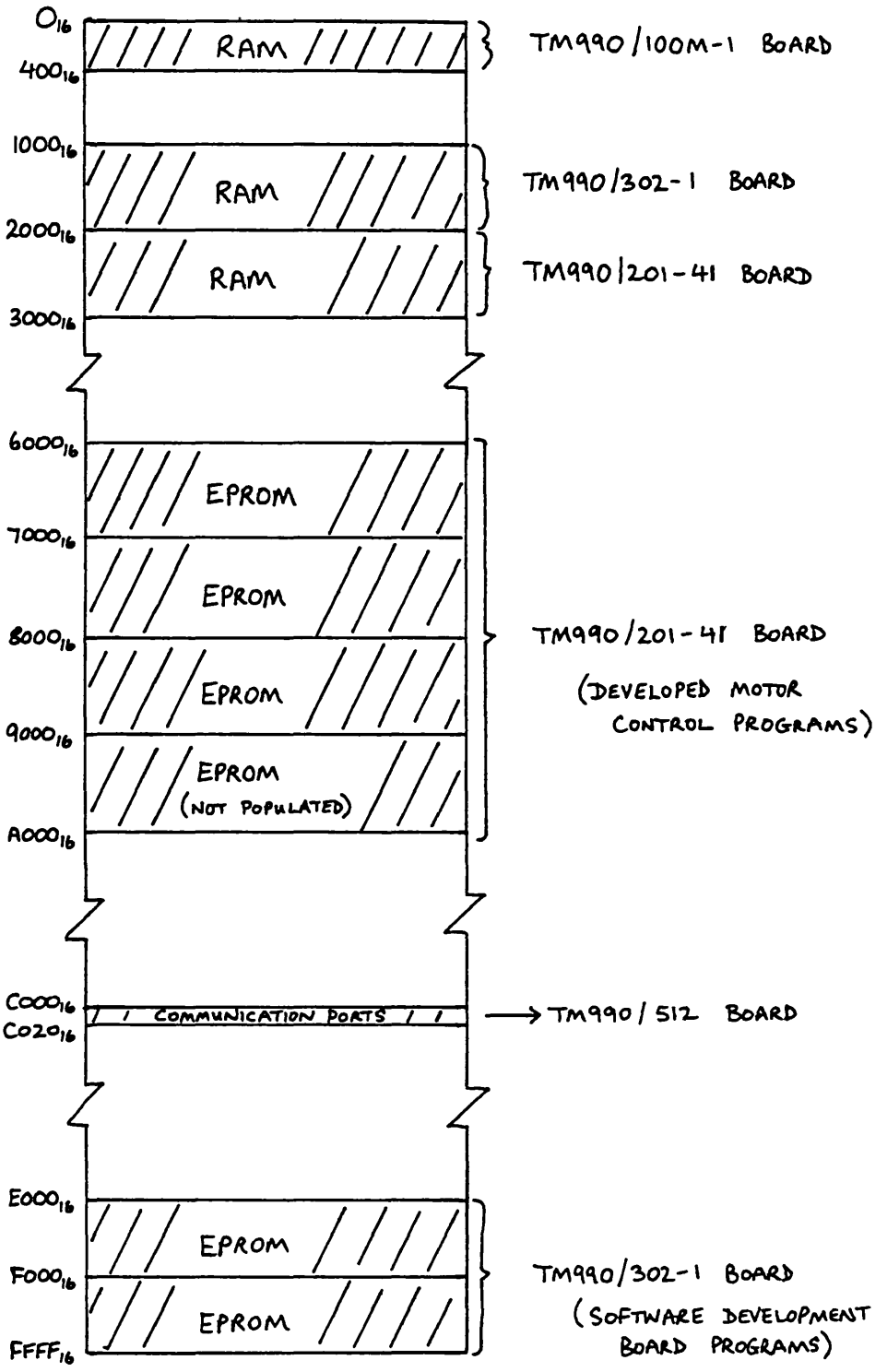


FIG. 4.14. THE MEMORY MAP FOR THE MOTOR CONTROL MICROPROCESSOR SYSTEM

$C00E_{16}$ - ±60 volt Power Supply Unit 15 bit "Combined Control and Enable" port.

$C010_{16}$ - Inverter 15 bit "Inverter Enable" port.

$C012_{16}$ - Data Display Unit 16 bit "Combined Data and Key Code" port.

Memory Mapped Output Addresses (Interrupt Clear Signals)

Note the interrupt latch numbers are NOT associated with the TMS9900 interrupts of the same number.

$C014_{16}$ - Interface board interrupt latch number 3 clear.

$C016_{16}$ - Interface board interrupt latch number 4 clear.

$C018_{16}$ - Interface board interrupt latch number 5 clear.

$C01A_{16}$ - Interface board interrupt latch number 6 clear.

$C01C_{16}$ - Interface board interrupt latch number 7 clear.

$C01E_{16}$ - Interface board interrupt latch number 8 clear.

4.7.2 The Provision of all Remaining I/O Requirements

The external instructions and the TMS9901 I/O ports were utilised as detailed below.

External Instructions

CKON - Interface board interrupt latch number 1 clear.

CKOF - Interface board interrupt latch number 2 clear.

TMS9901 I/O Lines

Both the microcomputer board and the prototyping board have P1 edge connectors which allow them to be plugged into the card cage back plane. By doing this the boards are interlinked, and the circuits on the prototyping board can gain access to the TMS9901 on the microcomputer board. Not all of the I/O lines of the TMS9901 are brought out to the P1 connector of the microcomputer board: only those which are programmable as either interrupts or I/O lines are available on the P1 connector (i.e. $\overline{\text{INT7}}$/P15, $\overline{\text{INT8}}$/P14,... ... to $\overline{\text{INT15}}$/P7). Some of these lines were needed for interrupt purposes as is shown shortly in sub-section 4.7.3. Therefore, the lines available for I/O use were somewhat restricted, but there were sufficient signals available as detailed below. The CRU bit numbers referred to in the following are based on a CRU base address of 120_{16} being in workspace register 12.

TMS9901 CRU bit 7 (CRU bit software address $12E_{16}$) -

Tachometer "Speed Reading" input port strobe signal: available on P1 pin 14 (TMS9901 $\overline{\text{INT15}}$/P7). The instruction SBZ 7 puts the signal to logic 0 and SBO 7 puts it to logic 1. This signal was made hardware switchable on the interface board and was used to clock data into the tachometer "Speed Reading" input port if an external strobe was not available. This facility allowed the "Speed Reading" input port to be used as a general purpose input port during development work tests. This signal was not used in the motor

control programs since the tachometer provides its own strobe signal.

TMS9901 CRU bit 8 (CRU bit software address 130_{16}) -

"System Initialise Line": available on P1 pin 11 (TMS9901 $\overline{\text{INT14}}$/P8). When set to logic 0 by the SBZ 8 instruction this line permits the keypad, inverter, and ±60 volt power supply to function. The SBO 8 instruction returns the line to logic 1, thus disabling the system.

TMS9901 CRU bit 9 (CRU bit software address 132_{16}) -

"Forward/Reverse Command" line: connected to P1 pin 12 (TMS9901 $\overline{\text{INT13}}$/P9). This line is controlled by the keypad forward/reverse line. The status of the line is tested by the TB 9 instruction when direction information is needed by the microprocessor.

TMS9901 CRU bit 10 (CRU bit software address 134_{16}) -

"Stop/Start Status" line: connected to P1 pin 9 (TMS9901 $\overline{\text{INT12}}$/P10). This line is actually the keypad stop/start interrupt line. The status of the line is tested by the TB 10 instruction when a stop/start interrupt is serviced.

4.7.3 The Interrupt Latch Circuitry and the Associated TMS9900 Interrupt Priorities

It can be seen by reference to sub-sections 4.7.1 and 4.7.2 that there are eight interrupt latch clear signals available from the interface circuits (six memory mapped latch clear signals plus two generated by external instructions). Each one of the eight latch clear signals was assigned to a motor system interrupt line. The eight incoming interrupt signals were called int1, int2, ... int8 respectively, but it should be noted that a particular numbered "int" line does not necessarily request the TMS-

9900 interrupt with the same number. The motor system interrupts are deliberately referred to by lower case letters (e.g. int1) to distinguish them from the TMS9900 interrupt priorities and the TMS9901 interrupt inputs, which are referred to by capital letters (e.g. INT3). The relationship between the motor system interrupt lines and the TMS9901 interrupt inputs that are actually activated is shown in fig. 4.15: the actual functions assigned to each interrupt in the motor system are also shown along with the interrupt acknowledgement "signal".

MOTOR SYSTEM "int" NUMBER	FUNCTION ASSIGNED TO "int" REQUEST	TMS9901 "INT" PRIORITY SELECTED	METHOD OF ACKNOWLEDGING "int" REQUESTS
1	SYNC	$\overline{\text{INT 1}}$	CKON
2	SEG	$\overline{\text{INT2}}$	CKOF
3	BRM *	$\overline{\text{INT5}}$	$CO14_{16}$
4	OVFL *	$\overline{\text{INT6}}$	$CO16_{16}$
5	CURRENT *	$\overline{\text{INT7}}$	$CO18_{16}$
6	STOP/START	$\overline{\text{INT8}}$	$CO1A_{16}$
7	DATA	$\overline{\text{INT9}}$	$CO1C_{16}$
8	DISPLAY	$\overline{\text{INT 10}}$	$CO1E_{16}$

* SEE TEXT FOR EXPLANATION

FIG. 4.15 SYSTEM INTERRUPT PRIORITIES, NAMES AND METHODS OF CLEARING (ACKNOWLEDGING).

Interrupt priorities INT3 and INT4 of the TMS9900 were not used: the programmable interval timer in the TMS9901 uses INT3 to indicate elapsed times, and the TMS9902 interrupt line on the TM990/100M-1 board is configured to use INT4. Hence, in order not to jeopardize future use of the interval timers, it was decided to avoid the use of these two interrupts in the motor system.

The SYNC and SEG lines were connected to int1 and int2 respectively so that they had the highest priorities available, as required to achieve high speed autopiloted operation. The next three interrupts (int3, int4, and int5) were connected for possible use in future system developments: int3 was designated for the "BRM" interrupt which would be generated if a software Torok-Valis tachometer was implemented in the manner proposed in Chapter 8; int4 was designated for the "OVFL" interrupt which could be used to detect if the microprocessor was approaching the end of its inverter control data table (discussed briefly in the software section of this chapter); and int5 was designated for "CURRENT", an overload interrupt which could be generated by the current sensor circuit described in Chapter 5. The relative priorities given to these three interrupts were somewhat arbitrary and it is not proposed to discuss this. Of the remaining three interrupt lines it was decided that the "STOP/START" interrupt should have the highest priority (int6), followed by the "DATA" interrupt (int7), with the lowest priority being assigned to the "DISPLAY" interrupt (int8). It is worth pointing out that although the STOP/START interrupt is of lower priority than the SYNC or SEG interrupts, it is nevertheless possible to stop the motor system rapidly in an emergency: this is achieved by pressing the microprocessor reset line which activates a $\overline{\text{INT0}}$ interrupt, thus resetting the "System Initialise Line" and disabling the inverter, keypad, and ±60 volt power supplies. Hence STOP/START does not need to be of high priority.

The latch circuitry for the interrupt lines is shown in fig. 4.16. The eight interrupt lines were made available to the motor system via the P3 connector of the prototyping board. Each interrupt line, with the exception of int6, is served by a 74LS74 D-type edge triggered flip flop. The latch circuitry for int6 is slightly more complicated because this interrupt is controlled by the Stop/Start Status line from the keypad. Both the rising and falling edge of the STOP/START signal must be sensed and latched as interrupts, and the level of the line following an interrupt determines whether a stop or start response from the microprocessor is required (line at logic 1 = stop; line at logic 0 = start). The rising and falling edges of the STOP/START signal are detected by a pair of 74LS74 D-type flip flops as shown in fig. 4.17 (there being insufficient room on fig. 4.16 to show the circuit in detail).

The glitch filters shown on the int6, int7, and int8 lines were necessary to prevent spurious interrupts, caused by cross-talk in the cables running between the keypad auxiliary logic unit and the interface board (as mentioned in Chapter 7). Each glitch filter consists of an RC network with a time constant of 47ns, which feeds a cascaded pair of 74LS14 schmitt trigger inverters. The 4.7kΩ pull-up resistors in the glitch filters were part of an integrated package of resistors (IC55).

The flip flop clear inputs were connected to the reset signals as shown in fig. 4.16, the "memory mapped" signals being routed from the circuitry shown in fig. 4.13. The $\overline{\text{CKON}}$ and $\overline{\text{CKOF}}$ signals, used to clear the int1 and int2 flip flops respectively, are generated on the TM990/100M-1 microcomputer board, and appear on that board at integrated circuit U20 ($\overline{\text{CKON}}$ at pin 10; $\overline{\text{CKOF}}$ at pin 9). Unfortunately the signals are NOT routed to any edge connector on the microcomputer board. They were therefore transmitted via a "flying lead" ribbon cable connected at either end by 14 pin dual-in-line plugs and sockets as indicated in fig. 4.18.

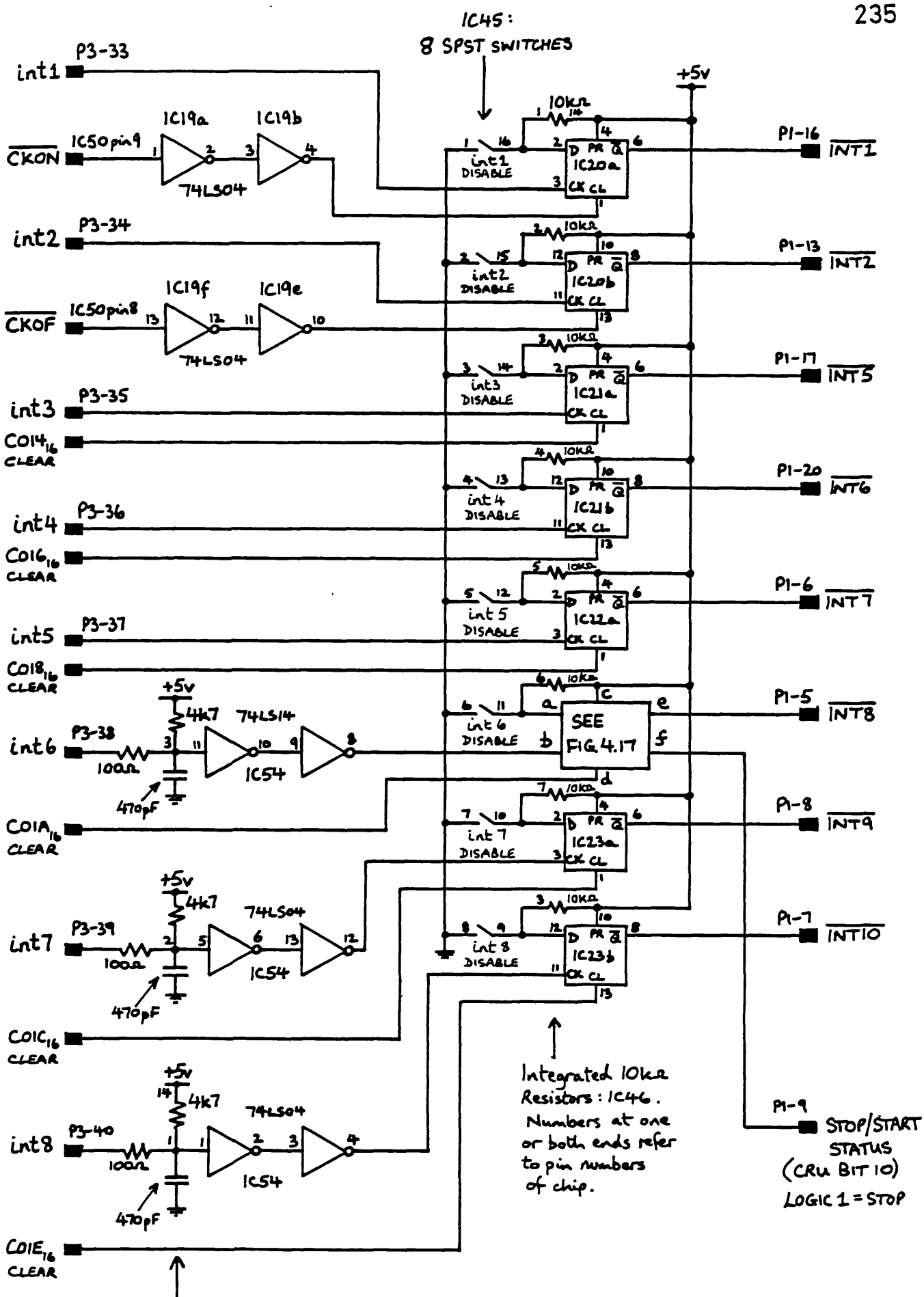
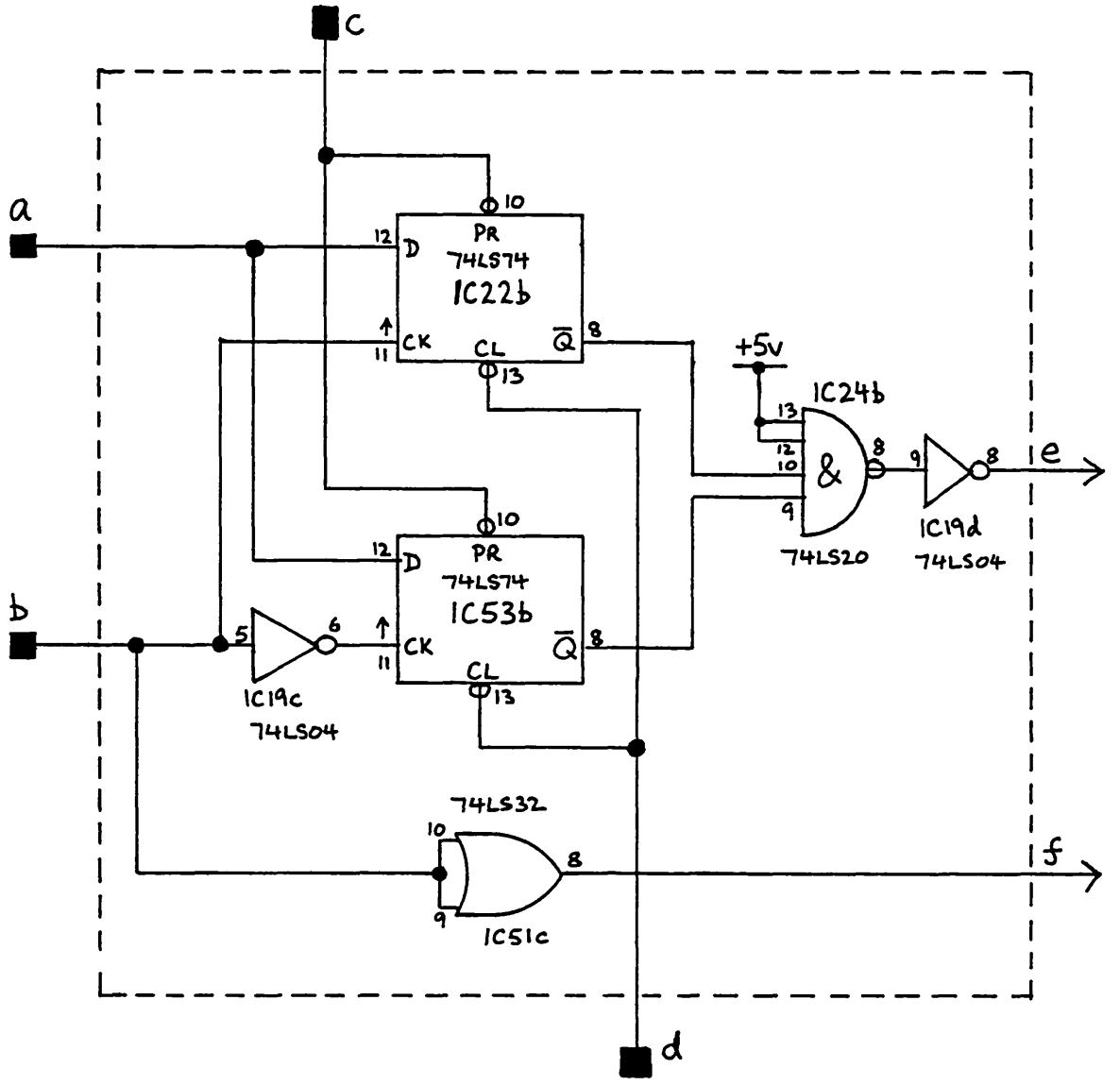


FIG. 4.16. THE INTERRUPT LATCH CIRCUITRY.



(REFER TO FIG.4.16 FOR DETAILS OF CONNECTIONS a TO f)

FIG.4.17. THE STOP/START INTERRUPT LATCH CIRCUIT

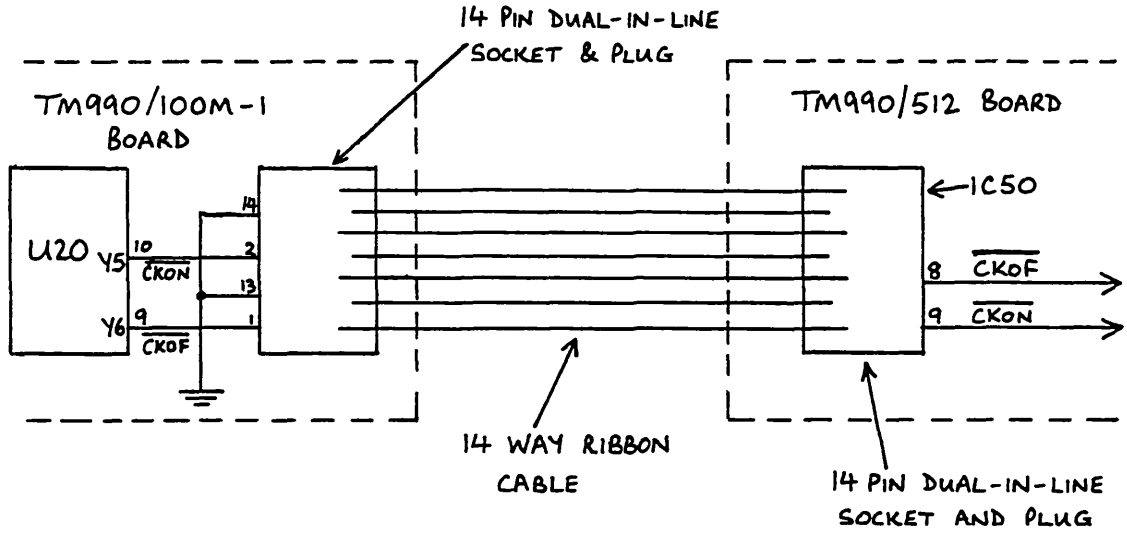


FIG. 4.18. ARRANGEMENT FOR TRANSMITTING THE CKON & CKOF SIGNALS TO THE int 1 & int 2 INTERRUPT CIRCUITRY

The 14 pin dual-in-line socket on the microcomputer board was mounted in the "prototyping area".

The flip flops can be individually disabled by a set of switches. This allows any unused interrupt line to be put into an inactive state. The switches are connected to the D inputs of the flip flops; when a switch is closed it puts a logic 0 level onto the associated flip flop D input, and hence the $\overline{Q}$ output of the flip flop is forced to remain at the inactive logic 1 state no matter how many interrupts occur at the clock input. A sub-miniature dual-in-line package (IC45) containing eight switches was used for the disable switch functions. Dual-in-line integrated 10kΩ resistors were used to pull-up the D inputs of the flip flops.

4.7.4 The Input and Output Port Organisation

The multi-bit input and output ports were connected to the data bus as shown schematically in fig. 4.19. The data bus is available on the prototyping board via connector P1. The four input ports are connected directly in parallel onto the data bus. The five output ports are connected in parallel but are not fed directly from the data bus; a buffer is used to limit the load imposed on the bus.

The input ports each require a strobe signal to clock data into them. They then hold the data for the microprocessor to inspect at its leisure. The tachometer provides a "New Speed Ready" signal for use as the "Speed Reading" input port strobe signal, and the optical position detector interface circuitry described later in this chapter supplies a "New Position Code Ready" strobe signal for the "Rotor Position Code" input port. However, no convenient signals were provided for the "Keypad Data" and "Key Code" input ports. Therefore, strobe signals were derived from the DATA and DISPLAY interrupt lines.

Full details of the input and output ports are given in the following sub-sections, and where appropriate the strobe signal generation circuits and any other associated circuits are shown. The input and output connections between the prototyping board and the motor system were made via intermediary D-type sockets mounted on an aluminium panel which was attached to the side of the card cage (henceforth referred to as the card cage side panel). The sockets allowed the system peripherals to be disconnected easily. The arrangement of the functions dealt with via the panel of sockets is indicated schematically in fig. 4.20.

The TACHO socket deals with all signals to and from the tachometer peripheral. The COUNTER and OPTO sockets

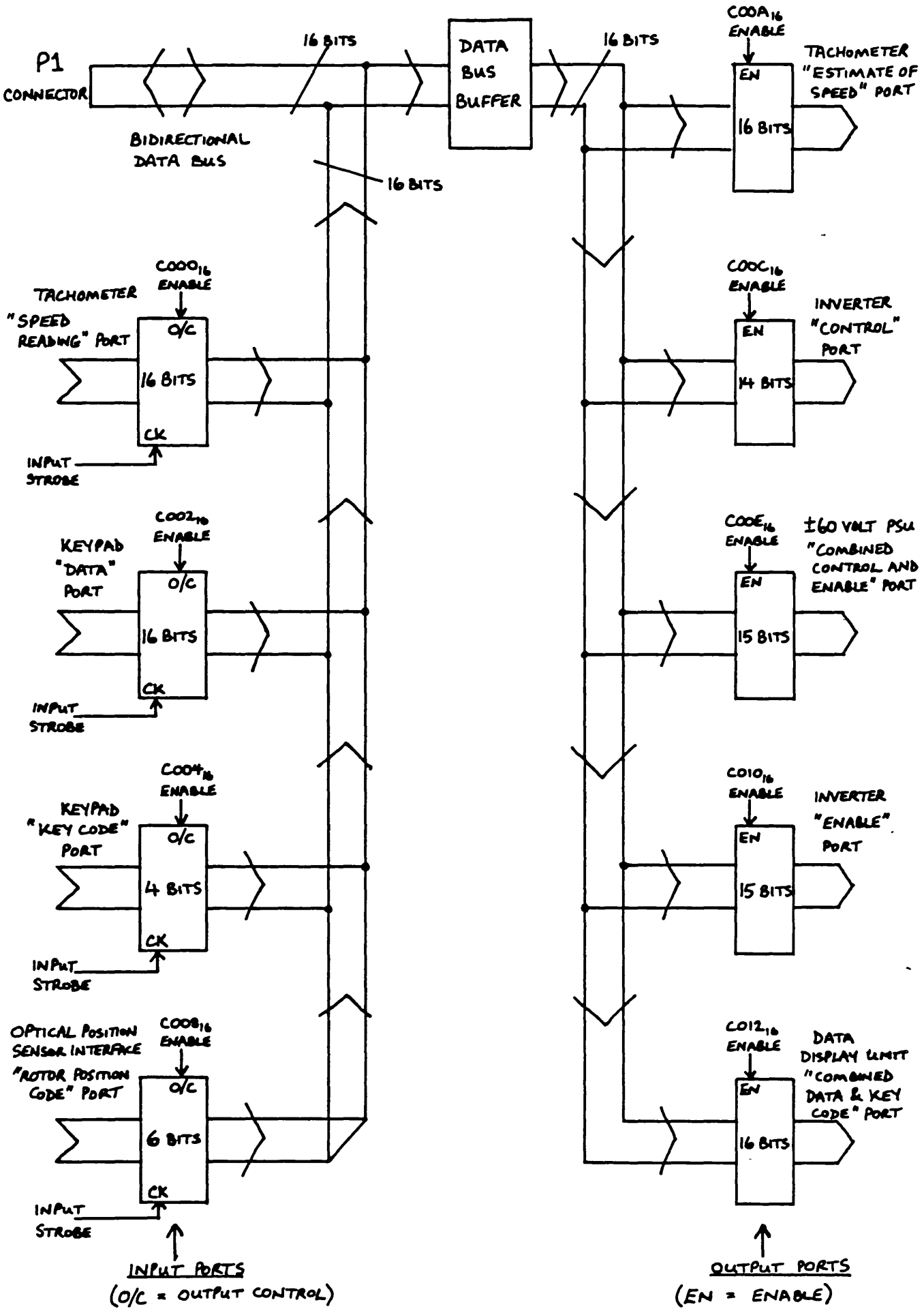
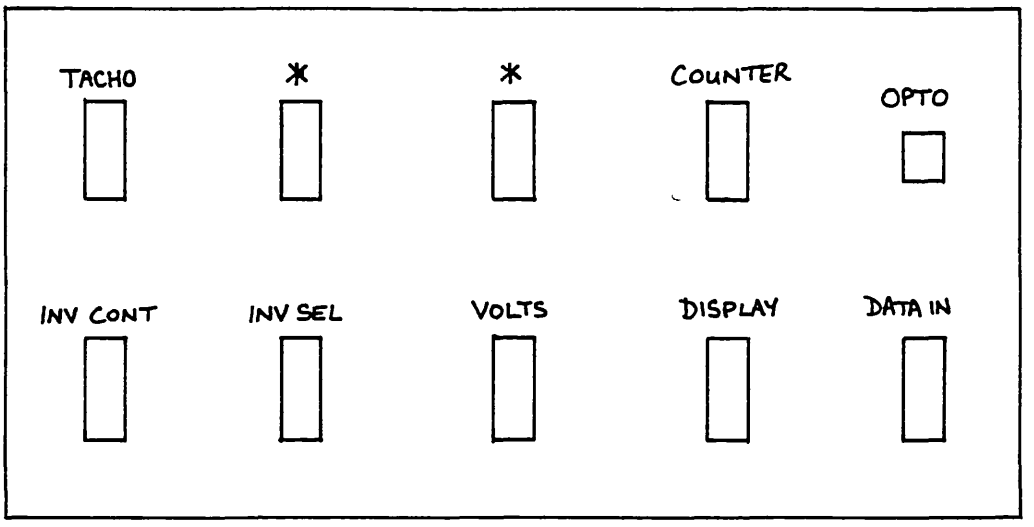


FIG. 4.19 PORT ARRANGEMENTS ON THE PROTOTYPING BOARD.



(* SPARE SOCKETS)

FIG.4.20 ARRANGEMENT OF THE SOCKETS ON THE CARD CAGE SIDE PANEL.

receive signals from the optical position detector interface circuitry: the "Rotor Position Code" is input via the COUNTER socket, and the SEG and SYNC interrupts are input via the OPTO socket. The INV CONT socket deals with the inverter "control" signals whilst the INV SEL socket deals with the inverter "enable" signals. The VOLTS socket carries the "combined control and enable" signals for the ±60 volt power supply. Finally, the DISPLAY socket carries the "combined data and key code" signals for the data display unit housed in the keypad auxiliary logic unit, and the DATA IN socket receives the keypad "data" and "key code" signals from the keypad logic (also housed in the auxiliary logic unit).

Details of the connections between the card cage side panel sockets and a particular port on the prototyping board are given in the appropriate sub-section dealing with that port. The OPTO socket was a 9 way D-type; all the

remainder were 25 way D-types.

4.7.4.1 Data Bus Output Buffer

The signals from the data bus going to the output ports were buffered by a pair of 74LS373 Transparent Latches as shown in fig. 4.21. The latches were permanently enabled in their transparent mode so that they simply acted as buffers.

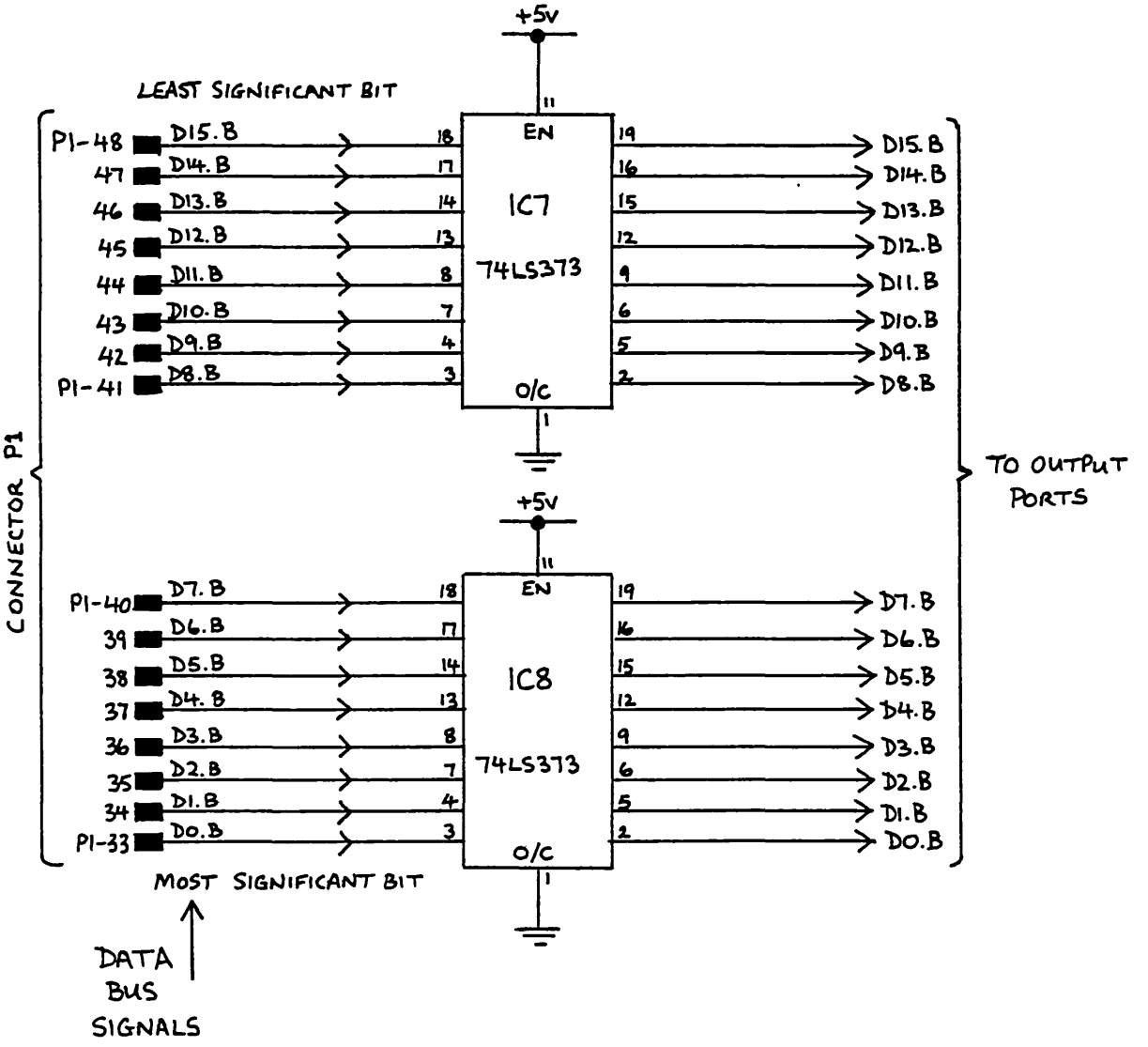


FIG. 4.21. DATA BUS OUTPUT BUFFER

4.7.4.2 The Tachometer Input and Output Ports

The "Speed Reading" input port and "Estimate of Speed" output port were connected to the tachometer by the same 16 bit bus via connector P2 on the prototyping board. This was possible because only one of the ports is active at any time ("Speed Reading" if hardware speed measurement is used; "Estimate of Speed" if software speed measurement is used). For clarity the two ports are shown in separate diagrams: the "Speed Reading" port in fig. 4.22, and the "Estimate of Speed" port in fig. 4.23.

The strobe signal for the "Speed Reading" input port is obtained either from the "New Speed Ready" signal produced by the tachometer or from the TMS9901 CRU bit 7 as shown in fig. 4.22. The dual-in-line switch S2 of IC49 is used to select "New Speed Ready" or CRU bit 7. The use of CRU bit 7 permits the input port to be used for other functions which do not provide suitable strobe information. The monostables (IC18(a) and (b)) provide a 2.5μs delay on the strobe signal. This guarantees that the new speed data generated by the tachometer is present at the port inputs when the strobe signal on the clock inputs of IC9 and IC10 operates.

The "Estimate of Speed" output port can be enabled by dual-in-line switch S1 of IC49. When enabled the port is used as part of a software based Torok-Valis tachometer system and the $\overline{\text{BRM SELECT}}$ line places the tachometer into the appropriate state (see Chapter 8 for a brief outline of the method).

The linking connections between the 25 way D-type socket P2 of the prototyping board and the TACHO socket on the card cage side panel are shown in fig. 4.24. The connections were made by ribbon cable terminated by insulation displacement D-type connectors. The use of ribbon cable resulted in a "twist" in the signal lines as can be seen

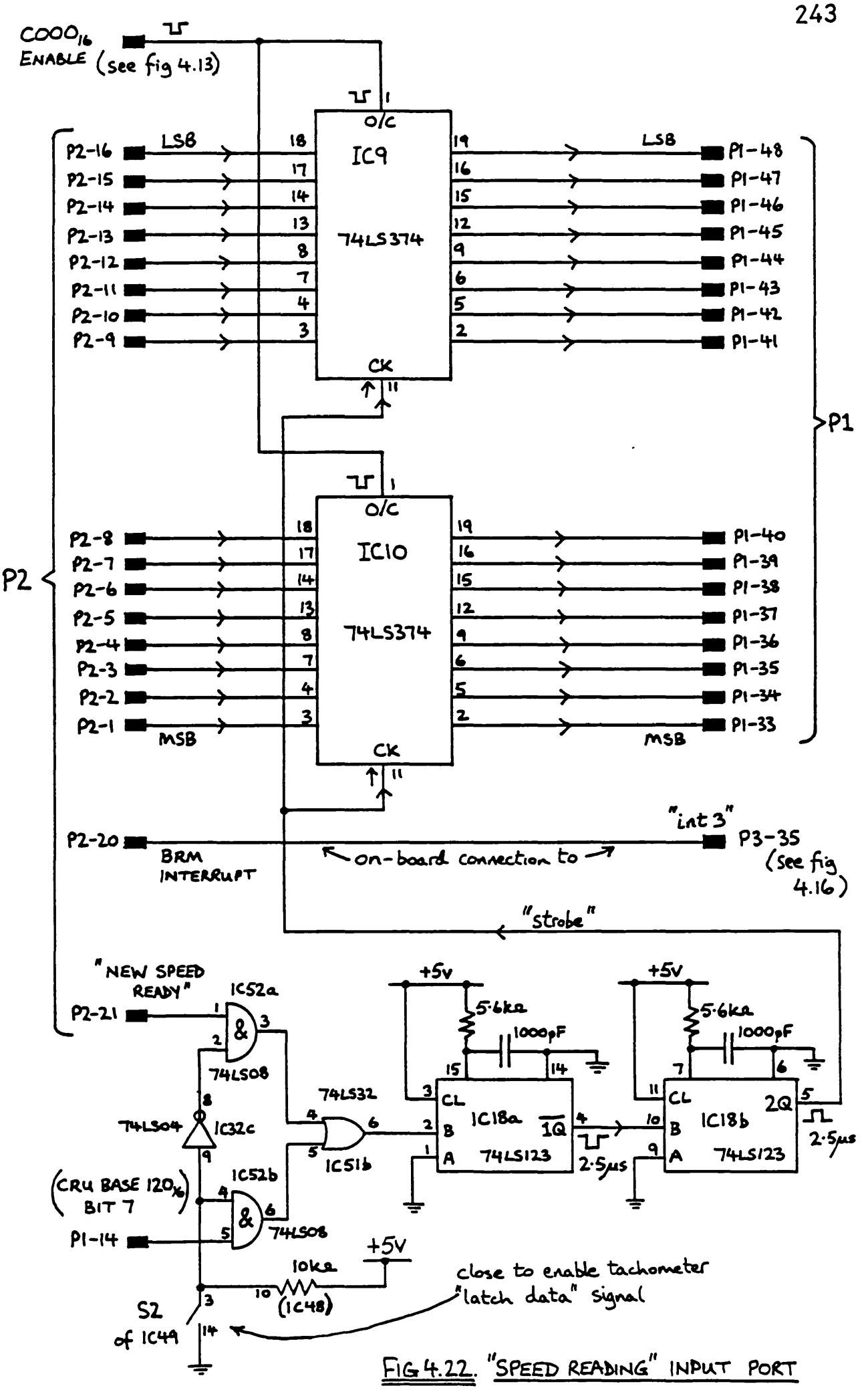


FIG 4.22. "SPEED READING" INPUT PORT

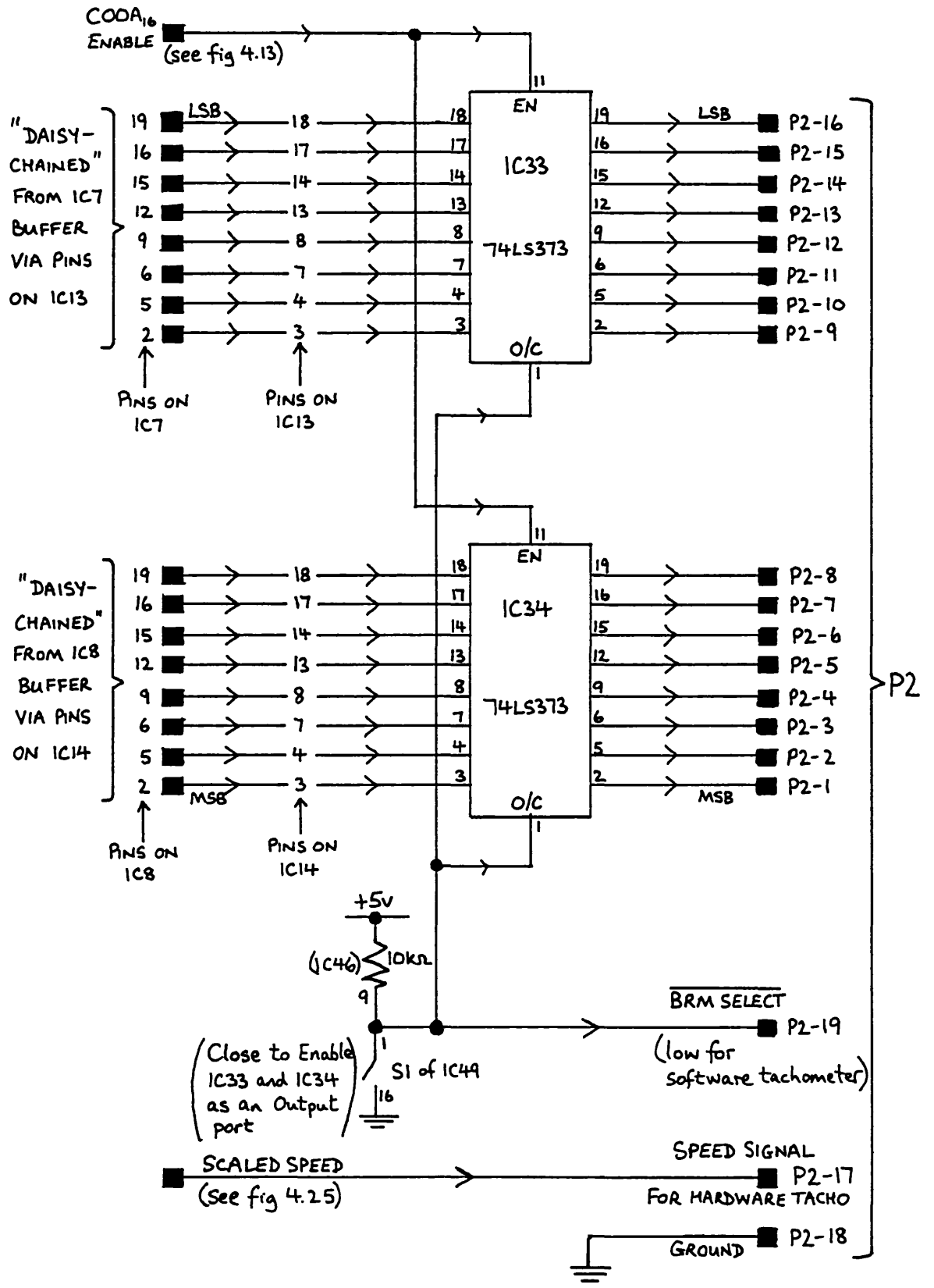


FIG. 4.23. "ESTIMATE OF SPEED" OUTPUT PORT.

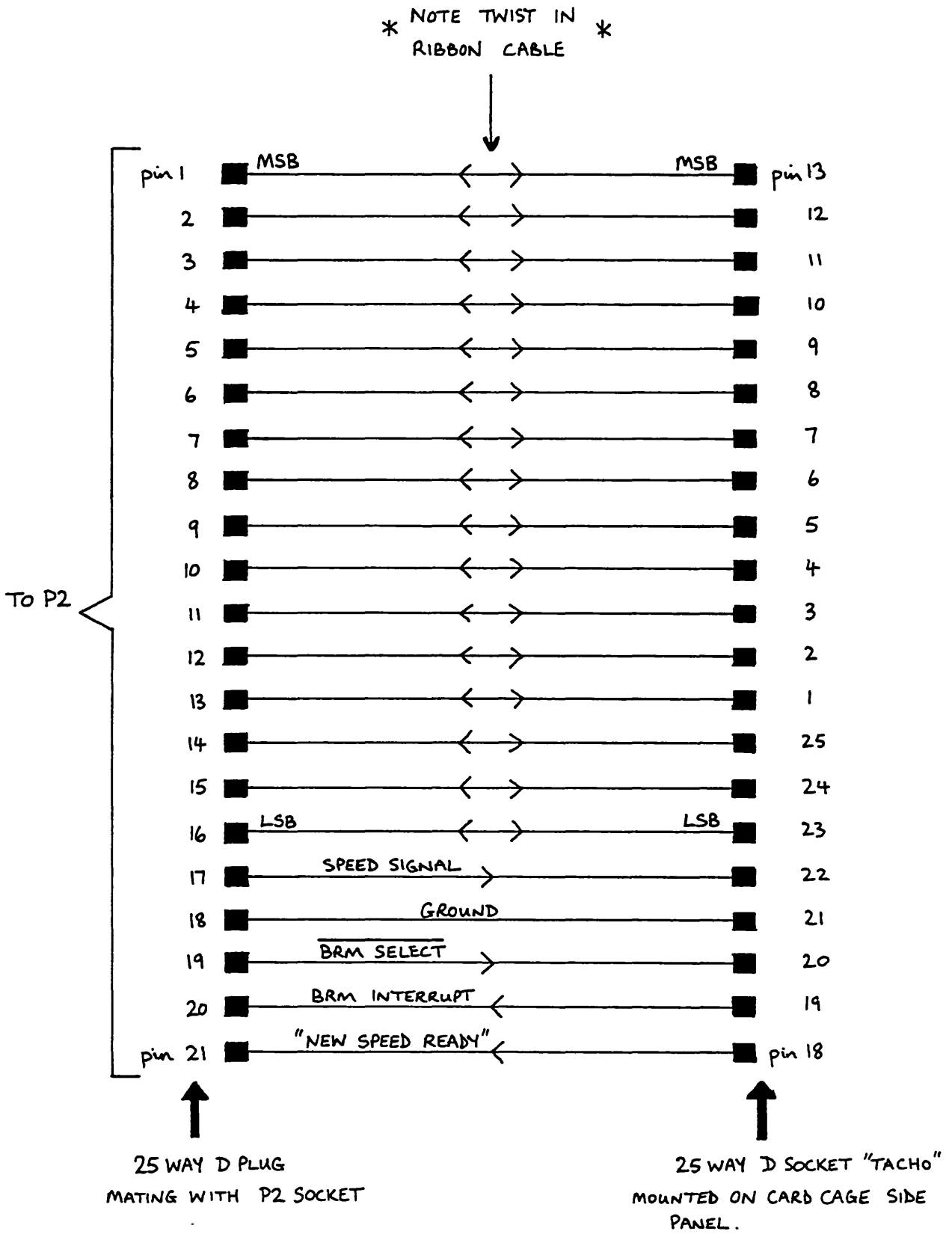
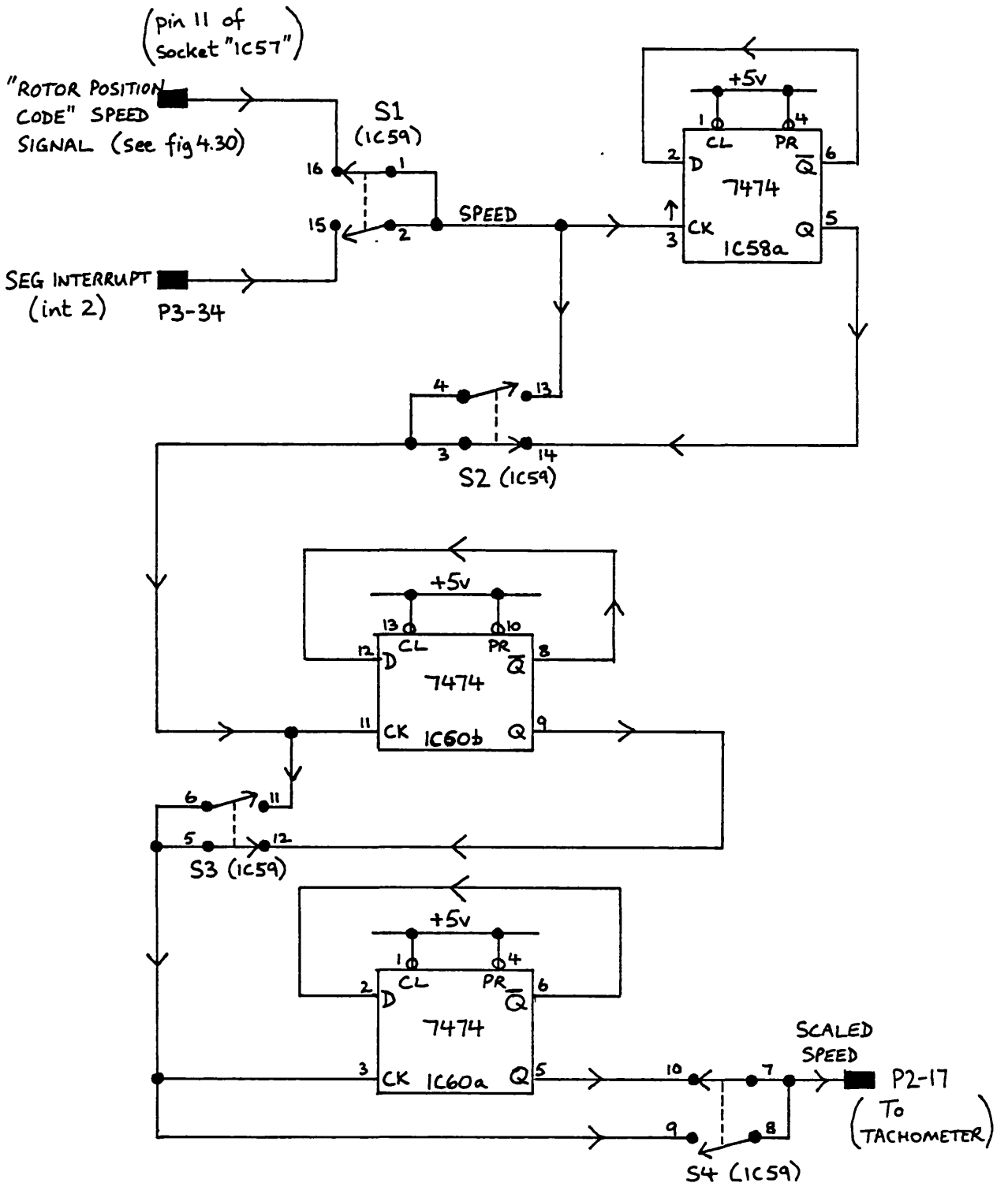


FIG.4.24. TACHOMETER "P2" TO CARD CAGE "TACHO" SOCKET CONNECTIONS.



IC59 TOP VIEW

SEG SPEED	S1	ROTOR CODE SPEED
NO DIVISION	S2	÷2
NO DIVISION	S3	÷2
NO DIVISION	S4	÷2

[NOTE IC58b IS UNUSED – ALL INPUTS TIED TO +5v]

FIG. 4.25 TACHOMETER SPEED-SIGNAL SCALING CIRCUITRY

in fig. 4.24.

It is appropriate in this sub-section to describe the division circuitry that was needed to scale the speed pulse signal, in order that the tachometer should produce a calibrated speed reading. The digital tachometer described in Chapter 8 generates a calibrated reading from a speed signal that occurs at a rate of either six or seven pulses per revolution. The circuit shown in fig. 4.25 was designed so that speed signals with two, four, or eight times the basic rate of pulses per revolution could be used (thus allowing a position sensor disc with up to 56 segments on it to be employed).

Switch S1 of the dual-in-line four switch unit (IC59) selects the speed signal either from the SEG interrupt line (int2) or from the "Rotor Position Code" input. Switches S2, S3, and S4 each enable a divide-by-two stage. The relevant "active" positions for the switches are indicated in fig. 4.25.

4.7.4.3 The "Keypad Data" Input Port

The "Keypad Data" input port circuitry is shown in fig. 4.26. The input data was routed in via edge-connector P4 of the prototyping board. The strobe signal for the two 74LS374 octal flip flops (IC41, IC42) is generated by a 74LS123 monostable (IC48(a)) which is triggered when a DATA interrupt (int7) is requested. The interrupt signal causes the $\overline{\text{INT9}}$ line to go to logic 0, and this transition initiates a 2.5μs strobe pulse from the monostable. The strobe pulse from the monostable is also sent to the keypad "Key Code" input port strobe circuit (the line going to IC51(a) pin 1), so that the identification code for the data is simultaneously strobed in.

The relevant connections between connector P4 and the DATA IN socket on the card cage side panel are detailed in

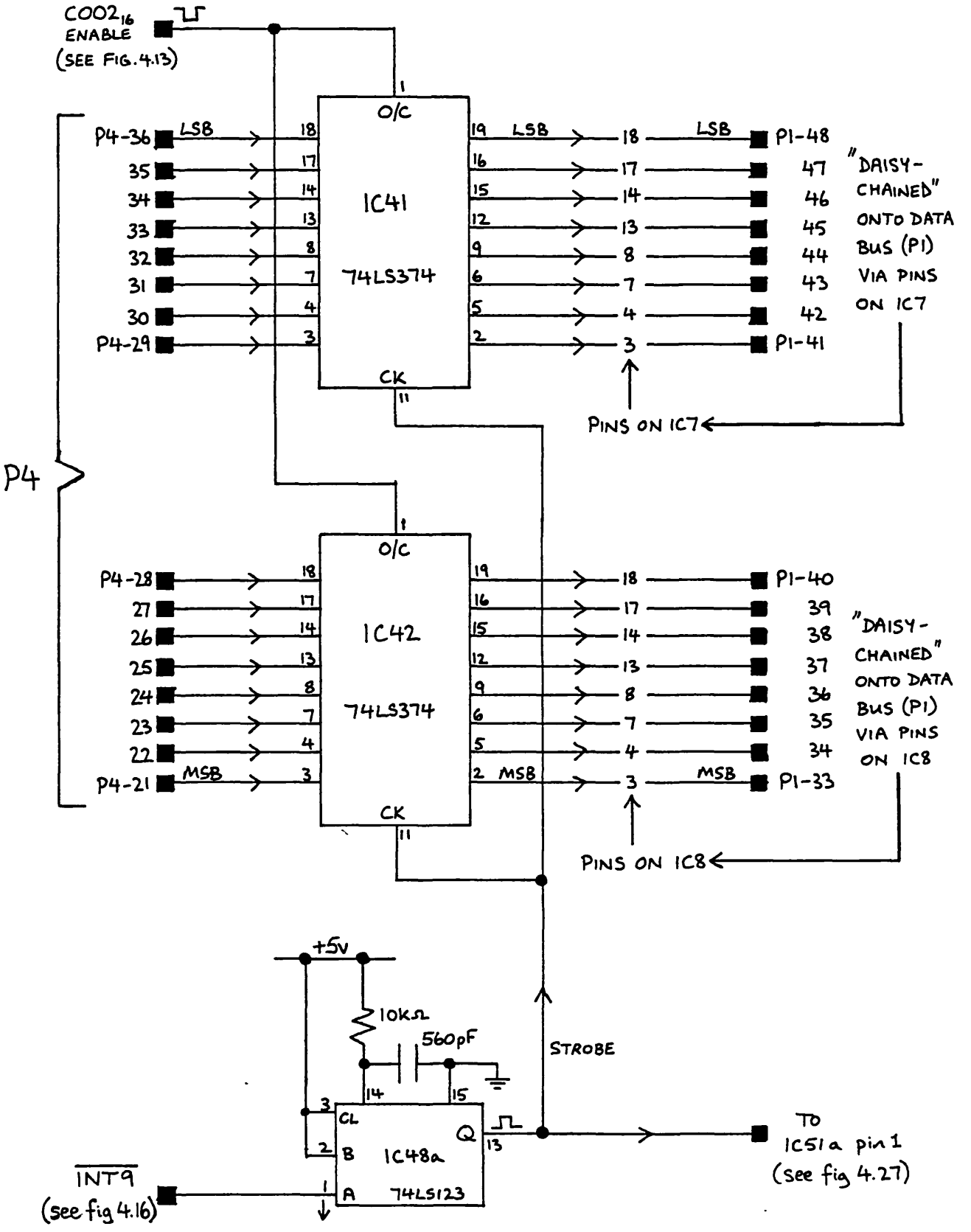


FIG. 4.26. "KEYPAD DATA" INPUT PORT

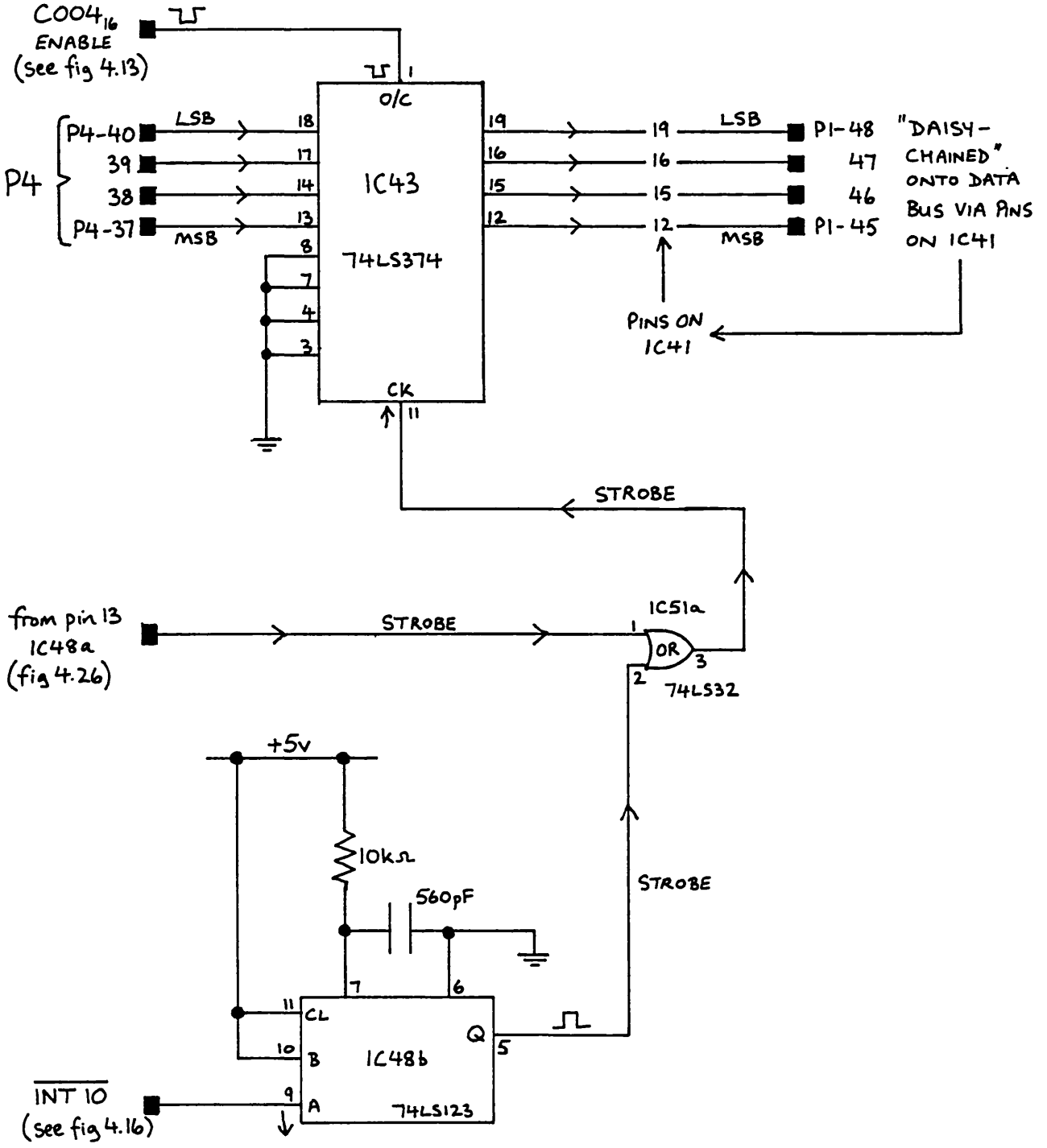


FIG. 4.27. KEYPAD "KEY CODE" DATA PORT

sub-section 4.7.4.6.

4.7.4.4 The Keypad "Key Code" Input Port

The keypad "Key Code" input port circuitry is shown in fig. 4.27. The code signals were routed in via edge-connector P4 of the prototyping board. The strobe signal for the 74LS374 octal flip flops (IC43) is generated by either: (a) the 74LS123 monostable (IC48(b)) which is triggered when a DISPLAY interrupt (int8) is requested. The interrupt signal causes the $\overline{\text{INT10}}$ line to go to logic 0 and this transition initiates a 2.5μs strobe pulse from the monostable; or (b) by the strobe circuitry associated with the keypad "Data" input port shown in fig. 4.26. This strobe signal occurs when a DATA interrupt is requested.

The 74LS32 OR gate (IC51(a)) routes the strobe signal from either source to the clock input of the 74LS374 octal flip flop.

The relevant connections between edge-connector P4 and the DATA IN socket on the card cage side panel are detailed in sub-section 4.7.4.6.

4.7.4.5 The "Forward/Reverse Command" Input Line

The "Forward/Reverse Command" line was routed onto the prototyping board via edge-connector P4 and was buffered as shown in fig. 4.28.

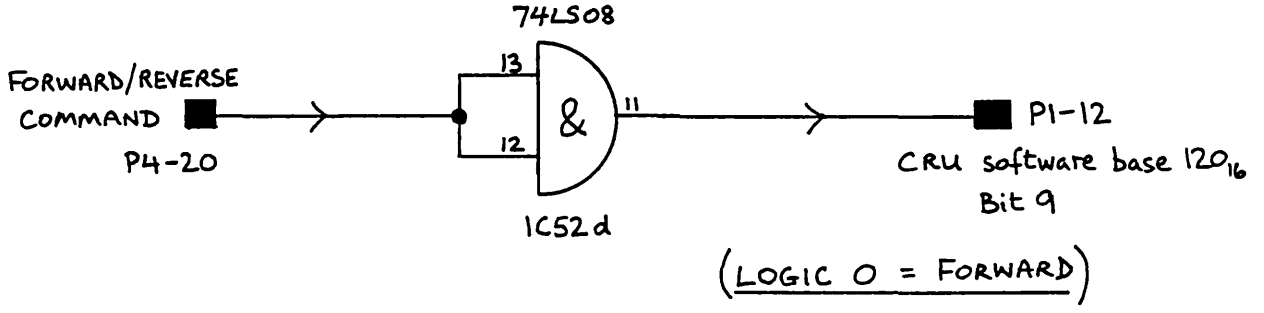


FIG. 4.28 "FORWARD/REVERSE COMMAND" LINE BUFFERING.

4.7.4.6 Connections between the "DATA IN" Card Cage Socket, and Prototyping Board Edge-Connectors P3 and P4

Fig. 4.29 shows the wiring connections between the keypad DATA IN socket on the card cage side panel and edge-connectors P4 and P3 on the prototyping board. These connections are relevant to the circuits shown in figs. 4.26, 4.27, and 4.28. It should be noted that the ground connection on DATA IN (pin 25) was routed from the prototype board via the DISPLAY socket (pin 1).

4.7.4.7 The "Rotor Position Code" Input Port

The "Rotor Position Code" input port is shown in fig. 4.30. The code signals were routed onto the prototyping board via a 14 way dual-in-line socket (IC57). The strobe signal for the 74LS374 octal flip flops (IC56) was provided by the Optical Position Sensor counter circuit (see subsection 4.7.6). The interconnections between the COUNTER D-type socket on the card cage side panel and the socket "IC57" were made by ribbon cable as detailed in fig. 4.31.

4.7.4.8 The "System Initialise Line"

The "System Initialise Line" is generated by CRU bit 8 (software base address number 120_{16}). The buffer shown in fig. 4.32 was required to prevent the line being overloaded by the various system peripherals connected to it via the output ports.

4.7.4.9 The "Inverter Control" Output Port

The "Inverter Control" output port is shown in fig. 4.33. The full 16 bit data bus was routed out via the port to edge-connector P3, even though only 14 bits of the data bus were required to control the 7 phase MOSFET inverter. The relevant connections between P3 and the "INV CONT" D-type socket on the card cage side panel are shown in fig. 4.34.

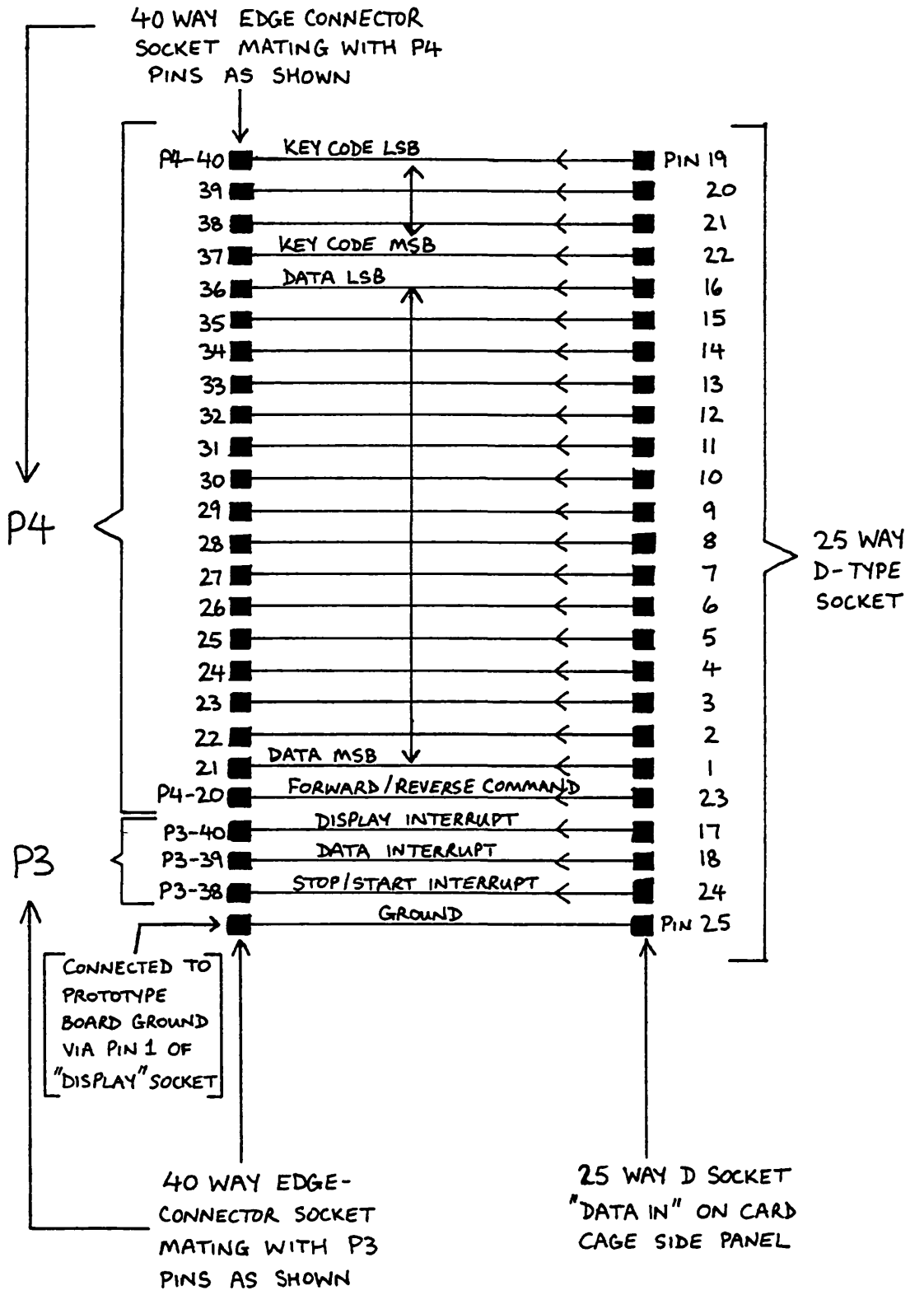


FIG.4.29. "DATA IN" SOCKET TO PROTOTYPING BOARD CONNECTIONS.

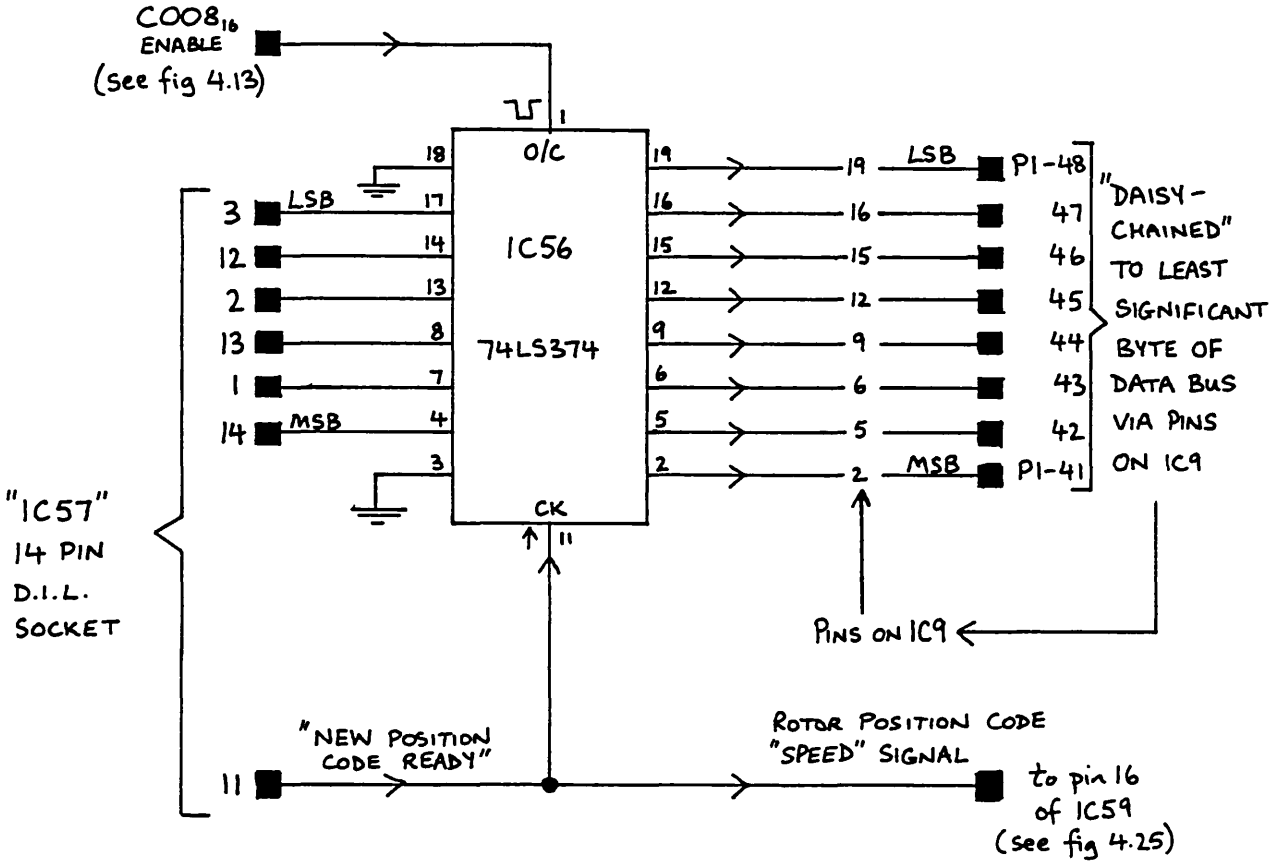


FIG. 4.30. "ROTOR POSITION CODE" INPUT PORT

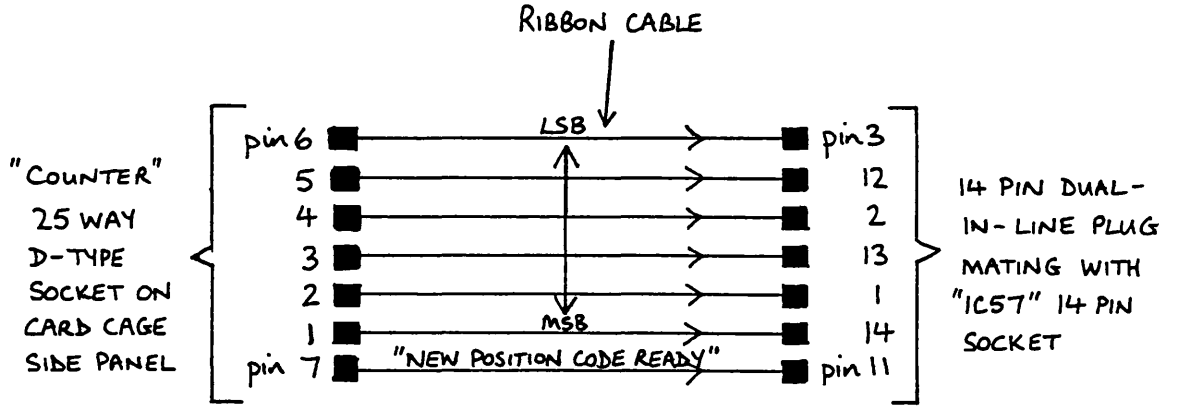


FIG. 4.31. INTERCONNECTIONS BETWEEN "COUNTER" SOCKET ON CARD CAGE SIDE PANEL AND THE 14 PIN PLUG MATING WITH SOCKET "IC57".

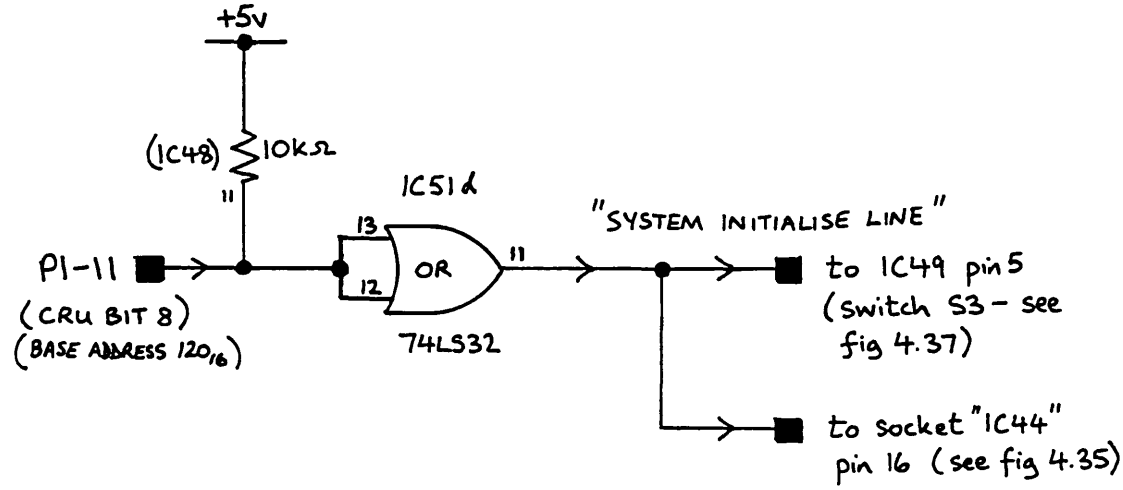


FIG. 4.32 "SYSTEM INITIALISE LINE"

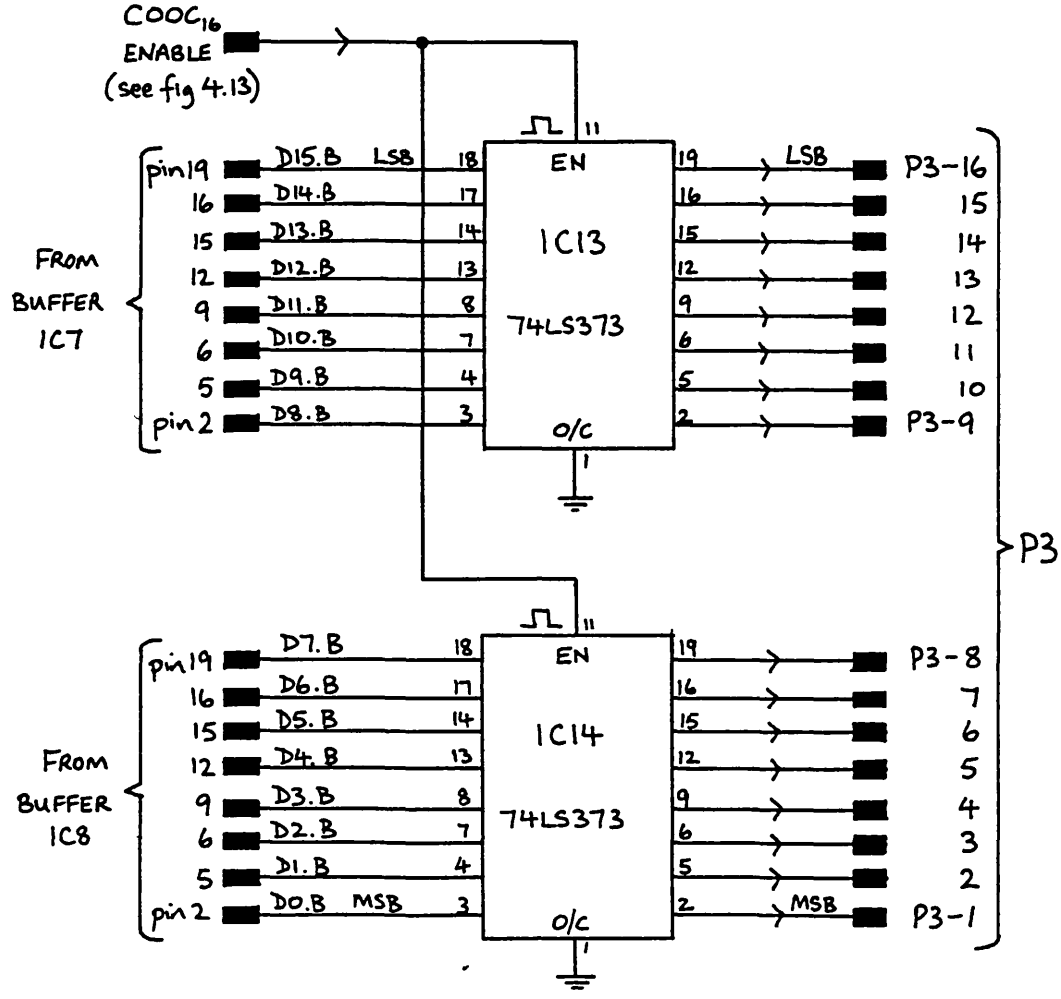


FIG. 4.33 "INVERTER CONTROL" OUTPUT PORT

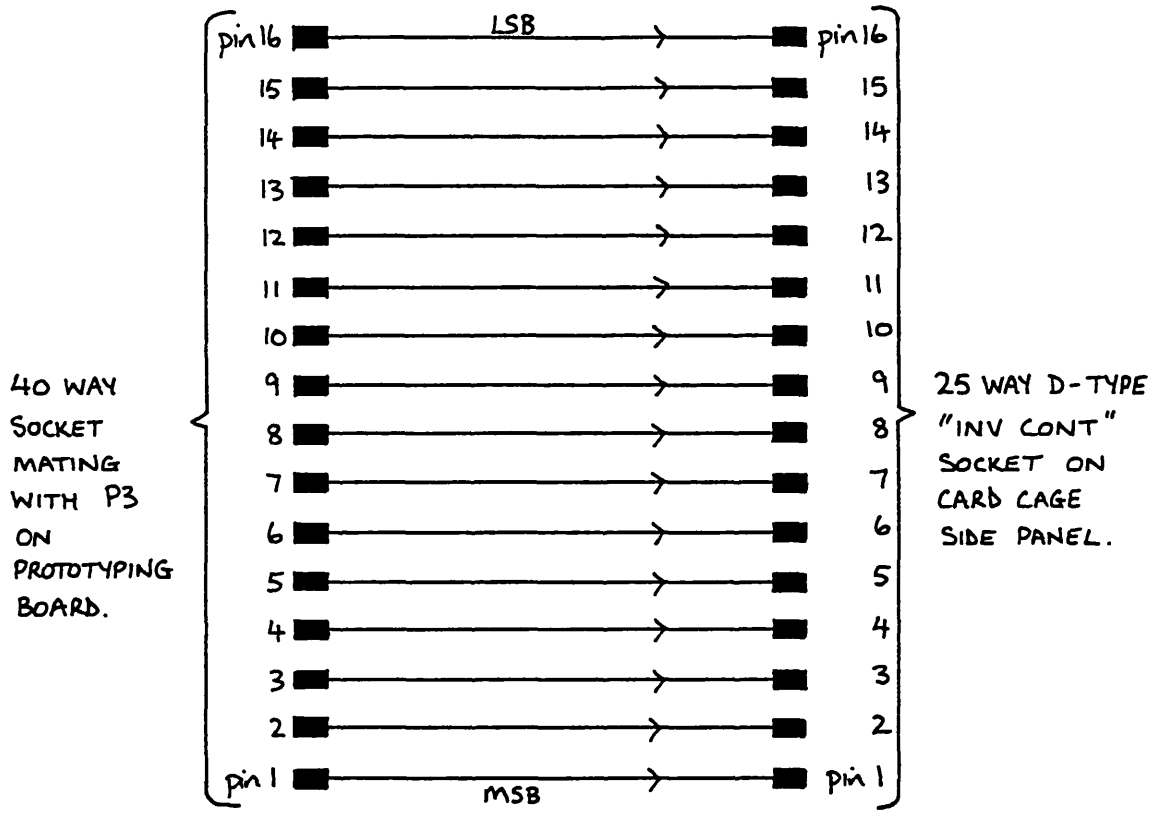


FIG. 4.34. INTERCONNECTIONS BETWEEN P3 AND CARD CAGE SIDE PANEL "INVERTER CONTROL" SOCKET.

4.7.4.10 The ±60 volt Power Supply "Combined Control and Enable" Port

The "Combined Control and Enable" port for the ±60 volt power supply is shown in fig. 4.35. The signals were routed off the prototyping board via a 16 way dual-in-line socket (IC44). The most significant bit of the output port was controlled by the "System Initialise Line". The connections between socket "IC44" and the "VOLTS" D-type socket on the card cage side panel are shown in fig. 4.36. The ground connection (pin 17) on the VOLTS socket was obtained via pin 1 of the DISPLAY socket.

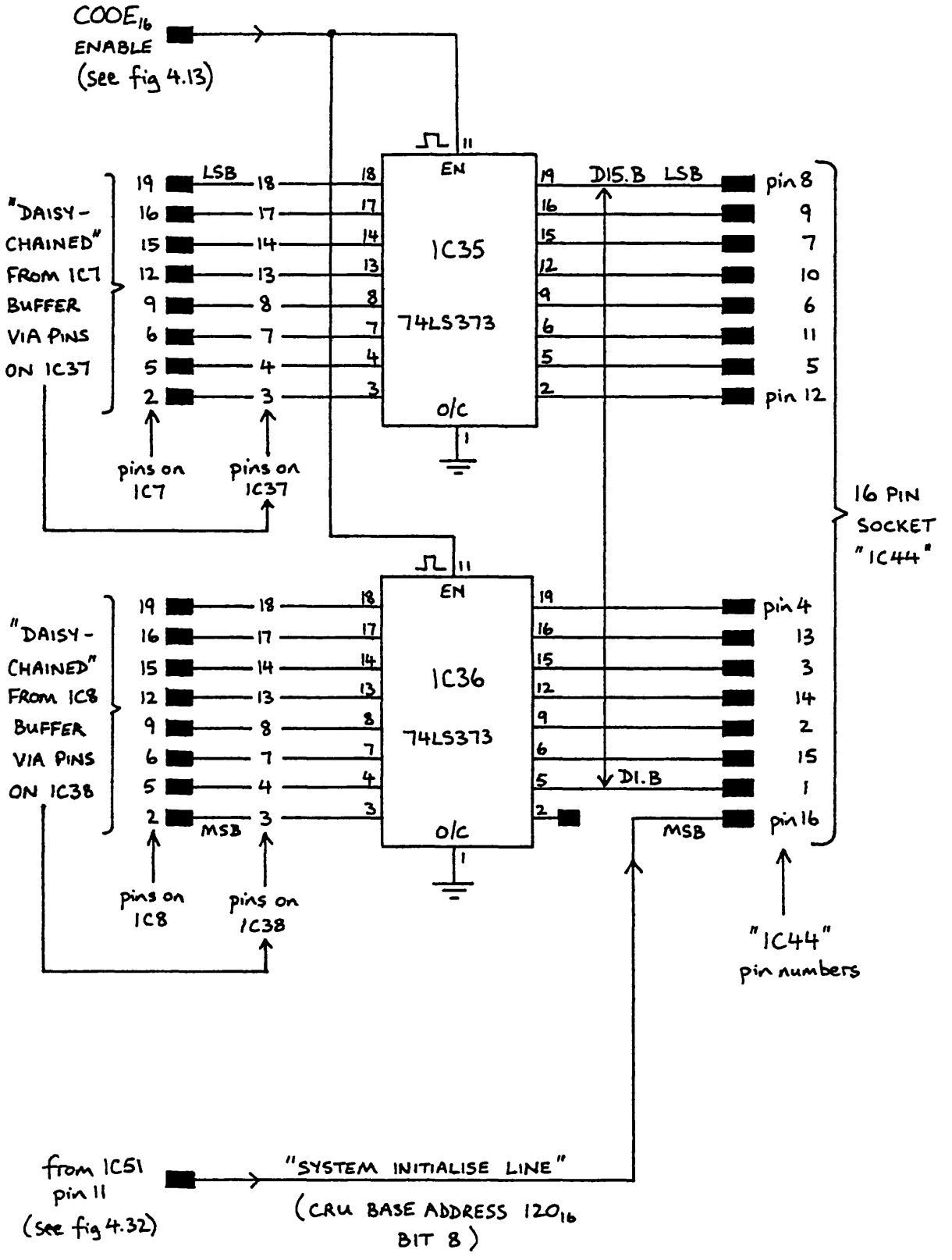


FIG. 4.35. "COMBINED CONTROL AND ENABLE" PORT FOR ±60 VOLT PSU.

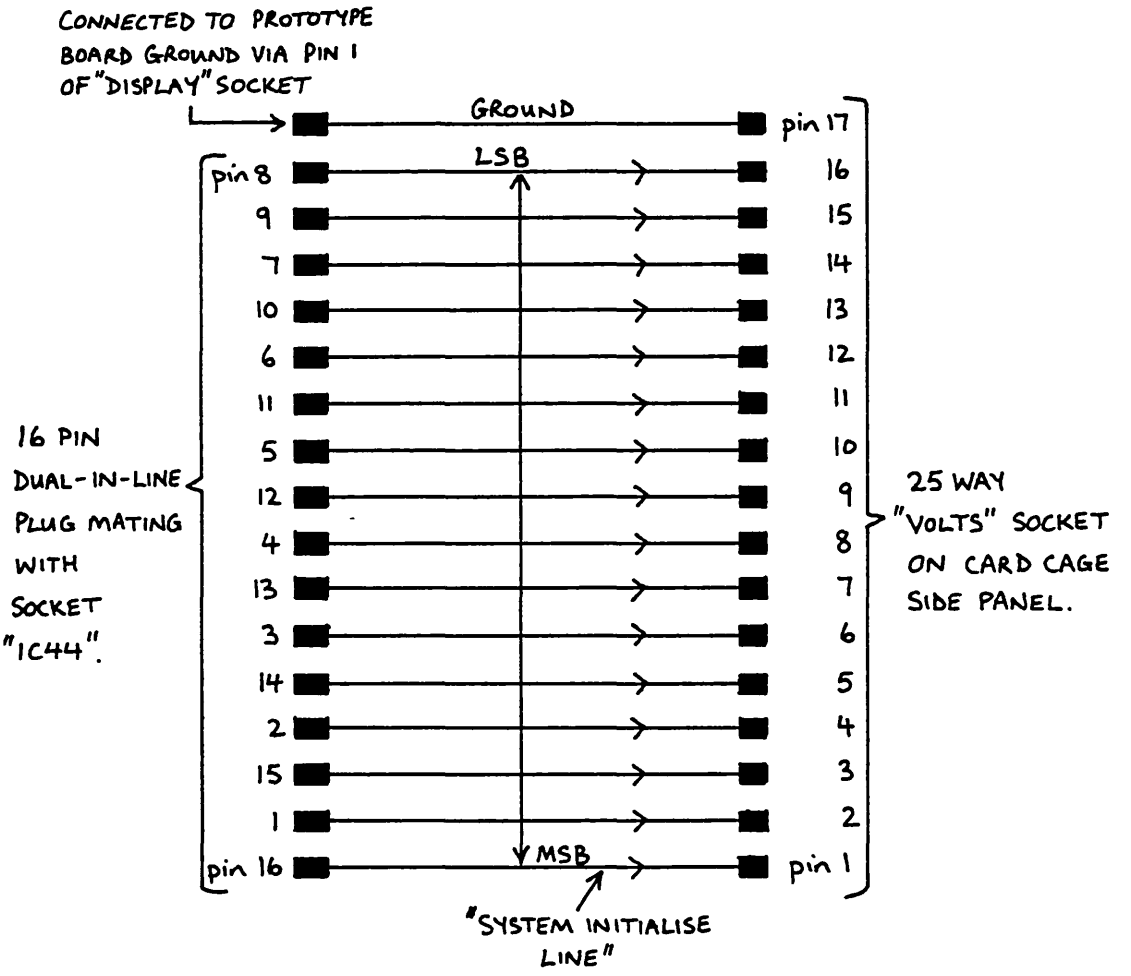


FIG.4.36. CONNECTION BETWEEN SOCKET "IC44" AND THE "VOLTS" SOCKET ON THE CARD CAGE SIDE PANEL.

4.7.4.11 The "Inverter Enable" Output Port

The "Inverter Enable" output port is shown in fig. 4.37. The signals were routed off the prototyping board via edge-connector P3. The most significant digit of the port could either be connected to D0.B of the data bus or to the "System Initialise Line" by use of the dual-in-line switch S3. This allowed the port to be used for general purpose 16 bit tasks if it was not required for its primary job of enabling the 7 phase MOSFET inverter.

The relevant connections between P3 and the "INV SEL" socket on the card cage side panel are shown in fig. 4.38. The ground connection for the INV SEL socket (pin 17) was obtained via pin 17 of the "VOLTS" socket.

4.7.4.12 The "Data Display" Output Port

The "Data Display" output port is shown in fig. 4.39. The "key strobe" and "data strobe" signals are used to load the data that the port presents on P4 into the "Data Display" circuitry housed in the keypad auxiliary logic unit. The strobes are derived from the "$C012_{16}$ Enable" signal that loads new data into the output port from the data bus. Two monostables (IC47(a) and (b)) delay the strobe signals suitably to allow for the propagation time of new data through the octal latches (IC39 and IC40). A flip flop (IC53(a)) enables "key strobe" and "data strobe" alternately on successive data transfers (thus allowing 16 bit data and a 4 bit key code to be transmitted to the data display unit over the same 16 bit bus). The spare memory mapped input line ($C006_{16}$ Enable) was used as the clear line for the flip flop. Having operated the line, the microprocessor "knows" the appropriate order in which it should present the data and key code to the output port for correct transfer to the Data Display unit. A CRU bit could have been used for the reset function but the "$C006_{16}$ Enable" line was spare and it offered a slightly faster

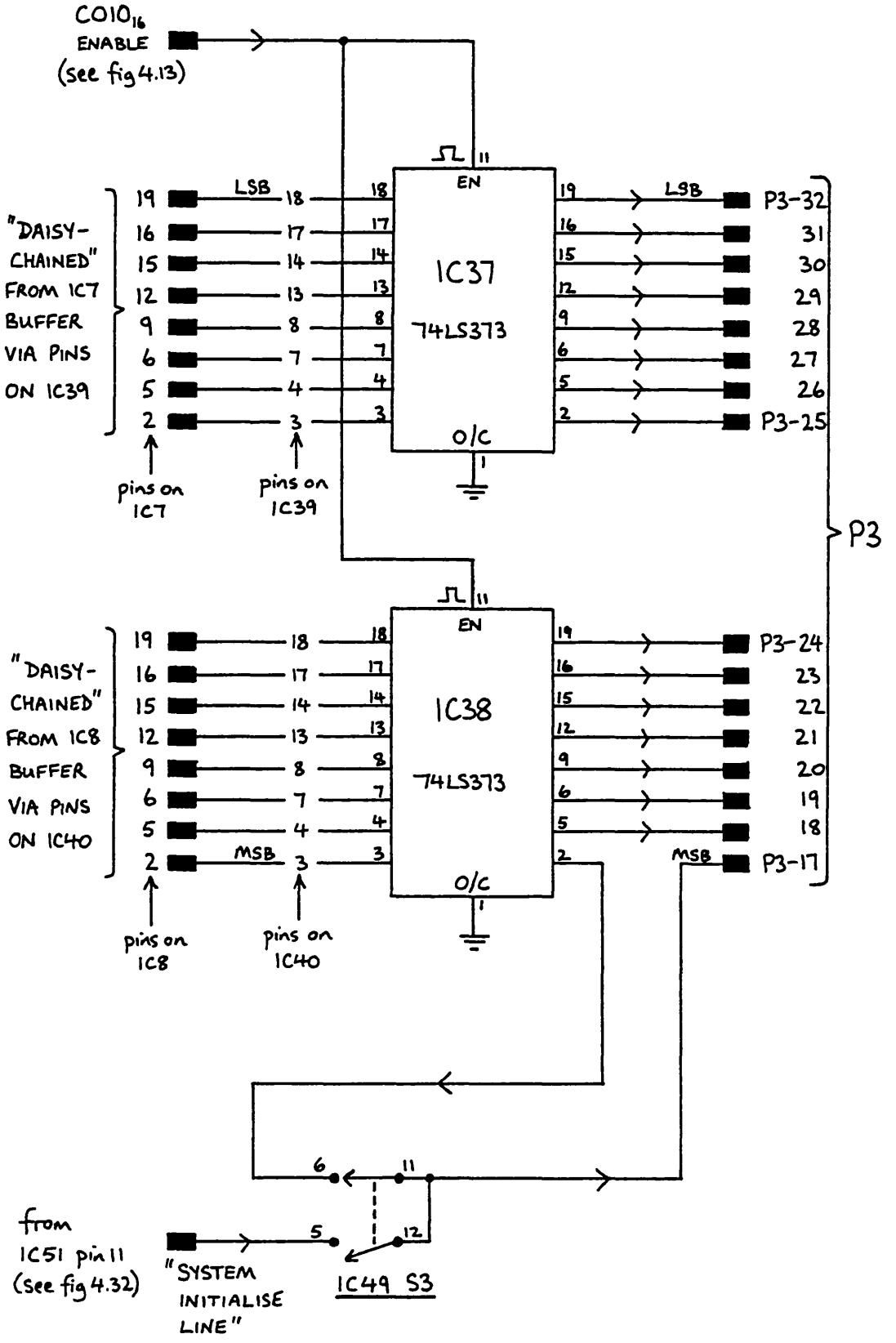


FIG. 4.37. "INVERTER ENABLE" PORT

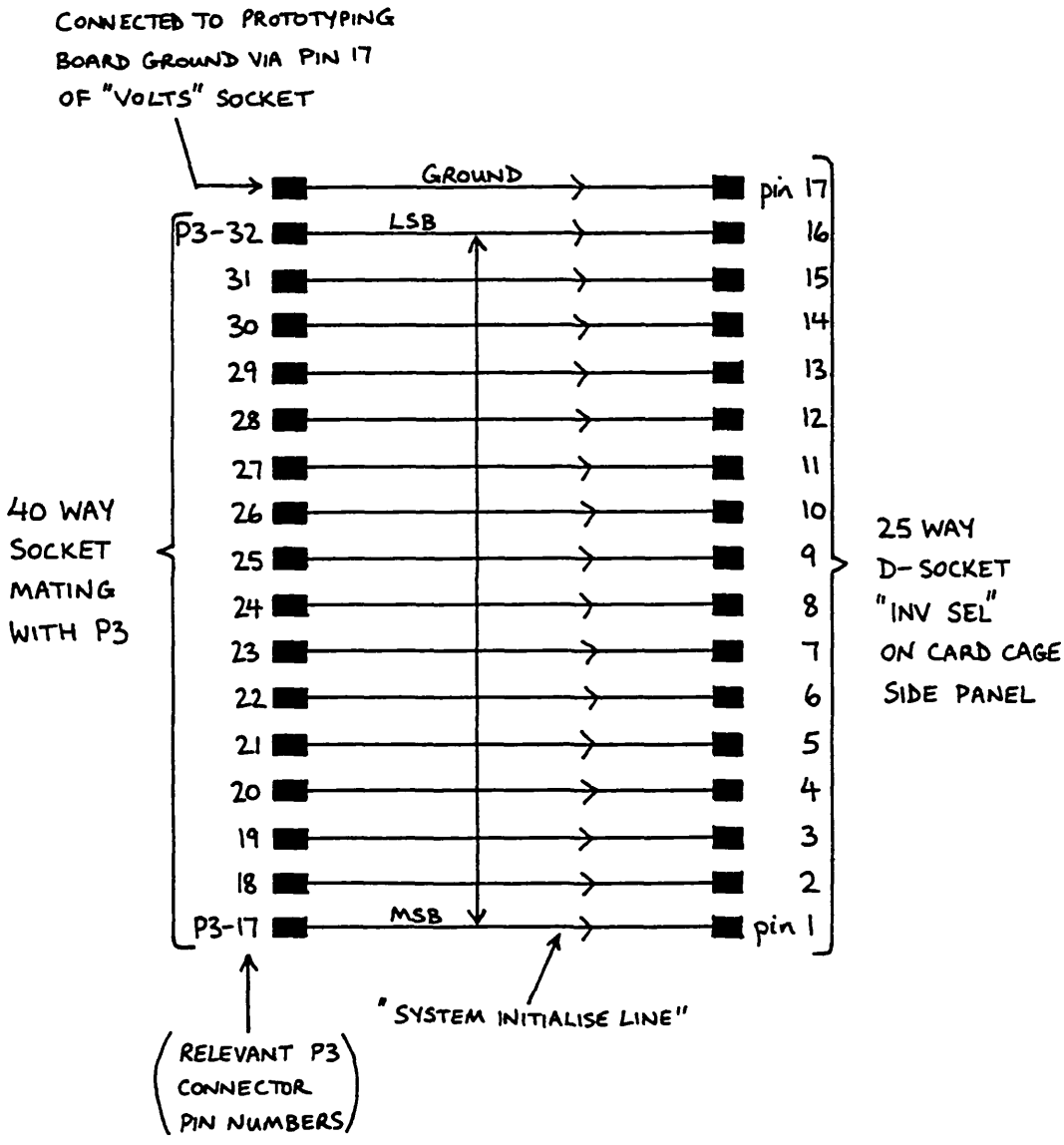


FIG. 4.38. THE INTERCONNECTIONS BETWEEN THE SOCKET MATING WITH P3 AND THE "INV SEL" SOCKET.

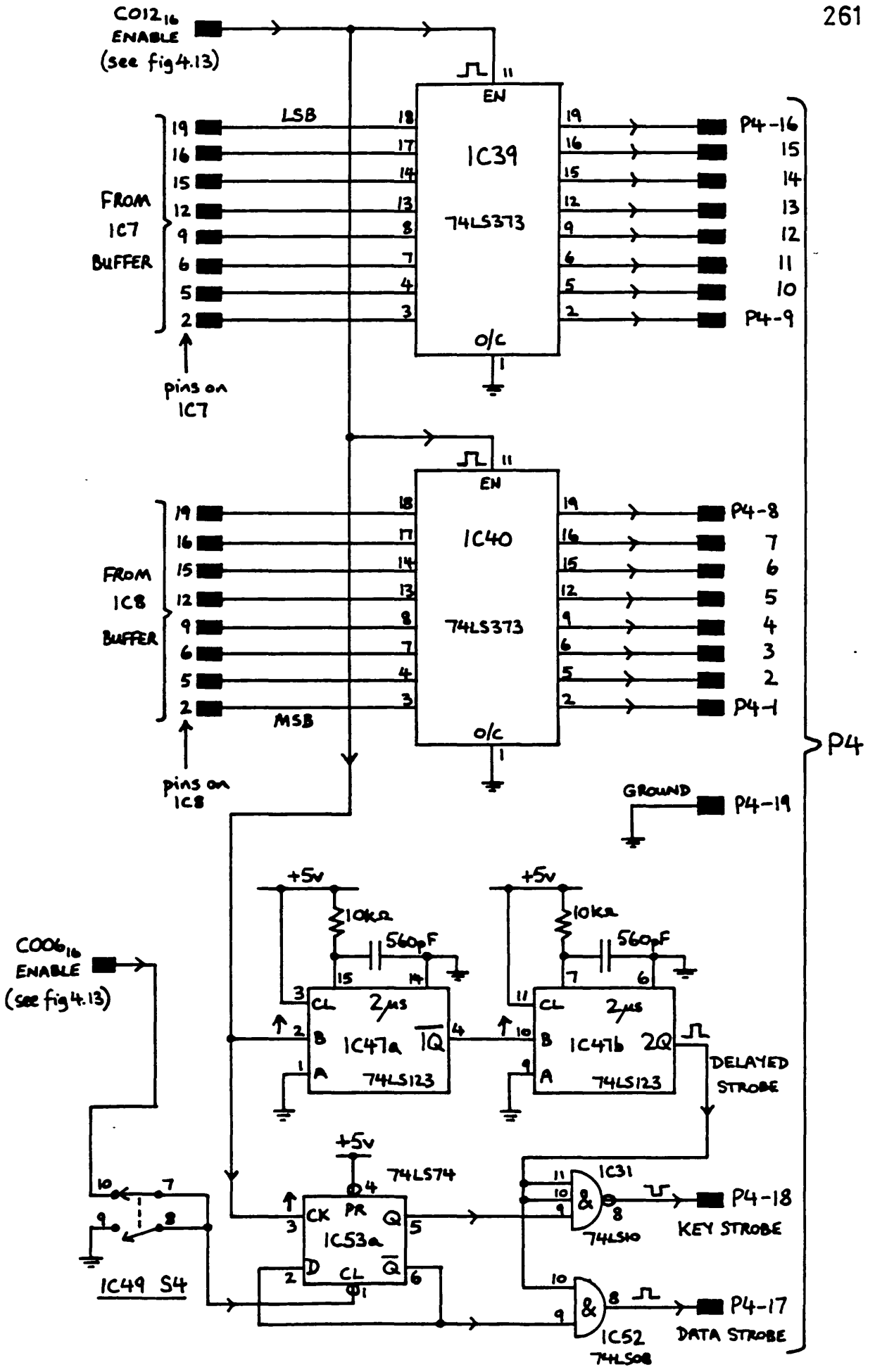


FIG. 4.39. "DATA DISPLAY" OUTPUT PORT

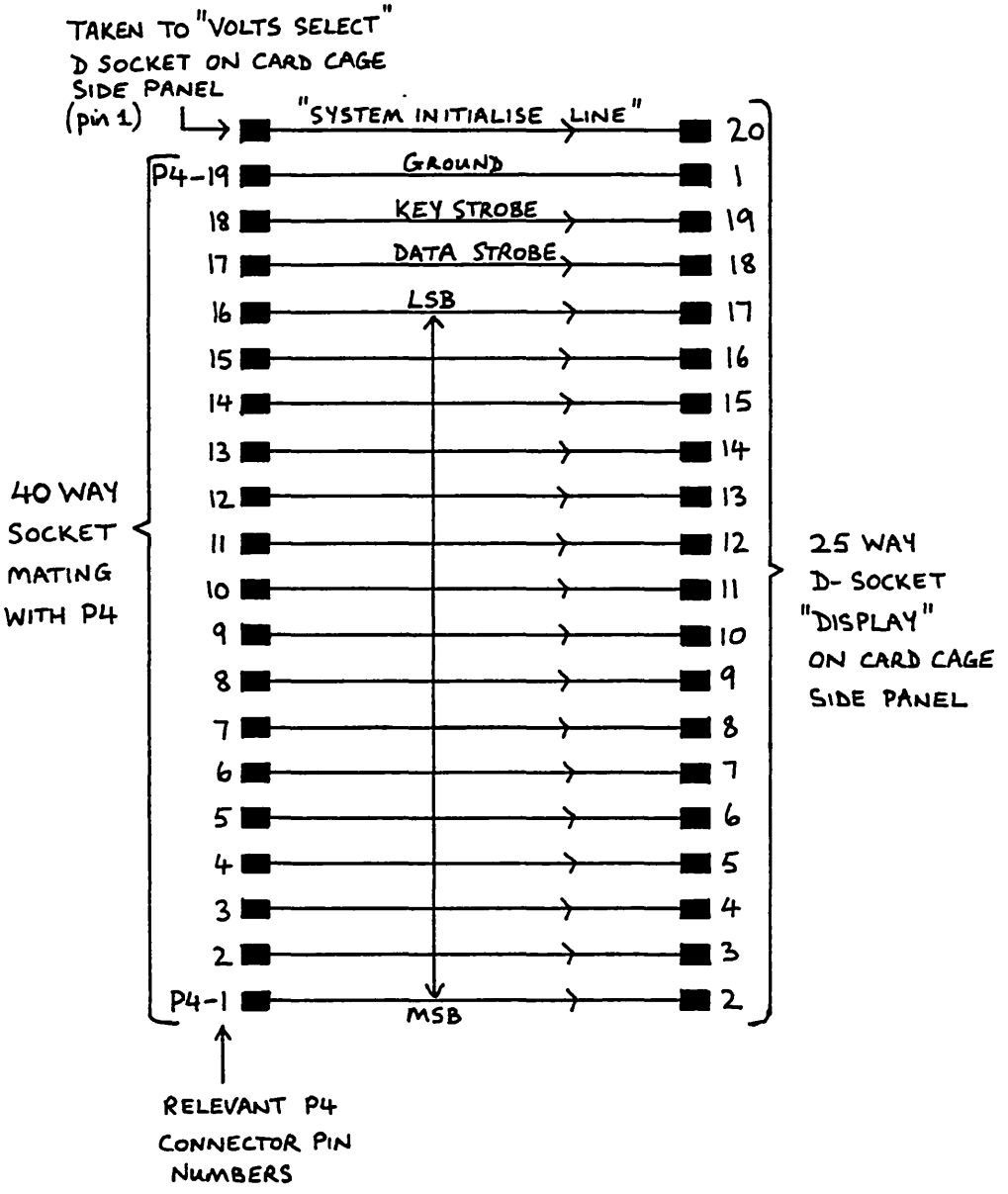


FIG.4.40 THE CONNECTIONS BETWEEN THE SOCKET MATING WITH P4 AND THE CARD CAGE SIDE PANEL MOUNTED "DISPLAY" SOCKET.

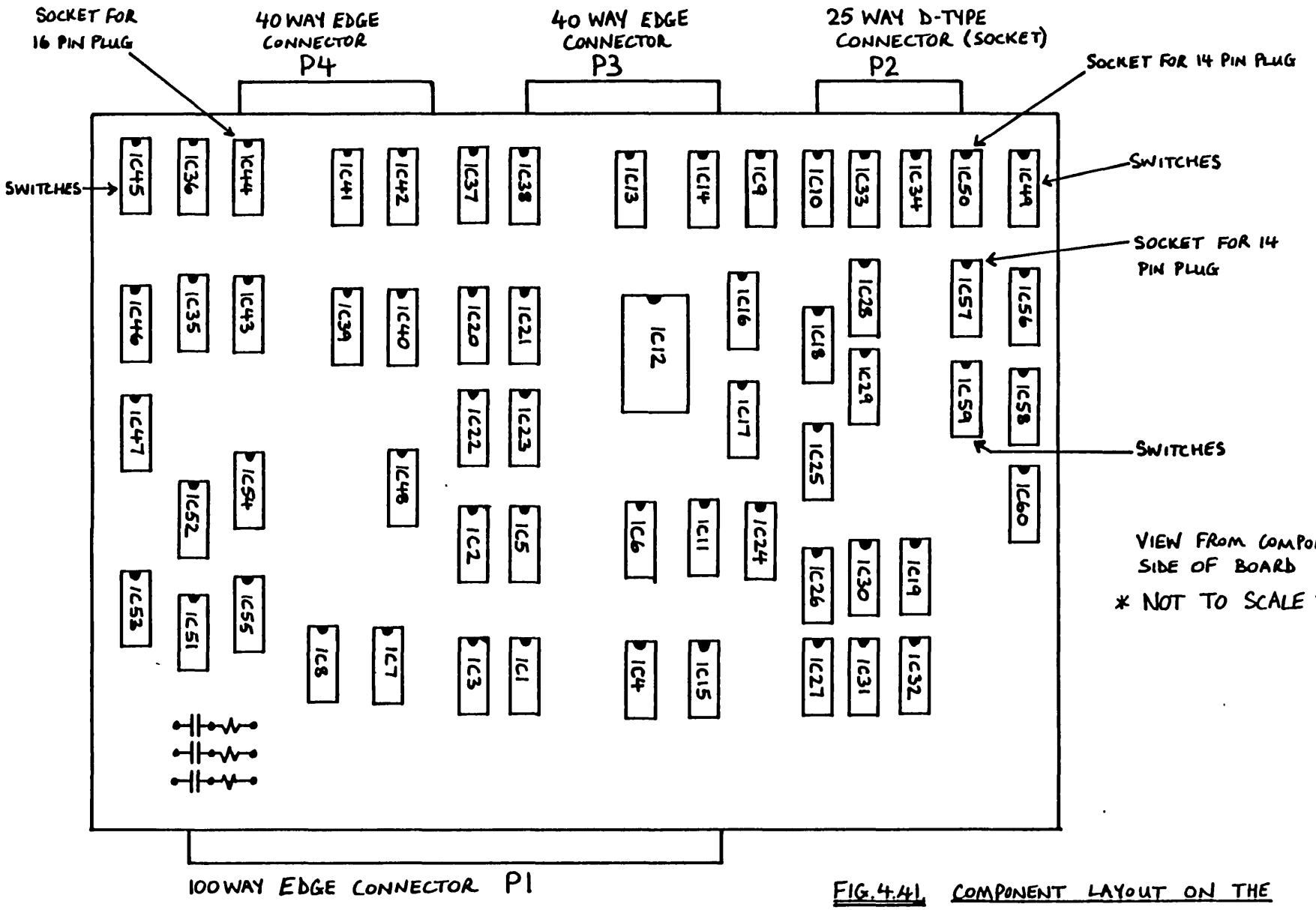


FIG. 4.41. COMPONENT LAYOUT ON THE PROTOTYPING BOARD.

execution time; hence it was used. The switch S4 of IC49 was included to allow the flip flop to be held in a cleared state if desired, thus causing only Data strobes to be generated.

The data and strobe signals were routed off the prototyping board via connector P4. The relevant connections between P4 and the "DISPLAY" D-type socket on the card cage side panel are shown in fig. 4.40. The "System Initialise Line" for the DISPLAY socket (pin 20) was routed from the "VOLTS" socket (pin 1) on the card cage side panel.

4.7.5 Miscellaneous Details about the Prototyping Board

The layout of the various integrated circuits and connectors on the prototyping board are as shown in fig. 4.41. The decoupling on the +5 volt power supply rails around the board comprised five 47μF tantalum capacitors, one 33μF electrolytic capacitor and nine 0.01μF ceramic capacitors. Virtually all the connections on the prototyping board were made by the wirewrap technique. The settings of the dual-in-line four-switch unit (IC49) are as shown in fig. 4.42.

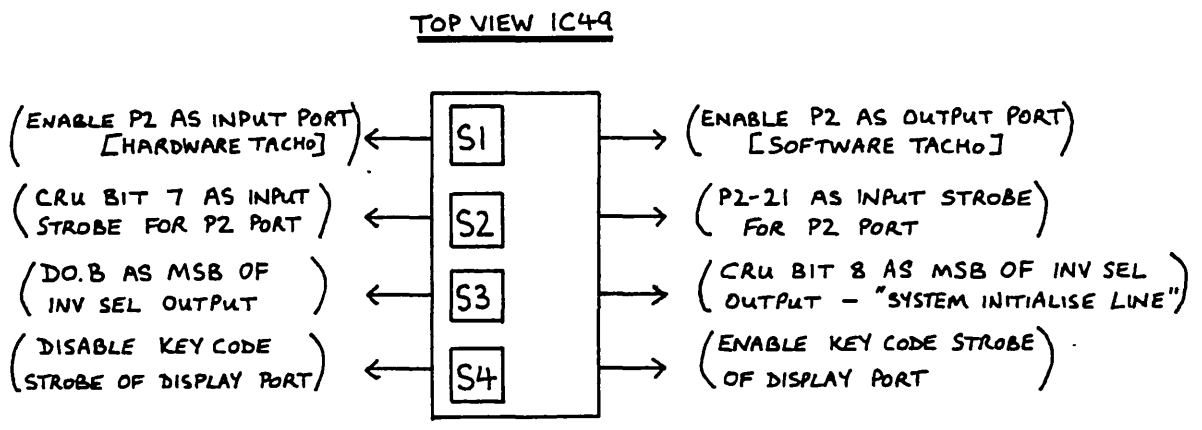


FIG. 4.42. THE SETTINGS OF SWITCHES S1 TO S4 OF "IC49"

For normal motor system operation (i.e. hardware tachometer) the switches are set with S1 to the left and S2, S3, and S4 to the right.

4.7.6 The Optical Position Detector Interface Unit

It was intended to use the microprocessor initially to control the magslip motor system described in Chapter 3, and then to control the 7 phase "square wave" motor and inverter described in Chapters 6 and 5 respectively. To make this possible a position detector interface unit was required to enable both motor systems to communicate with the microprocessor. In addition, it was hoped to control at least one of the motor systems by both an interrupt based control program and a sampled position-code based program. A choice was available in the hardware required to implement the sampled position code system: it was possible to use either a Gray code position sensor disc or an incremental position sensor disc feeding a cyclic binary counter (as briefly outlined in section 4.2). It was decided to use the latter approach, since it allowed the use of the same position sensor disc for both the interrupt based and position-code based programs.

The TTL based circuit shown in fig. 4.43 was designed to interface either the magslip position detector or the 7 phase motor position detector to the microprocessor. The magslip position detector produces +5 volt TTL level SYNC' and SEG' signals, whereas the 7 phase motor position detector produces +12 volt CMOS level $\overline{\text{SYNC}}$' and $\overline{\text{SEG}}$' signals. Therefore, the first stage of the circuit inverts and level shifts the CMOS $\overline{\text{SYNC}}$' and $\overline{\text{SEG}}$' signals down to TTL levels. This is achieved by a CD4049 CMOS Hex Inverting Buffer/ Converter (IC12) which is supplied from the +5 volt power supply. The CMOS $\overline{\text{SEG}}$' signal is inverted and level shifted by IC12(a), and the CMOS $\overline{\text{SYNC}}$' signal is similarly processed by IC12(b).

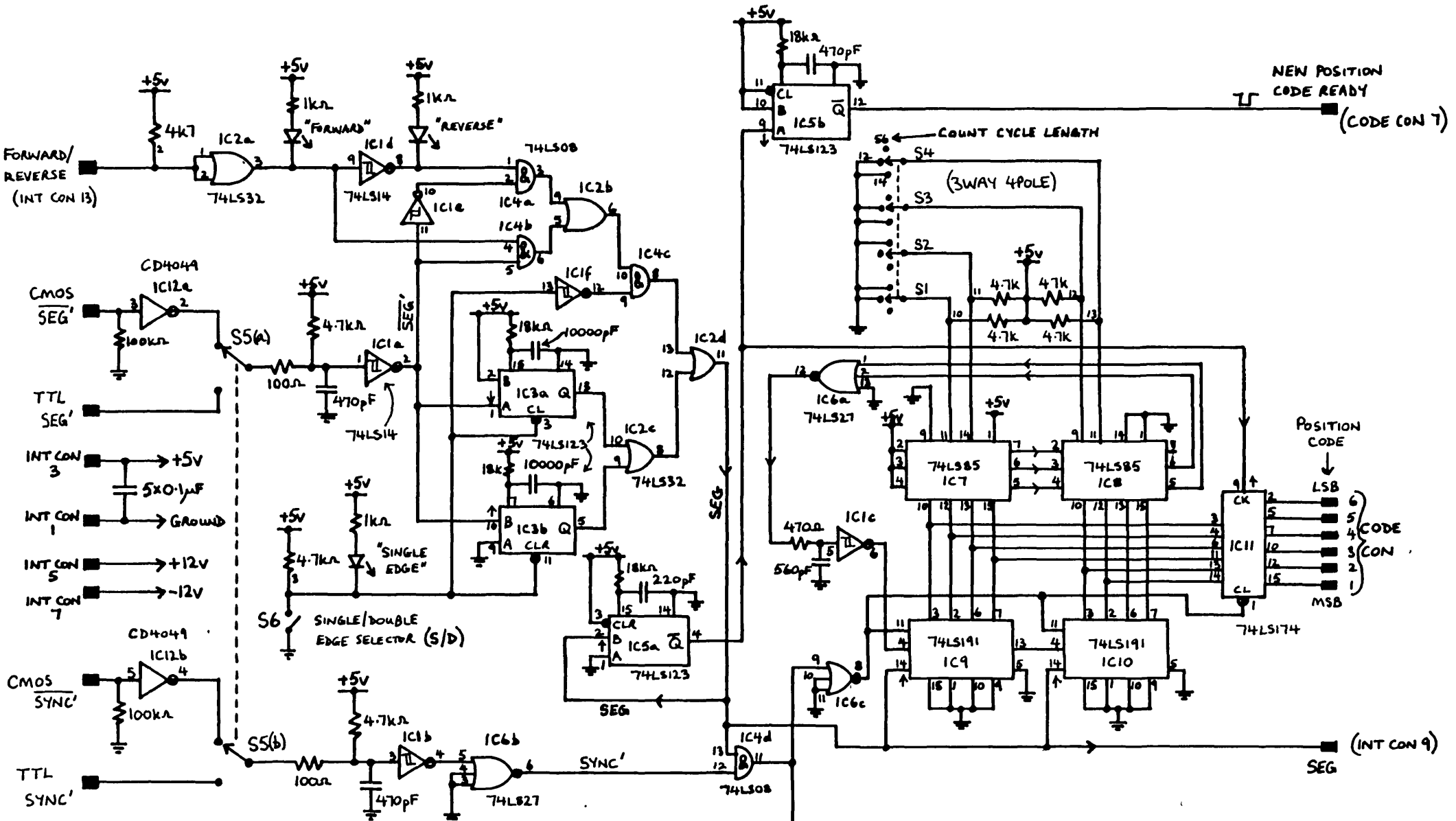


FIG 4.43. OPTICAL POSITION DETECTOR INTERFACE CIRCUIT

A double pole toggle switch (S5(a) and (b)) selects either the TTL or level shifted CMOS signals, and they are then debounced by 100Ω/470pF RC networks in combination with 74LS14 inverting schmitt triggers (IC1(a) for SEG' signal, IC1(b) for SYNC' signal). The 47ns time constant of the RC networks is sufficient to remove spurious glitches.

The debounced $\overline{\text{SEG}}$' signal from IC1(a) then passes through either the "Single-Edge" or "Double-Edge" processing logic depending on the form of position detector pattern used (discussed in Chapter 3). The "Single-Edge" logic (IC1(e), IC4(a), IC4(b), IC2(b), IC1(f), and IC4(c)) is enabled when the S/D switch (S6) is closed. The logic is controlled by the Forward/Reverse Command line: it inverts the $\overline{\text{SEG}}$' input signal in the forward direction of rotation, but allows it to pass straight through in the reverse direction. Hence the SEG output signal from IC4(c) has the correct "polarity" in either direction of rotation (as discussed in Chapter 3). If the S/D switch is open, the "Double-Edge" logic is enabled. The 74LS123 monostables (IC3(a) and IC3(b)) in combination with an OR gate (IC2(c)) produce a SEG pulse on each edge of the input $\overline{\text{SEG}}$' signal (also explained in Chapter 3). Only the "Single-Edge" or the "Double-Edge" circuit is enabled at any time, and IC2(d) takes the SEG signal from whichever circuit is active.

The SEG signal from IC2(d) is combined with SYNC' by an AND gate (IC4(d)) to produce the SYNC signal. The SEG and SYNC signals form the interrupt information for the interrupt based motor control programs; they are available from the Optical Position Detector Interface Unit via "INT CON" plug pins 9 and 11 respectively.

SEG and SYNC are also used to control the position-code counter circuitry. Two 74LS191 TTL synchronous up/down counters (IC9 and IC10) are connected to form a binary up counter with a maximum count of 2^8-1 (i.e. 255). However,

the maximum count value that the counter can reach is determined by the 74LS85 magnitude comparators (IC7 and IC8) which compare the binary count value with a number set by switches S1, S2, S3, and S4. The magnitude comparators control the count enable input of the counters. The counter is "permitted" to count successive SEG pulses so long as the count value is less than the number set by the switches S1 to S4. The counter is disabled when its count reaches the value set by the switches; it is also disabled if its count exceeds the value set by the switches, such as can occur at initial switch on. The counter is reset to zero by the SYNC pulse. Hence, if the correct number of SEG pulses occur in a revolution, the count value just reaches its maximum permitted value as a SYNC pulse occurs to reset the counter.

Switches S1 to S4 were actually implemented by a three-way four-pole switch and so three switch-selectable numbers are possible. The numbers 12, 14, or 56 can be preset into the magnitude comparators, and so count cycles of 0 to 11, 0 to 13, and 0 to 55 can be selected. The 0 to 11 count cycle was intended for use with the magslip, and the other two cycles were intended for possible use with the "square wave" motor (the 0 to 55 cycle being intended for use with the 56 SEG pulse disc shown in Chapter 6).

The glitch filter, formed by IC6(a) feeding a 74LS14 schmitt (IC1(c)) via a 470Ω/560pF RC network, was necessary to prevent spurious pulses from the magnitude comparators reaching the enable input of IC9. This input must only change level when the SEG signal to the clock inputs of the counters is at logic 1; failure to observe this precaution can result in spurious count values. With the circuit arranged as shown, the counters operated with no problems.

Each successive new count value is latched into the 74LS174 output buffer (IC11) by a pulse from the $\overline{Q}$ output

of monostable IC5(a). The monostable is triggered at the same time as the counter by the SEG pulse, and its period of 1.8μs ensures that the count value latched by IC11 is the fully stable new value. The "New Position Code Ready" strobe signal (required to transfer the position code from IC11 into the "Rotor Position Code" input port on the prototyping board) is generated by monostable IC5(b) and it has a period of 3.8μs.

The combined periods of monostables IC5(a) and (b) do mean that there is a 5.6μs delay before a new count value reaches the prototyping board but it was not felt that this delay was significant; it certainly could be reduced if necessary.

The Optical Position Detector Interface Unit was housed in a 190 x 110 x 60mm A.B.S. box as shown by the photograph in fig. 4.44. Connections between the unit and the microprocessor card cage side panel sockets were made via two 25 way D-type connectors: the "CODE CON" socket carried the signals that were routed via the "COUNTER" socket on the card cage side panel, and the "INT CON" plug carried the signals that were routed via the "OPTO" socket. Connections between the interface unit and the position sensor circuits of both the magslip and the "square wave" motor were made via banana sockets, and the "square wave" motor position sensor (described in Chapter 6) obtained its ±12 volt power supply via the interface unit.

4.7.7 Miscellaneous Details Relating to the Card Cage

4.7.7.1 Connections between the Card Cage Side Panel Sockets and the System Peripherals

A photograph of the card cage side panel is shown in fig. 4.45. Multicore cables were used to connect the card cage (C.C.) side panel sockets to the various system peripherals as follows:-

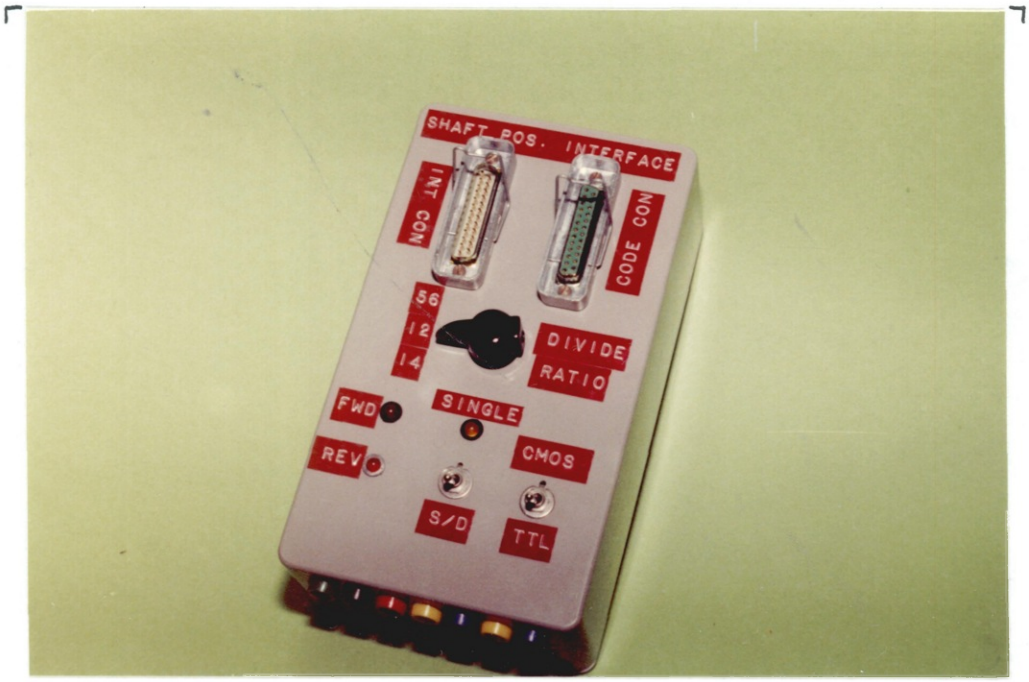


FIG. 4.44. THE OPTICAL POSITION DETECTOR INTERFACE UNIT.

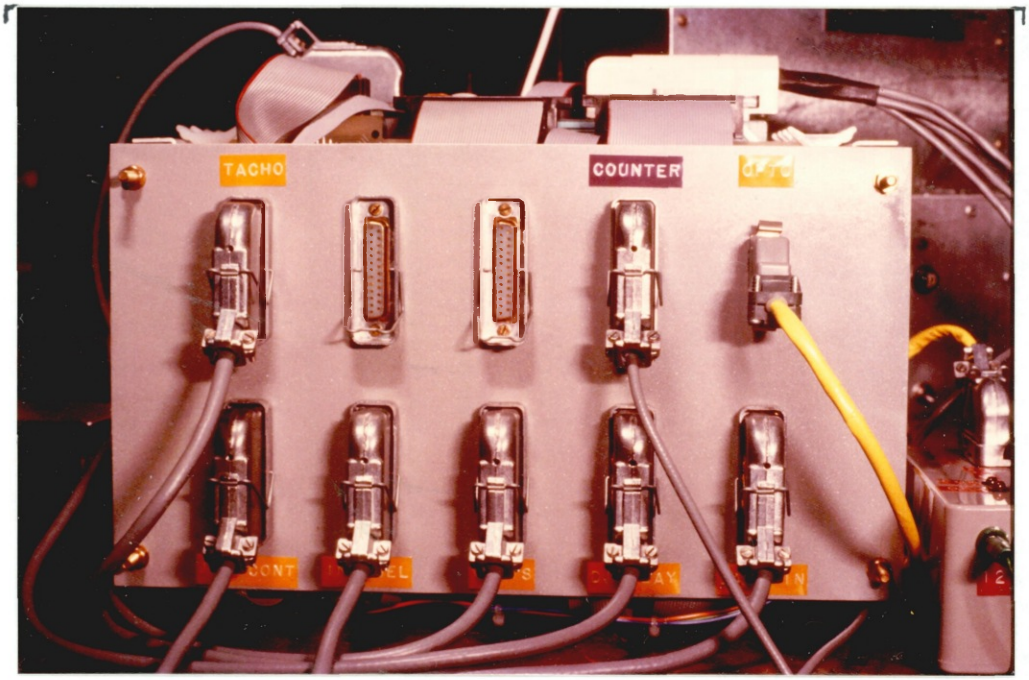


FIG. 4.45. THE CARD CAGE SIDE PANEL.

(a) C.C. TACHO socket linked to the 25 way D-socket on the Tachometer;

(b) C.C. COUNTER socket linked to the "CODE CON" 25 way D-socket on the Optical Position Detector Interface Unit;

(c) C.C. OPTO socket linked to the "INT CON" 25 way D-plug on the Optical Position Detector Interface Unit;

(d) C.C. INV CONT socket linked to the "INV CONT" 25 way D-socket on the 7 phase MOSFET inverter;

(e) C.C. INV SEL socket linked to the "INV SEL" 25 way D-plug on the 7 phase MOSFET inverter;

(f) C.C. VOLTS socket linked to the "VOLTS" 25 way D-socket on the keypad auxiliary logic unit;

(g) C.C. DISPLAY socket linked to the "DISP" 25 way D-socket on the keypad auxiliary logic unit;

(h) C.C. DATA IN socket linked to the "DATA" 25 way D-plug on the keypad auxiliary logic unit.

With the exceptions of the linking cables used for links (a) and (c), all other cables were wired such that a pin at one end of a given cable was connected to a pin with the corresponding number at the other end (i.e. pin x at end "A" was connected to pin x at end "B"). This straightforward "mapping" was not possible for the cables associated with the TACHO (link "a") and OPTO (link "c") sockets due to the arrangement of the signals at the peripherals. The pin-to-pin connections for cables "a" and "c" were therefore as shown in figs. 4.46 and 4.47 respectively.

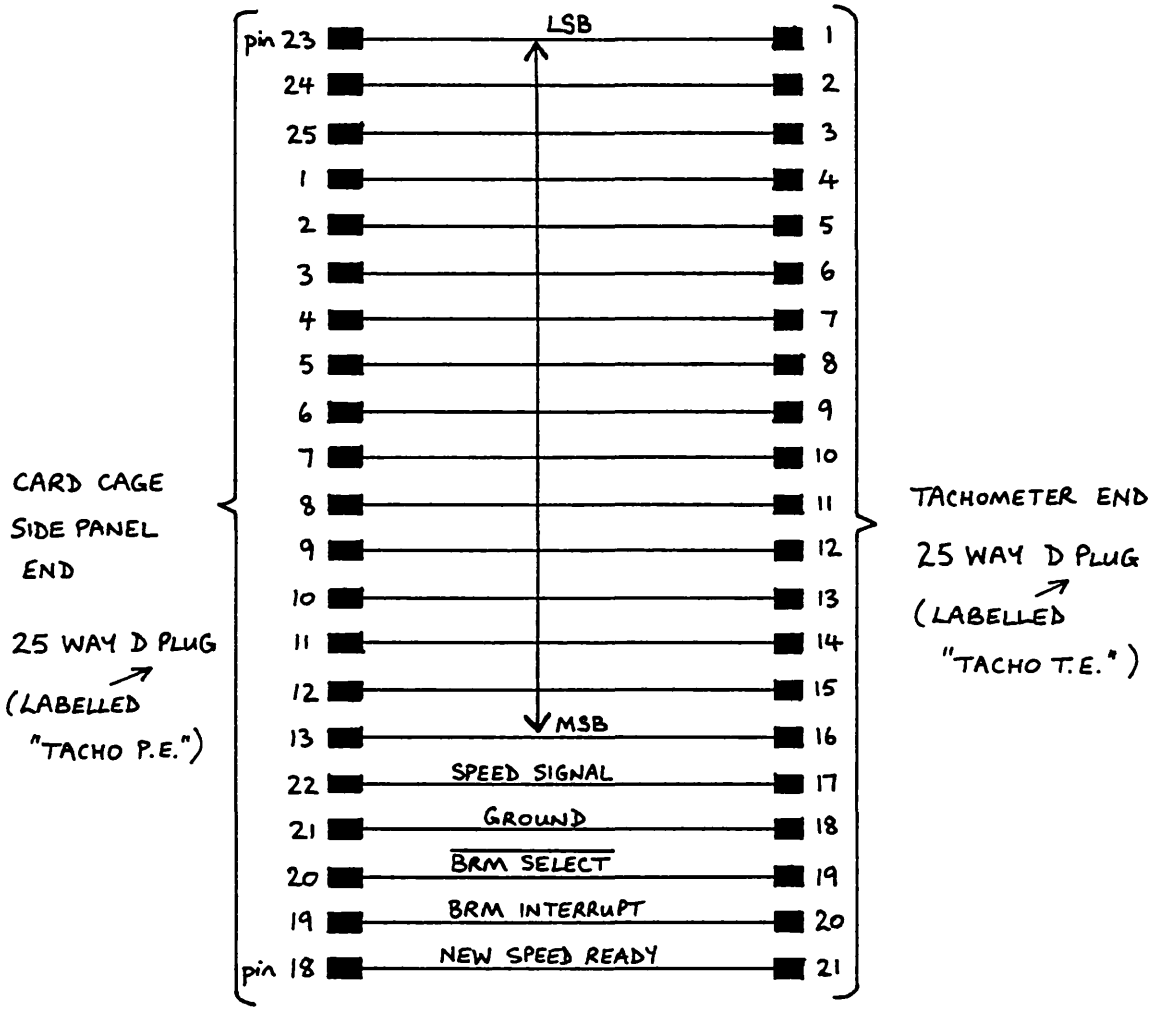


FIG. 4.46. INTERCONNECTING CABLE BETWEEN CARD CAGE SIDE PANEL AND THE TACHOMETER.

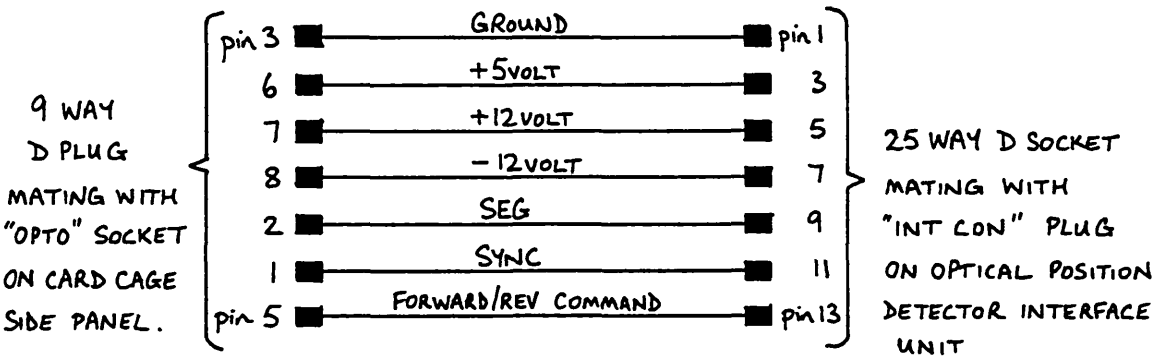


FIG. 4.47. INTERCONNECTING CABLE BETWEEN OPTICAL POSITION DETECTOR INTERFACE UNIT AND THE CARD CAGE OPTO SOCKET.

4.7.7.2 Connections between Various Sockets on the Card Cage Side Panel

To reduce the number of connections between the prototyping board and the card cage side panel sockets, some of the signals common to two or more sockets were routed from one socket to another on the side panel as follows:

(a) The ground connection between the prototyping board and the card cage side panel sockets is from P4-19 to the DISPLAY socket pin 1. The various other side panel sockets receive the ground line via the DISPLAY socket as shown in fig. 4.48.

(b) The "System Initialise Line" for the keypad logic is transmitted from pin 20 of the DISPLAY socket to the

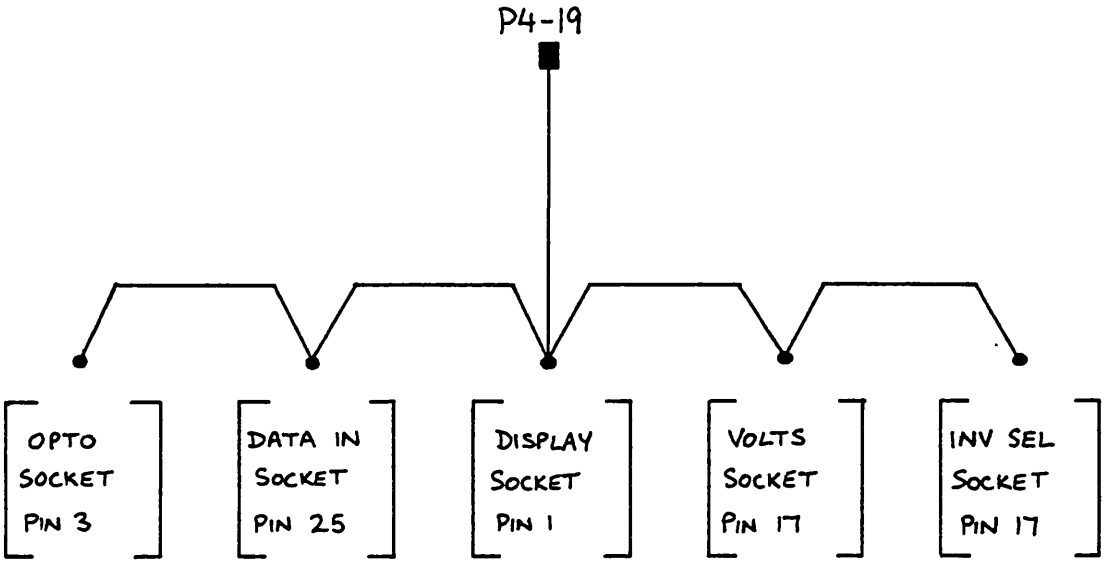


FIG. 4.48. THE GROUND CONNECTIONS ON THE CARD CAGE SIDE PANEL SOCKETS.

keypad auxiliary logic unit. The signal is obtained via a connection from pin 1 of the VOLTS socket to pin 20 of the DISPLAY socket.

(c) The "Forward/Reverse Command" line present on pin 5 of the OPTO socket is obtained via a connection from pin 23 of the DATA IN socket.

4.7.7.3 Summary of Hardwired Interrupt Connections

The connection of the eight interrupt lines available on edge connector P3 to the system peripherals is summarised in fig. 4.49.

4.7.7.4 The "System Initialise Line" Reset Button

As explained earlier, the "System Initialise Line" (CRU bit 8) returns to an inactive state when a reset func-

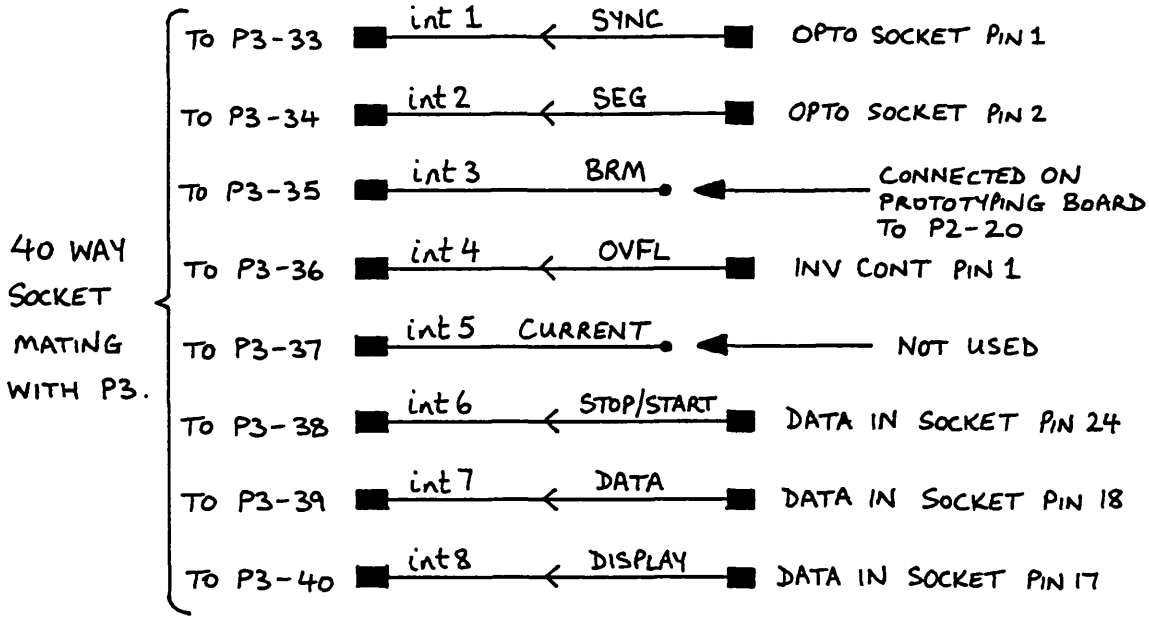


FIG. 4.49. INTERRUPT CONNECTIONS BETWEEN P3 AND THE CARD CAGE SIDE PANEL

tion is initiated. A RESET microswitch is available on the microcomputer board to initiate the reset function, but it was desired to have an easily accessible "panic" button that could be hit in an emergency. The $\overline{\text{PRES}}$.B connection available on the card cage back panel is connected to the microcomputer board via its P1 connector. When $\overline{\text{PRES}}$.B is taken to logic 0 it initiates a reset. Therefore, a microswitch was connected between the $\overline{\text{PRES}}$.B and ground connections. It was located conveniently close to the keypad and VDU so that it could be reached easily. A 47μF tantalum capacitor was added to the microcomputer board to debounce the external reset switch (see p5-3 of reference 4.13 for details).

4.7.7.5 Card Cage Circuit Board Photographs

A photograph of the four circuit boards making up the microprocessor control system is shown in fig. 4.50. A close-up photograph of the prototyping board, showing the interfacing electronics is shown in fig. 4.51.

4.7.8 Extra Hardware Required to Interface the Magslip System to the Microprocessor

The magslip inverter (described in Chapter 3) does not have any "Inverter Enable" circuitry, and so the inverter enable word available from the INV SEL socket on the card cage side panel is superfluous. However, it was thought sensible to arrange the magslip inverter control signals so that they would assume an inactive state if the "System Initialise Line" was not enabled. Therefore, the circuit shown in fig. 4.52 was added into the magslip inverter control signal lines. The three phases of the magslip inverter are controlled by the six signals: A1, B1, C1, A2, B2, and C2. The inverting action of the NOR gates shown in fig. 4.52 is required so that active-low control signals from the microprocessor can be used. Each inverter control signal is related to a particular bit of the microprocessor's data bus as follows: A1 = $\overline{D10}$, B1 = $\overline{D11}$, C1 = $\overline{D12}$, A2 = $\overline{D13}$, B2 = $\overline{D14}$, and C2 = $\overline{D15}$. (Note that C2 is the least significant bit of the bus). The LED in fig. 4.52 indicates when the NOR gates are enabled by the System Initialise Line.

A small modification to the ±50 volt power supply of the magslip was also necessary, so that it could be controlled via the D/A circuitry housed in the keypad auxiliary logic unit. The simple switch arrangement shown in fig. 4.53 allows the power supply programming voltage to be supplied either from a manually operated potentiometer, or from the microprocessor controlled D/A circuits. One of the voltage outputs of the D/A circuits is used to supply

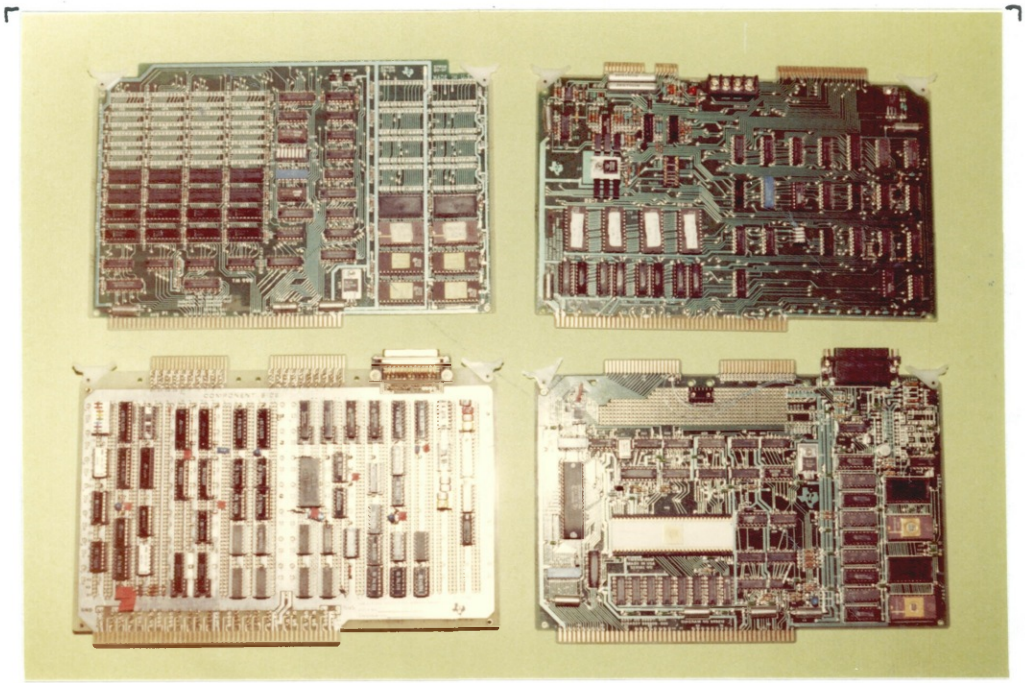


FIG. 4.50. THE FOUR MICROPROCESSOR CONTROL SYSTEM BOARDS
TOP LEFT :- TM990/201-41 ; TOP RIGHT :- TM990/302-1
BOTTOM LEFT :- PROTOTYPING BOARD ; BOTTOM RIGHT :- TM990/100M-1

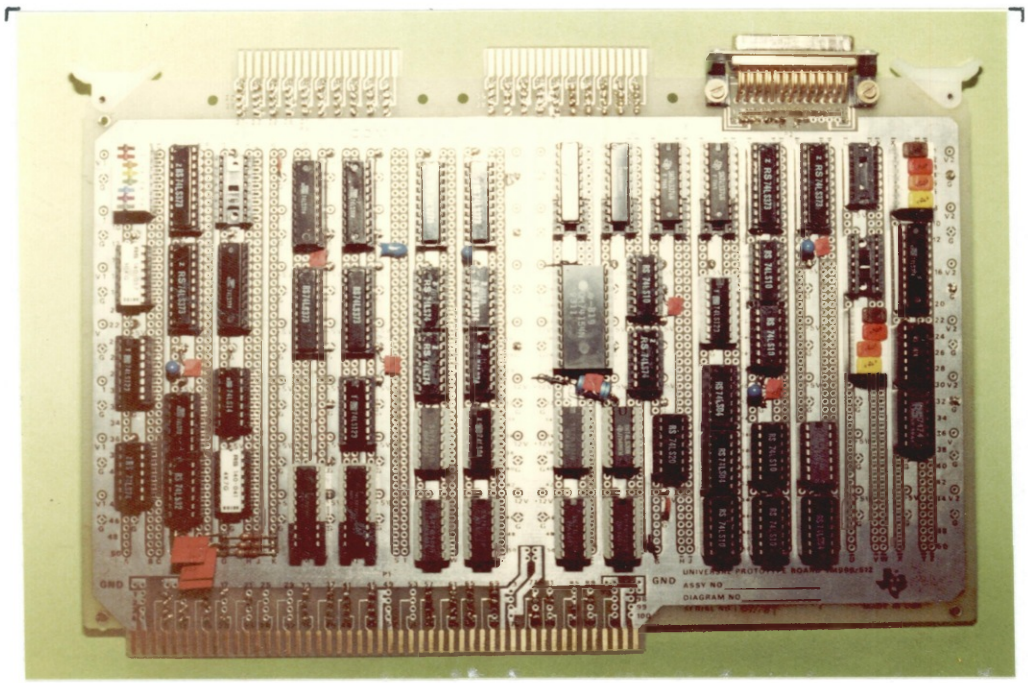


FIG. 4.51. THE PROTOTYPING BOARD.

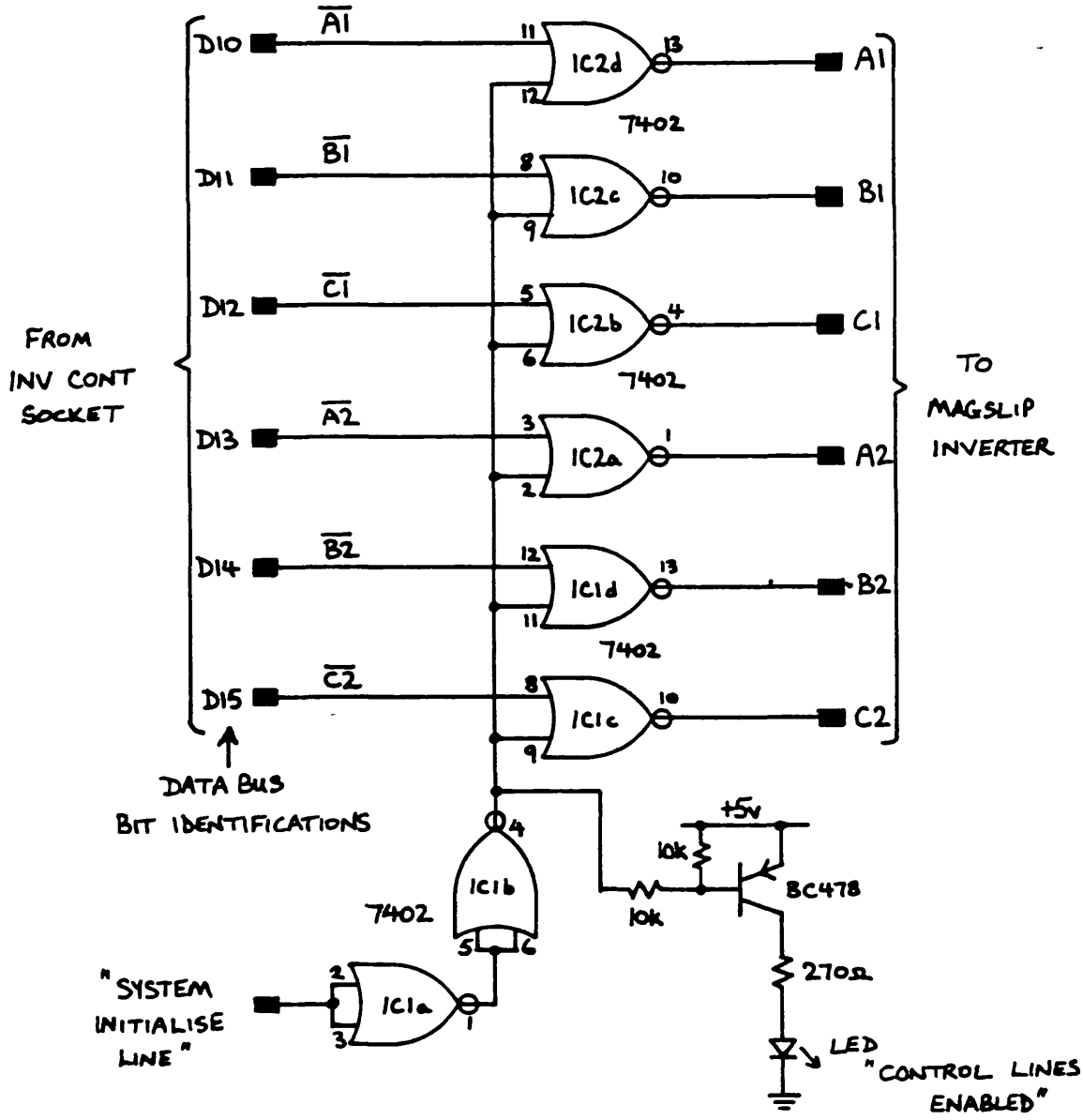


FIG.4.52. MAGSLIP INVERTER "ENABLE" CIRCUIT.

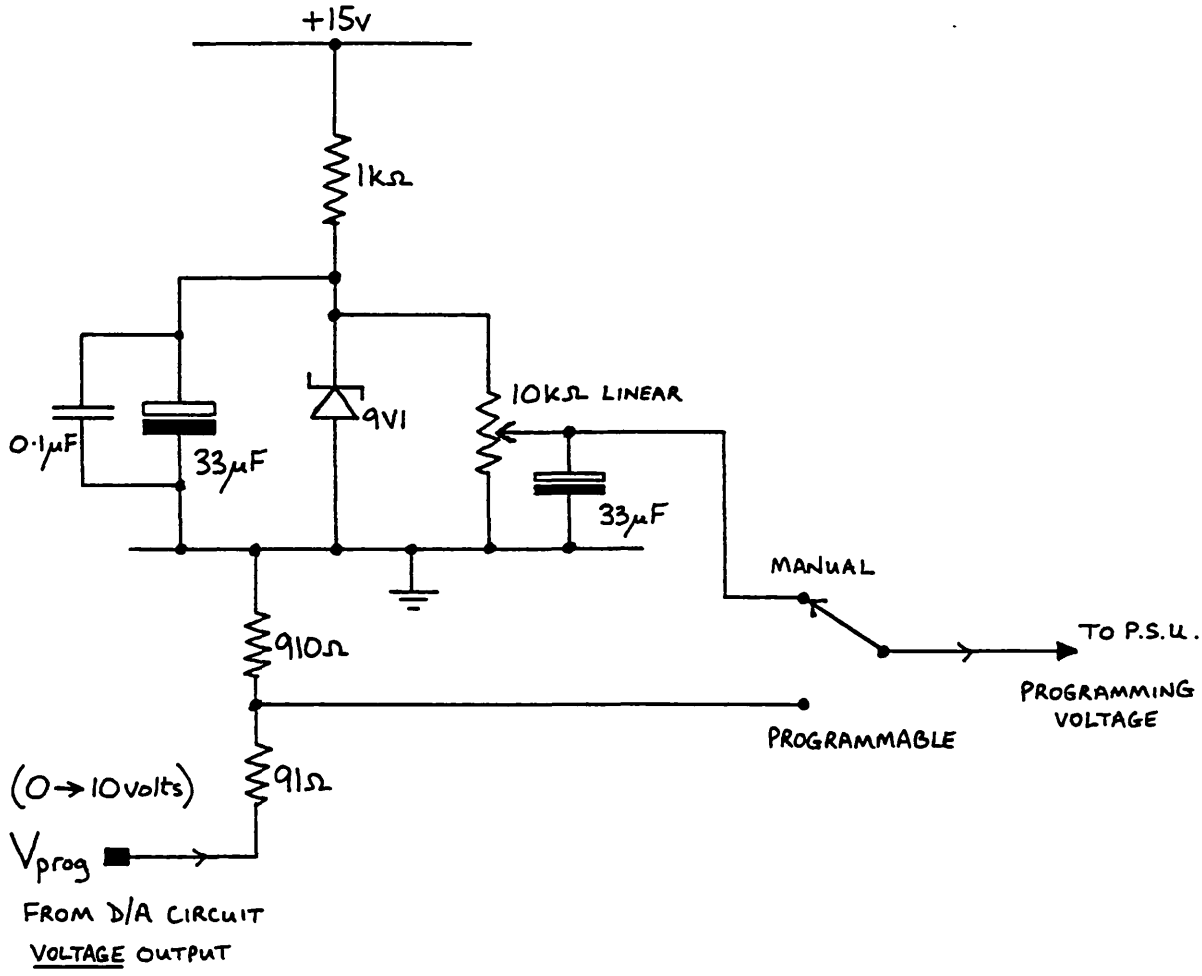


FIG.4.53. THE ±50 VOLT P.S.U. PROGRAMMING VOLTAGE SWITCH ARRANGEMENT.

the control signal V_{prog}. The potential divider, formed by the 910Ω and 91Ω resistors, scales the programming voltage from the microprocessor so that the maximum programming voltage can just demand a ±50 volt output from the supply.

The various signals between the magslip system and the microprocessor system were connected as indicated in fig. 4.54.

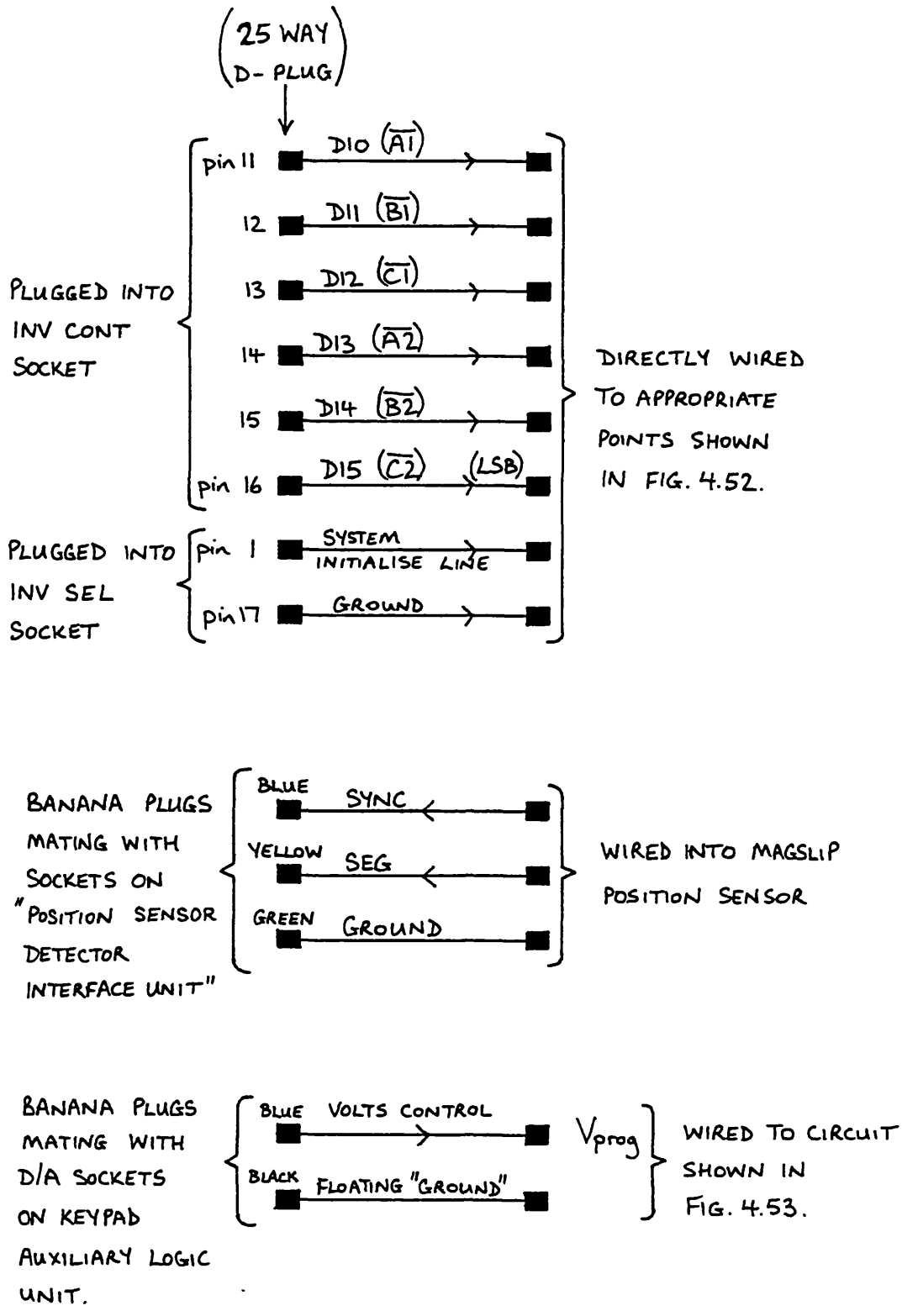


FIG. 4.54. INTERCONNECTIONS BETWEEN MAGSLIP SYSTEM AND MICROPROCESSOR SYSTEM

4.8 The Development of Microprocessor Programs Capable of Controlling a High Speed Autopiloted Synchronous Motor System

The primary aim of the work involving the microprocessor system was to autopilot the 7 phase square wave motor up to speeds of 30000 rpm. However, the 7 phase motor was a totally untried device and it was realised that the problems of debugging the control software would be much greater whilst the 7 phase motor was part of the control loop. What was required was a reliable piece of motor hardware on which to develop the necessary software. Having developed the software it would then be possible to introduce the 7 phase motor into the system and concentrate on its performance. Hence, it was decided to use the magslip motor system during the software development. An additional bonus of using the magslip was that it gave an opportunity to implement adjustable load angle control on the magslip, using a microprocessor in place of the discrete electronics described in Chapter 3.

Two programs were written for the control of the magslip motor system. The first was based on the use of interrupts for the detection of rotor position; the second was based on the sampling of a rotor position code. The programs were written so that they were also basically suitable for the control of the 7 phase motor. Hence the motor reversal was achieved by means of stator flux control so that the permanent magnet 7 phase motor could be reversed. Every effort was made to minimise the number of modifications needed to convert the successful magslip programs into 7 phase motor control programs. The idea behind this philosophy was that a minimal number of changes would reduce the chances of software bugs being inadvertently introduced. To ensure that the interrupt based magslip program would be more or less suitable for the control of the 7 phase motor, it was therefore necessary to make the execution times of the autopiloting interrupt service routines short enough to

allow correct program operation with the 7 phase motor rotating at 30000 rpm. To simplify the change over from one motor to the other, the hardware described in sub-section 4.7.8 was arranged so that the magslip system could be connected into the microprocessor system as a direct replacement for the 7 phase motor. This meant that no microprocessor system hardware changes were necessary.

The following sub-sections outline the software development work that was performed for the magslip and 7 phase motor systems.

4.8.1 The Basic System Tasks for the Microprocessor to Perform

The main jobs for the microprocessor system were identified as the following:

1. to initialise the system and hold active peripherals such as the inverter in a safe state until otherwise instructed;

2. to start the motor rotating in a particular direction when requested and subsequently synchronise it if possible;

3. to keep the motor synchronised at any speed within its operating range;

4. to stop the motor when requested and set the system safe;

5. to receive data and commands from the operator at any time during the motor operation and successfully act on them;

6. to display system data if requested to do so.

Optional jobs for the microprocessor were identified as:

1. the implementation of "outer" control loops intended for the regulation of such parameters as speed or phase current. Other control functions could include the calculation of optimum operating load angle for the magslip motor using an algorithm based on equation 2.53, and so forth;

2. the monitoring of the "health" of the system including the magnitude of quantities such as phase current and winding temperature. During the design of the microprocessor interrupt latch circuitry, the int 5 interrupt was arbitrarily designated for use as the CURRENT interrupt to allow current overloads in the system to be detected if desired, but it could of course be used to sense other limiting quantities.

The lack of time available and the desire to concentrate on the basic autopiloting task meant that the optional tasks listed above were not implemented on the magslip or 7 phase motor systems.

The design of the interfacing hardware described in section 4.7 is such that the operations that the microprocessor is required to perform on the motor system are very simple indeed, although they occur very frequently. Several important objectives were achieved by way of the hardware design, including the aim that the system should to some extent be failsafe (achieved by the use of active-low control signals for the TTL inputs of the inverter and various other signal lines); the microprocessor should be "invisible" to the operator so that the system is easy to operate (achieved by the use of a keypad input peripheral); and the system should be easy to disable rapidly (achieved by the use of the microprocessor reset function). The one important objective of the software part of the system was that the speed of program execution should be fast enough to meet the REAL TIME demands of the system: the interrupts

associated with the autopiloting functions have to be acted upon as soon as possible to ensure synchronism is maintained.

4.8.2 Interrupt Based Autopiloting Program Flowchart

The philosophy of the interrupt based programs used to control the magslip and the 7 phase motor is indicated in the basic flowchart shown in fig. 4.55. The basic flowcharts for the SYNC, SEG, STOP/START, DATA, and DISPLAY interrupts are shown respectively in figs. 4.56 to 4.60 inclusive. The operations indicated in the flowcharts are necessarily brief and idealised: this is especially so for the main program flowchart shown in fig. 4.55. For example, changes in the motor variables are executed over a number of successive loops through the program; this results in the variables changing value gradually and not abruptly.

4.8.3 The Magslip System Interrupt-Based Control Program

The magslip system TMS9900 assembly language program, based on the flowcharts shown in figs. 4.55 to 4.60 inclusive, is given in Appendix 4A. It should be noted that hexadecimal numbers in the program are indicated by a > symbol: e.g. $>100 = 100_{16}$. The assembled object form of the program has not been given because of its excessive length. In its assembled form the program occupies 616_{16} (1558_{10}) 16 bit words of memory starting from address 1040_{16}. It is not possible to discuss every aspect of the program without getting involved in great detail, and this in itself would hinder any attempt to explain its operation. Therefore, the main features have been included in the more detailed flowchart shown in figs. 4.61(a) to (r) inclusive, and the following discussion on the program operation is based on the flowchart. Arrows alongside the flowchart indicate the areas where interrupts can occur. The relationship between a particular box in the flowchart, and the corresponding lines of instructions in the assembly language source listing in Appendix 4A, is indicated by line numbers

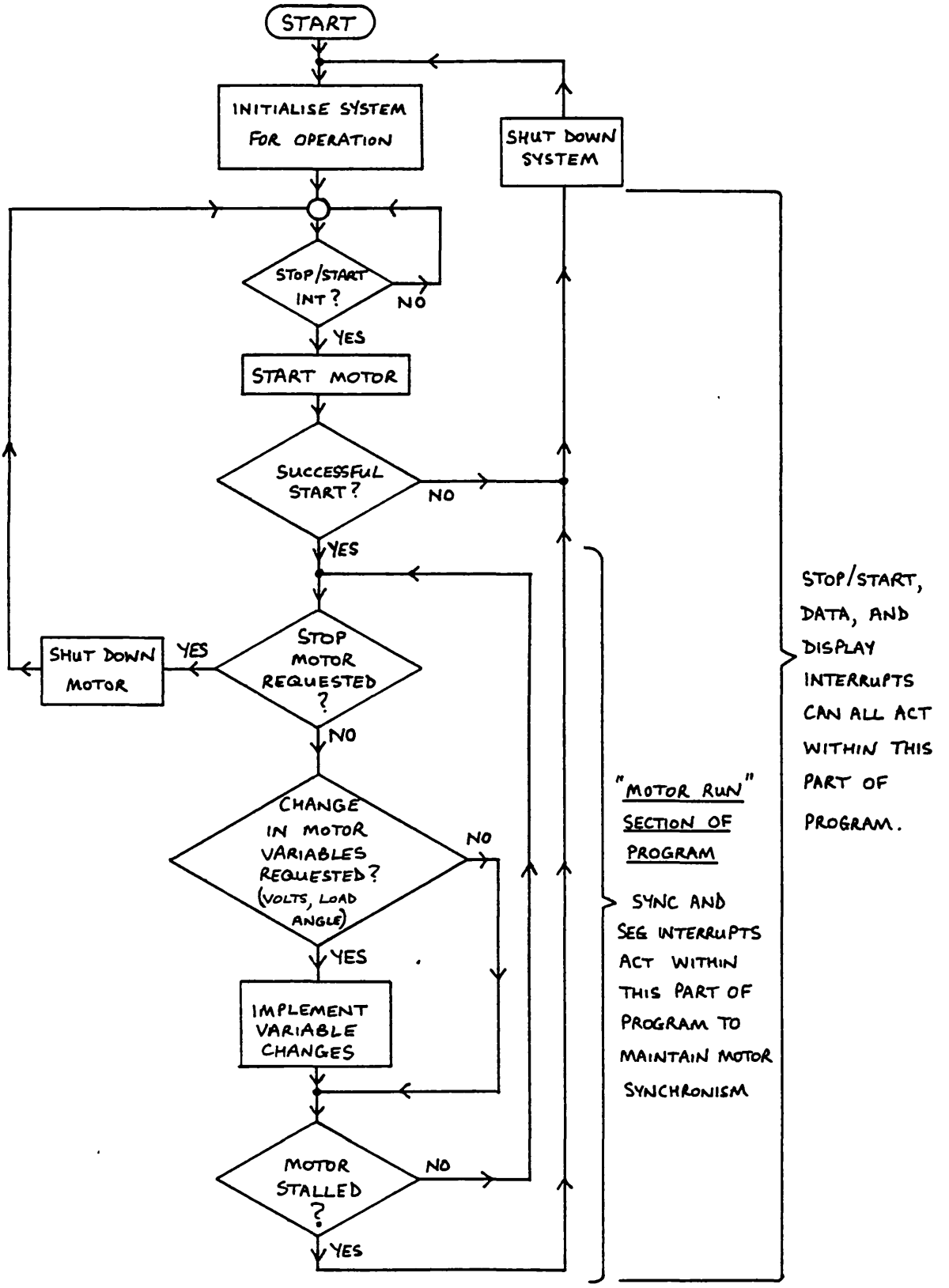


FIG. 4.55. MAIN PART OF INTERRUPT BASED MOTOR CONTROL PROGRAM (REPRESENTED IN FLOWCHART FORM)

int 1 : SYNC INTERRUPT SERVICE ROUTINE

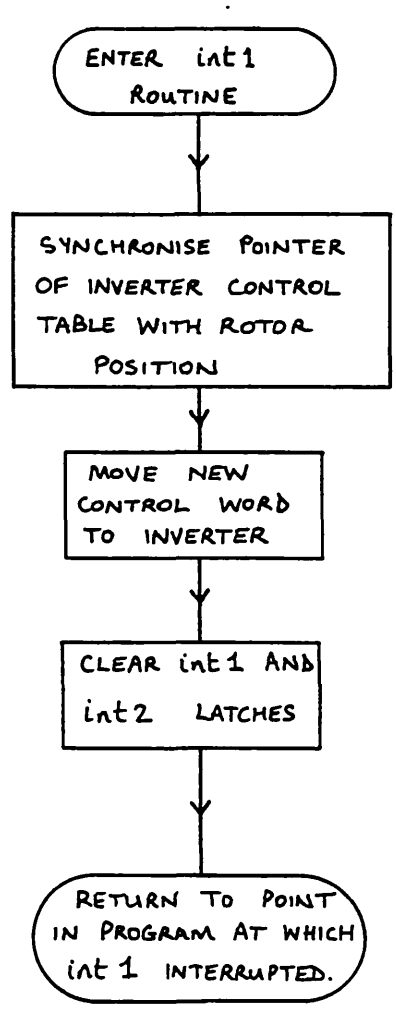


FIG. 4.56. BASIC FLOWCHART FOR SYNC (int 1) INTERRUPT SERVICE ROUTINE

int 2 : SEG INTERRUPT SERVICE ROUTINE

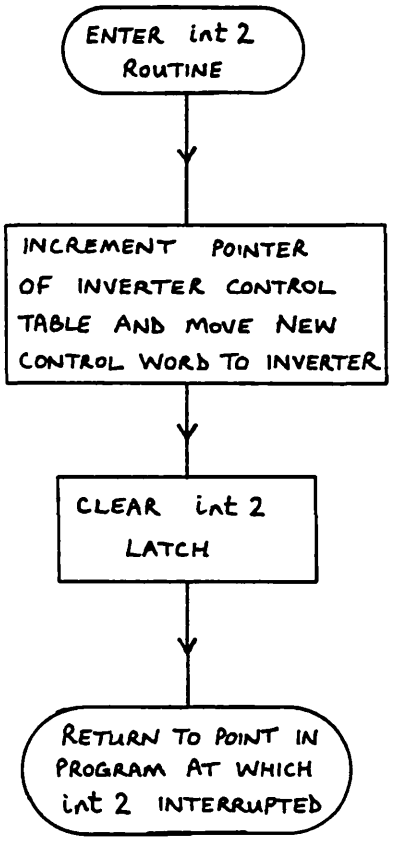


FIG. 4.57. BASIC FLOWCHART FOR SEG (int 2) INTERRUPT SERVICE ROUTINE

int 6 : STOP/START INTERRUPT SERVICE ROUTINE

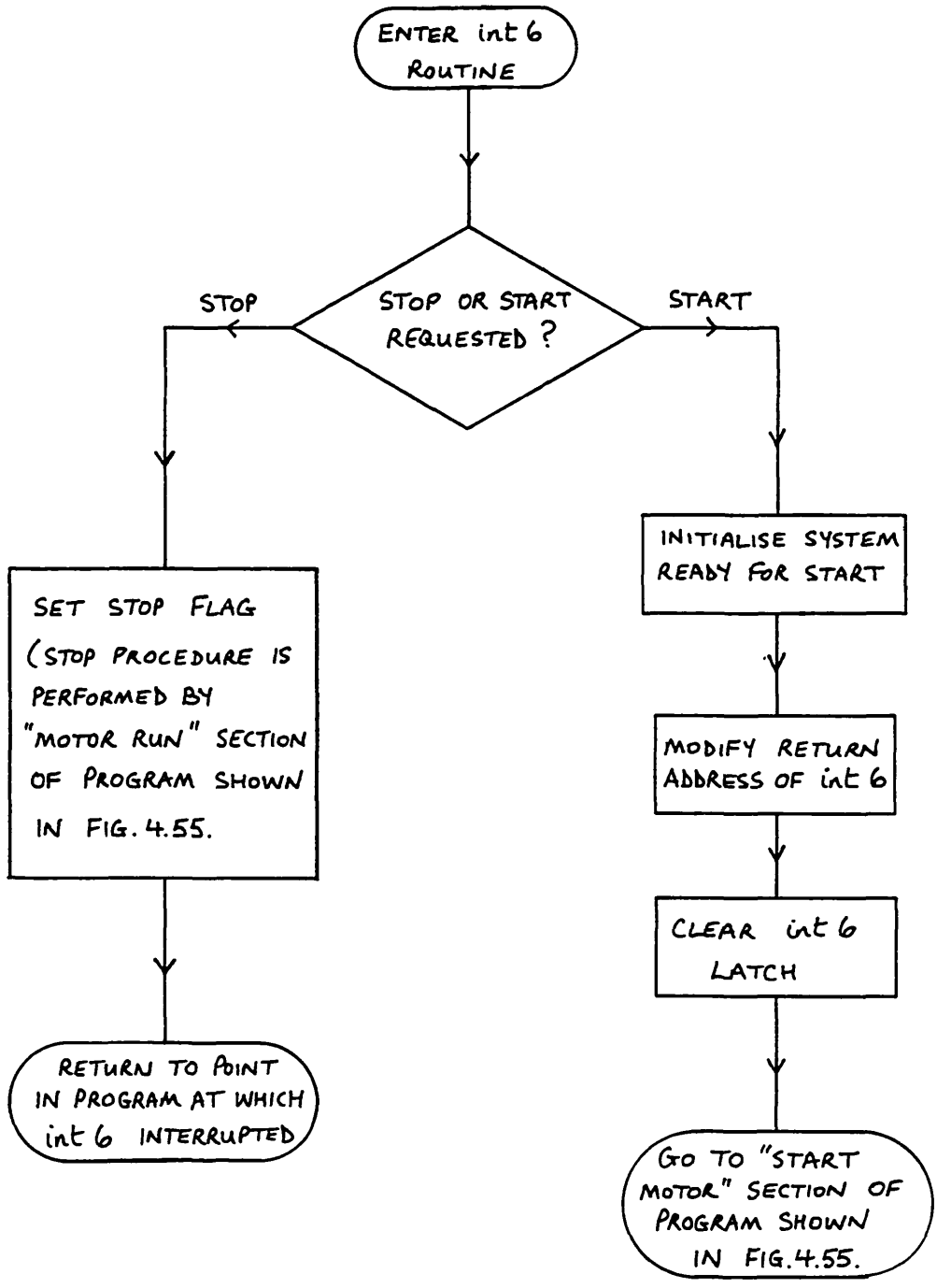


FIG. 4.58. BASIC FLOWCHART FOR STOP/START (int 6) INTERRUPT SERVICE ROUTINE

int 7 : DATA INTERRUPT SERVICE ROUTINE

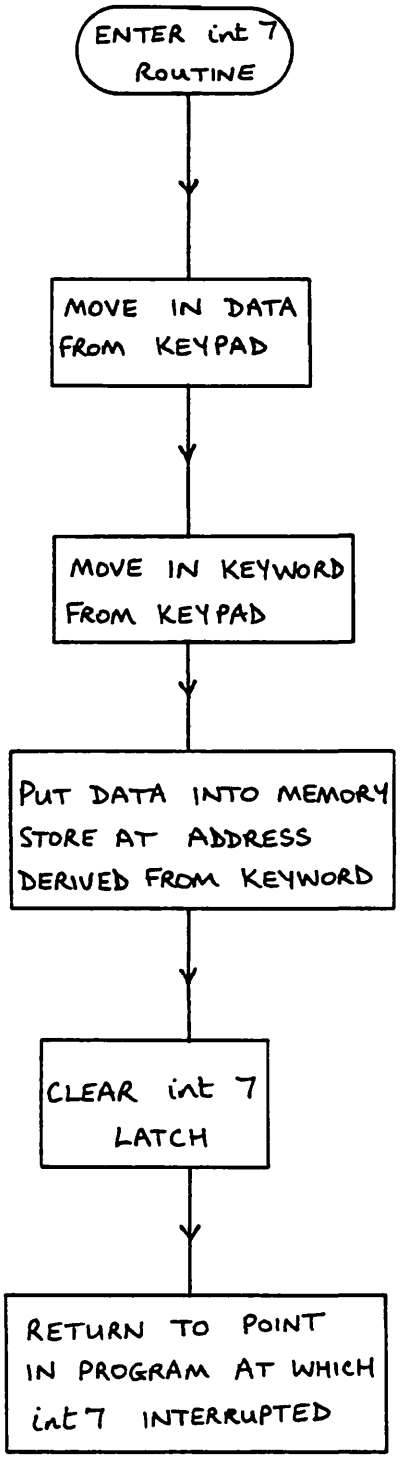


FIG. 4.59. BASIC FLOWCHART FOR DATA (int 7) INTERRUPT SERVICE ROUTINE

int 8 : DISPLAY INTERRUPT SERVICE ROUTINE

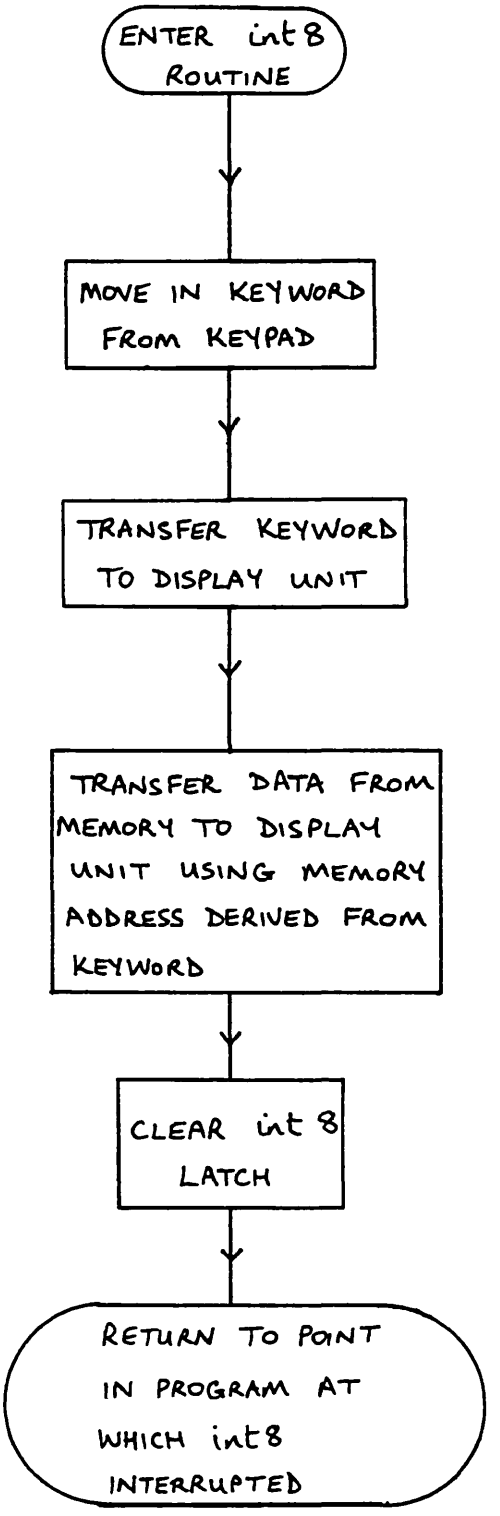


FIG. 4.60. BASIC FLOWCHART FOR DISPLAY (int 8) INTERRUPT SERVICE ROUTINE

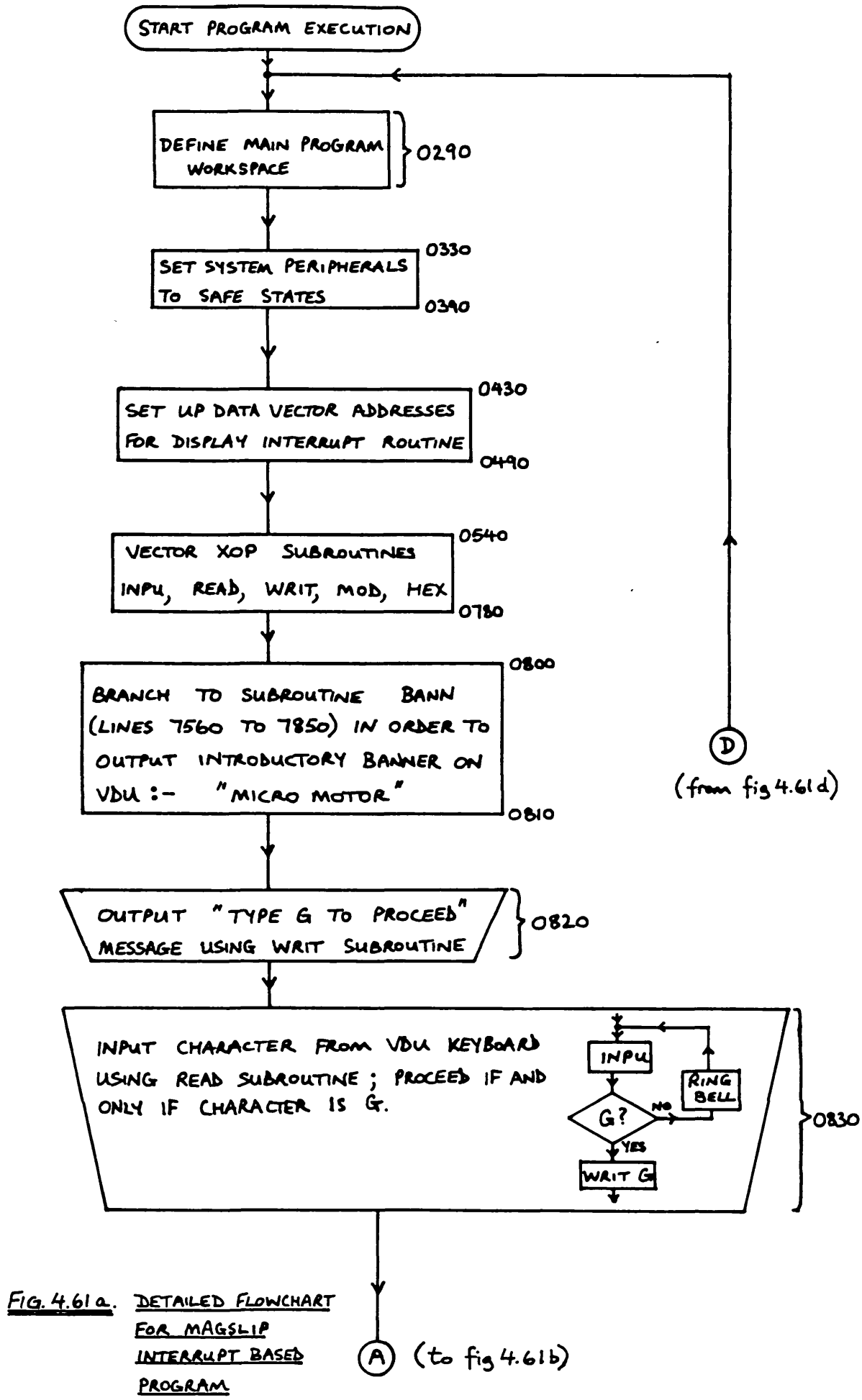


FIG. 4.61a. DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM

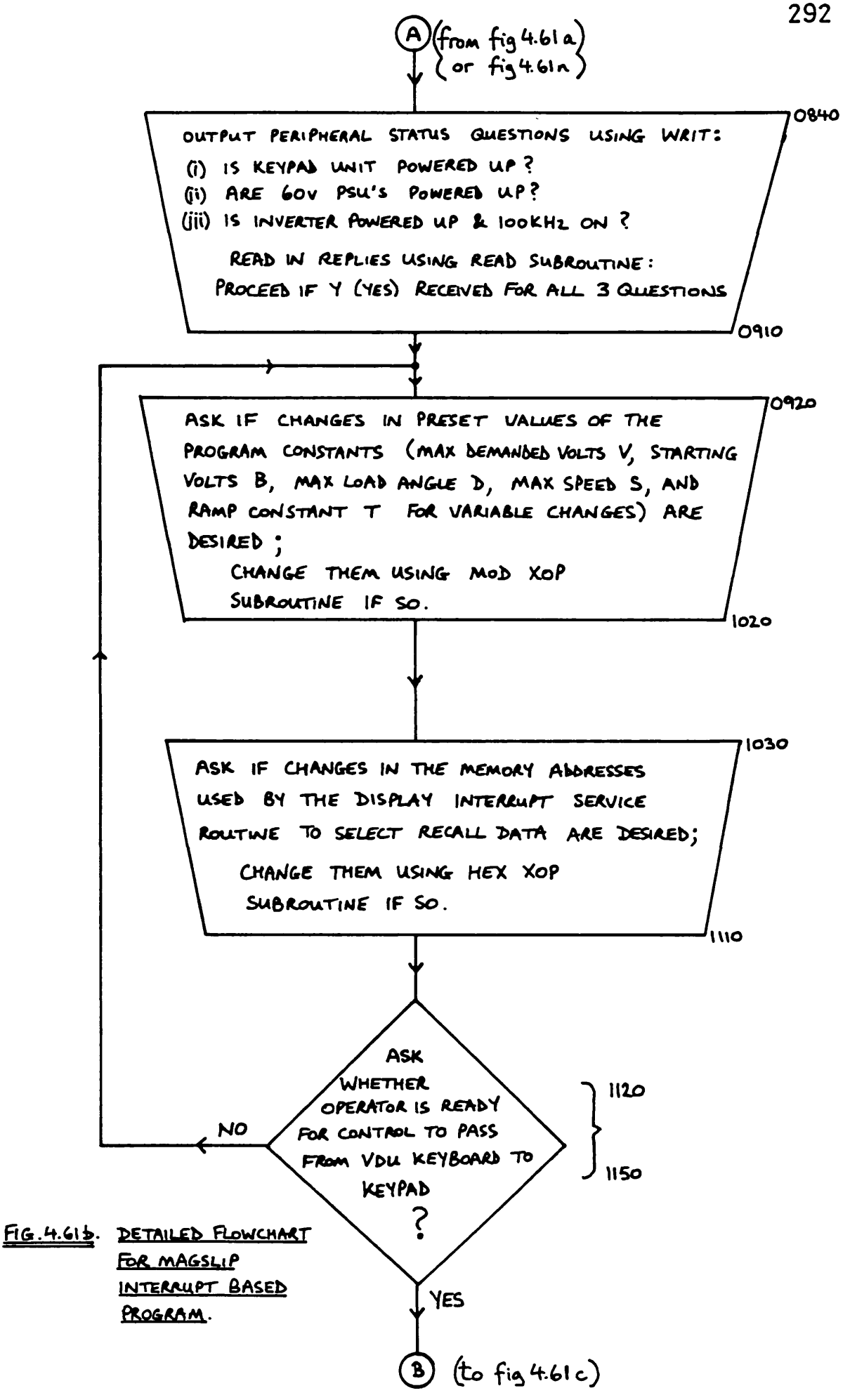


FIG. 4.61b. DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM.

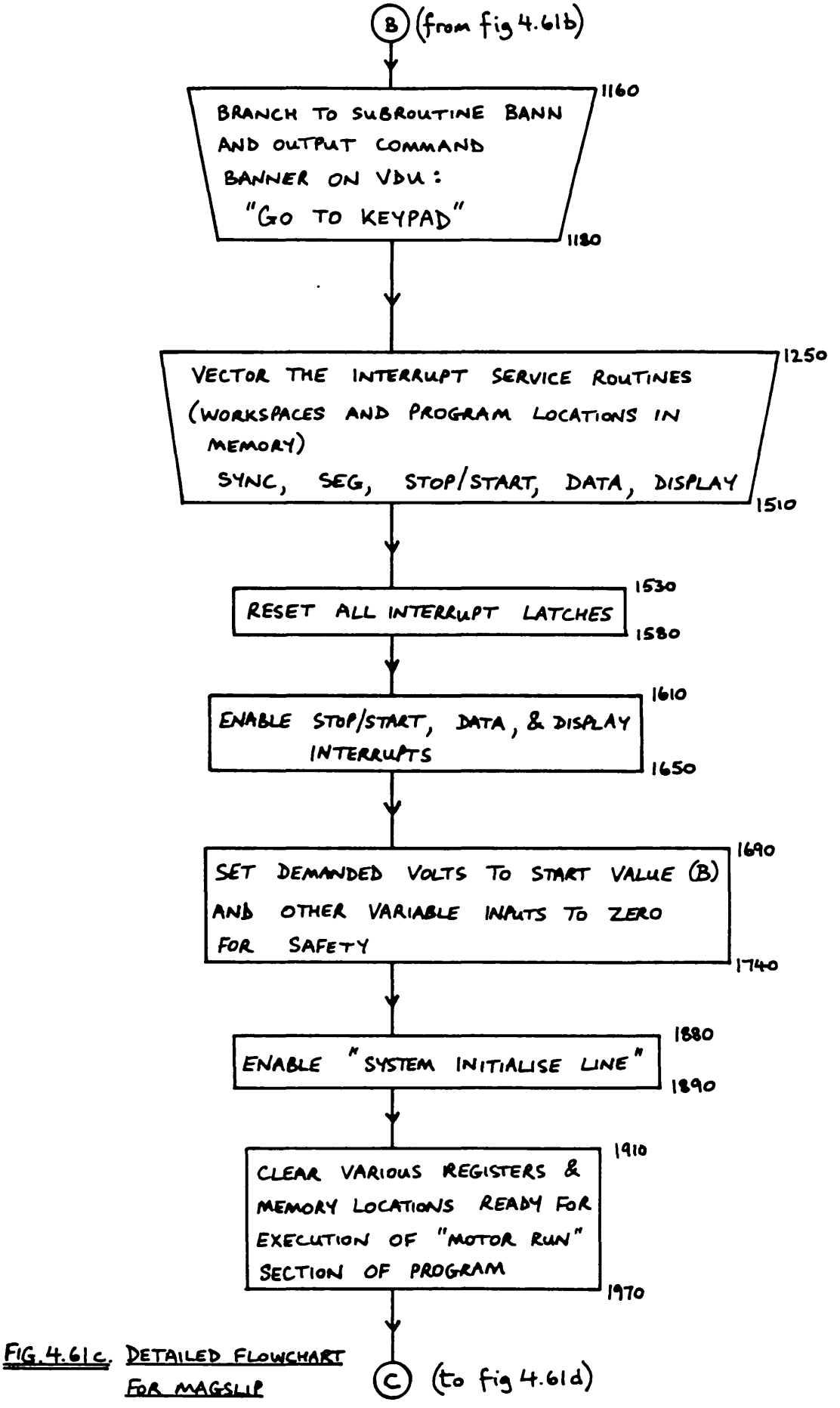


FIG. 4.61c. DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM.

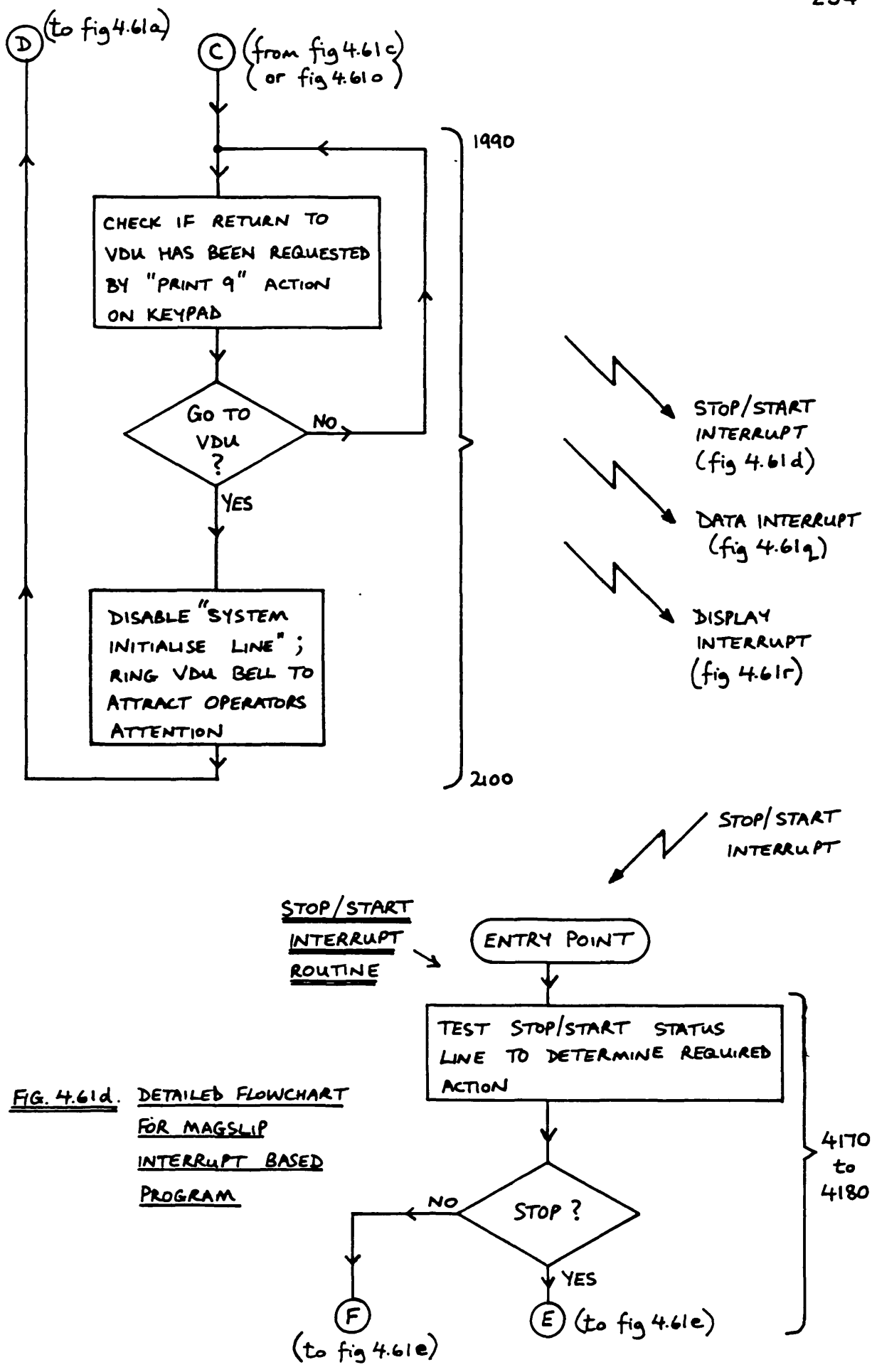


FIG. 4.61d. DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM

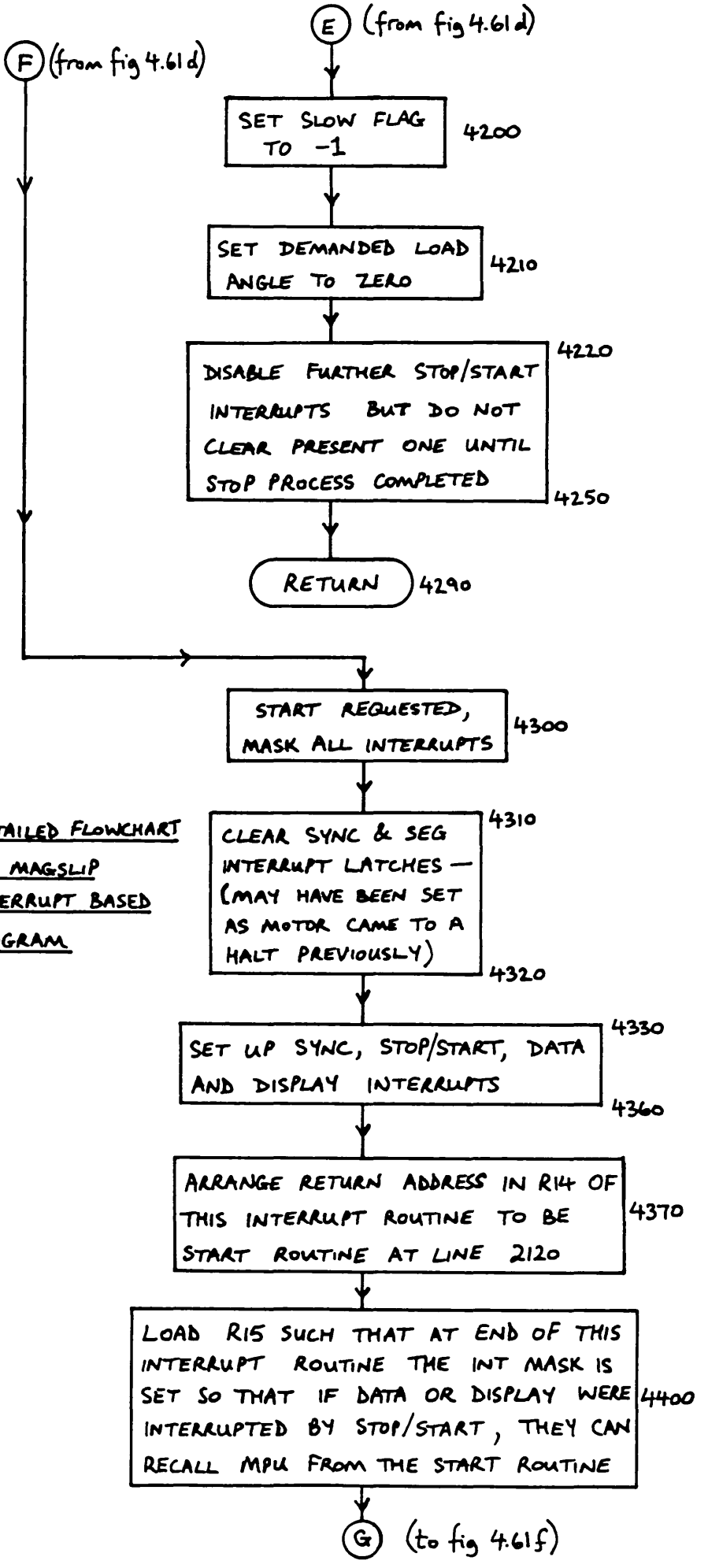


FIG. 4.61e. DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM

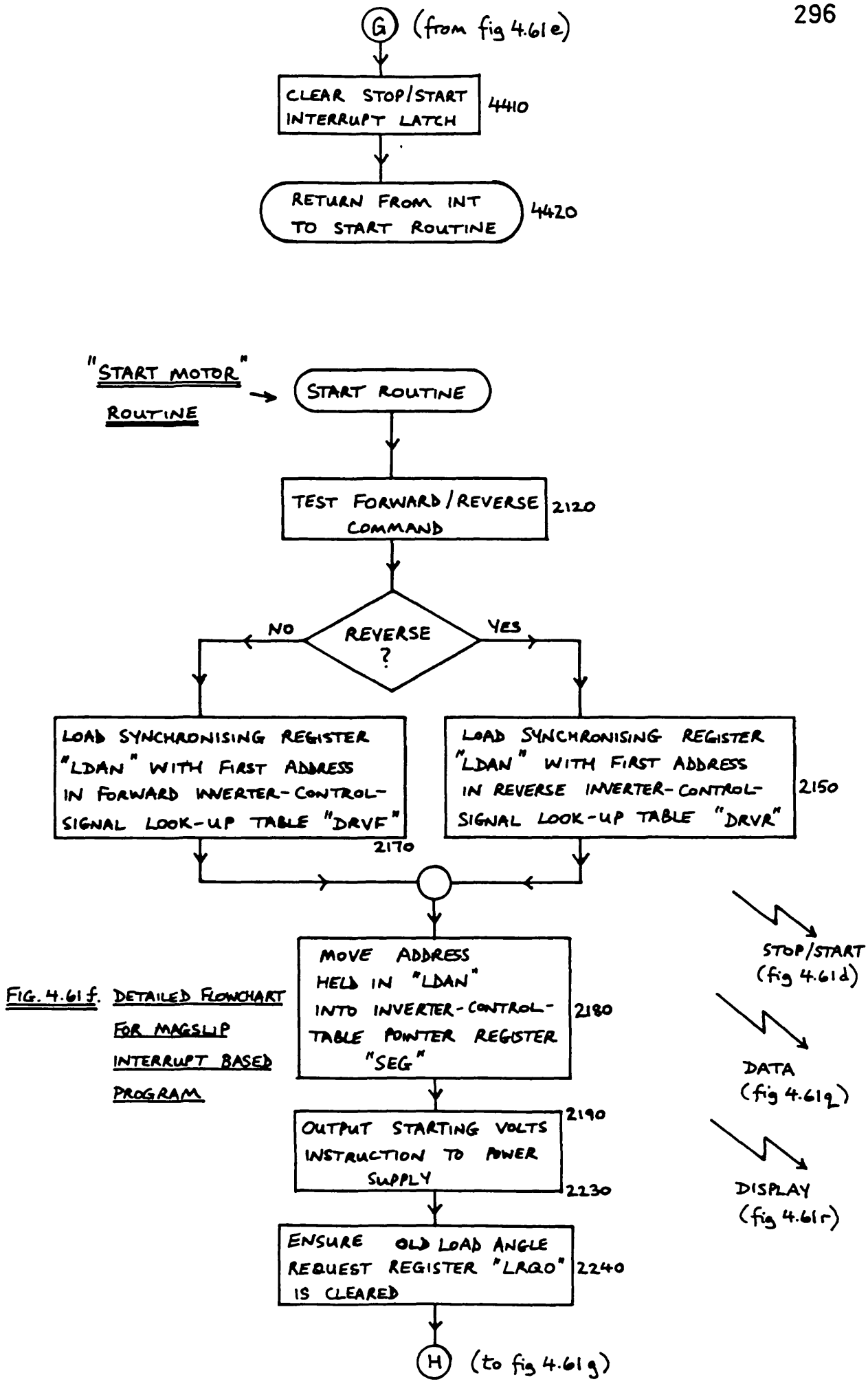


FIG. 4.61f. DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM

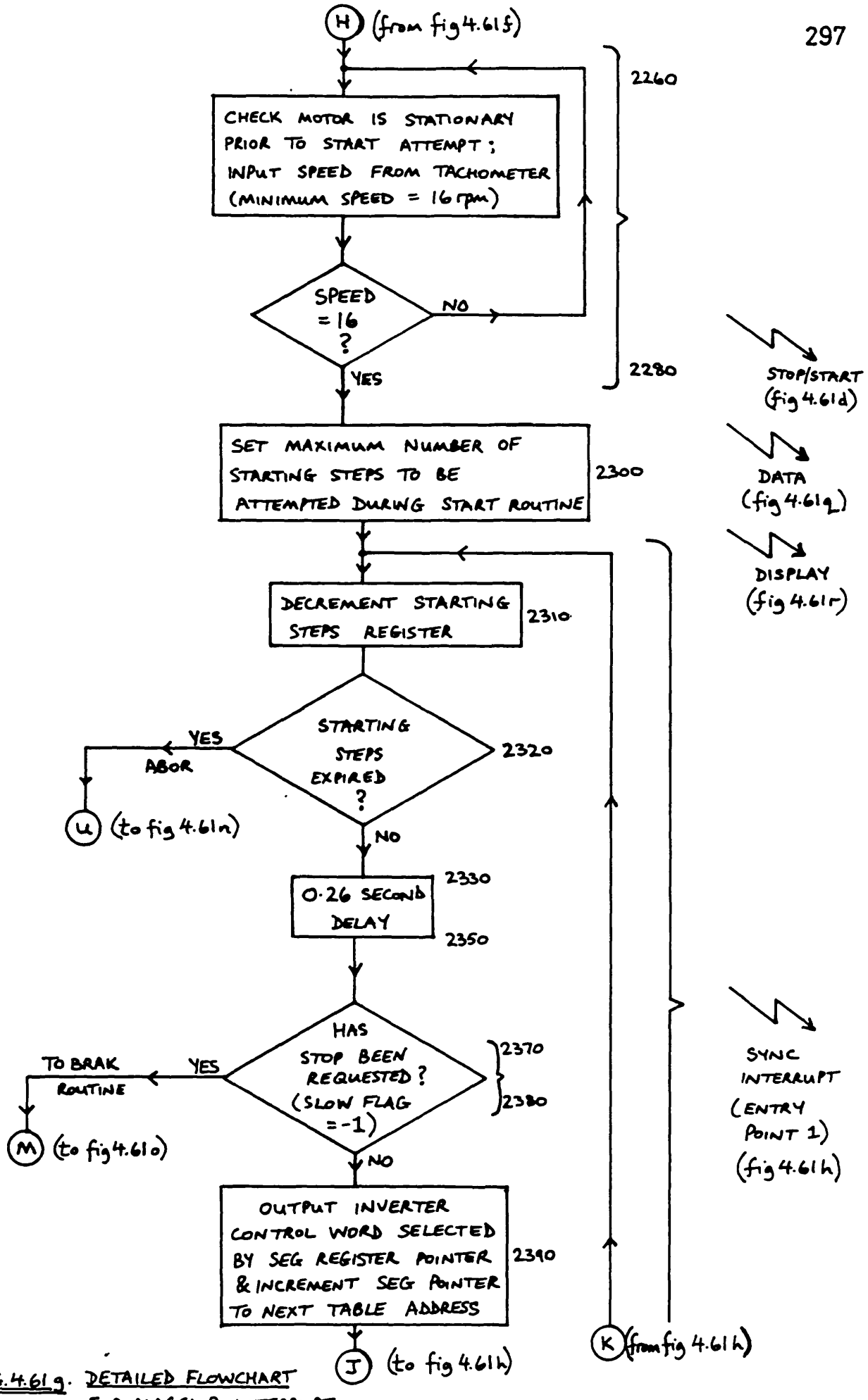


FIG. 4.61g. DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM

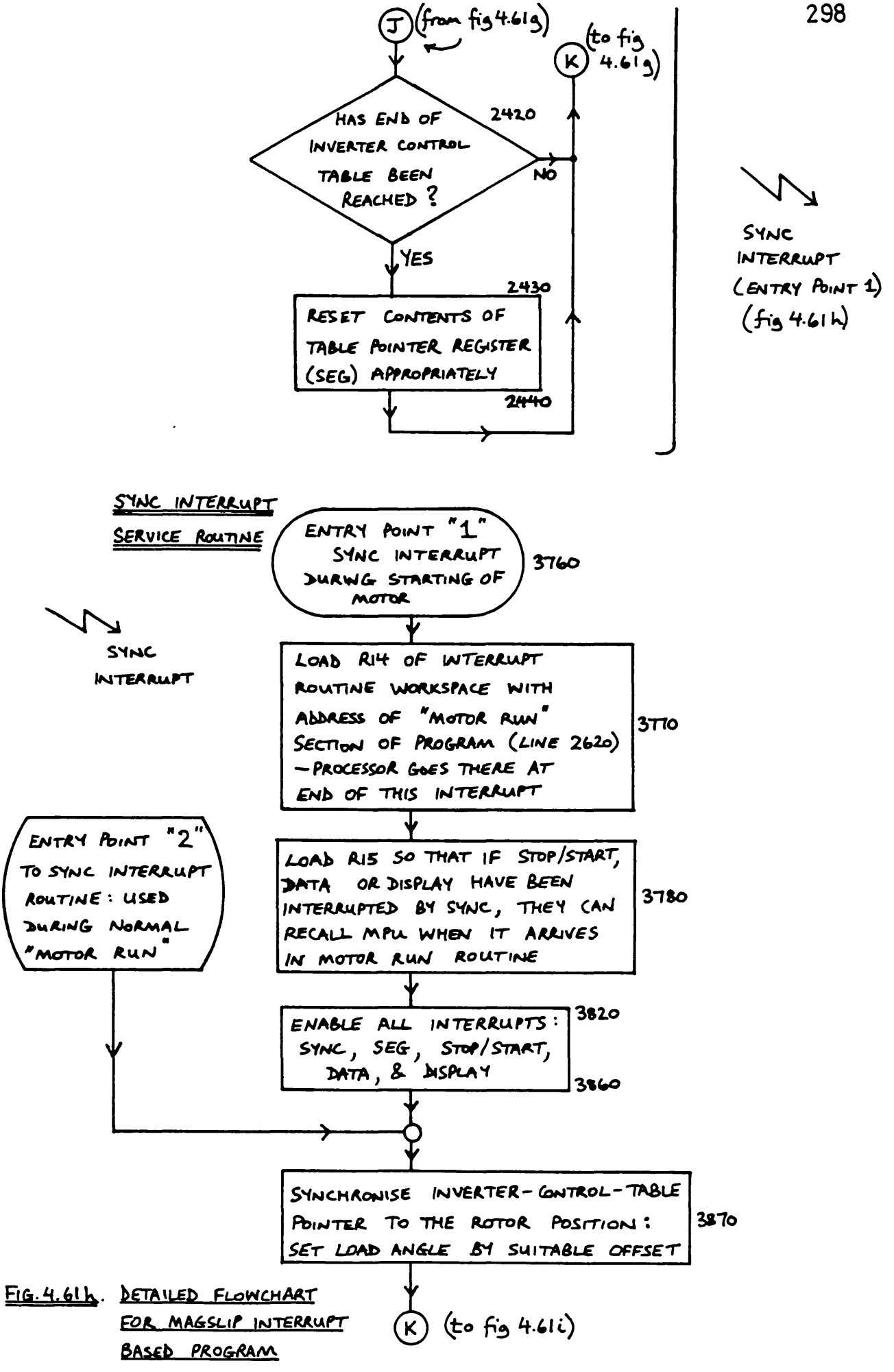


FIG. 4.61h. DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM

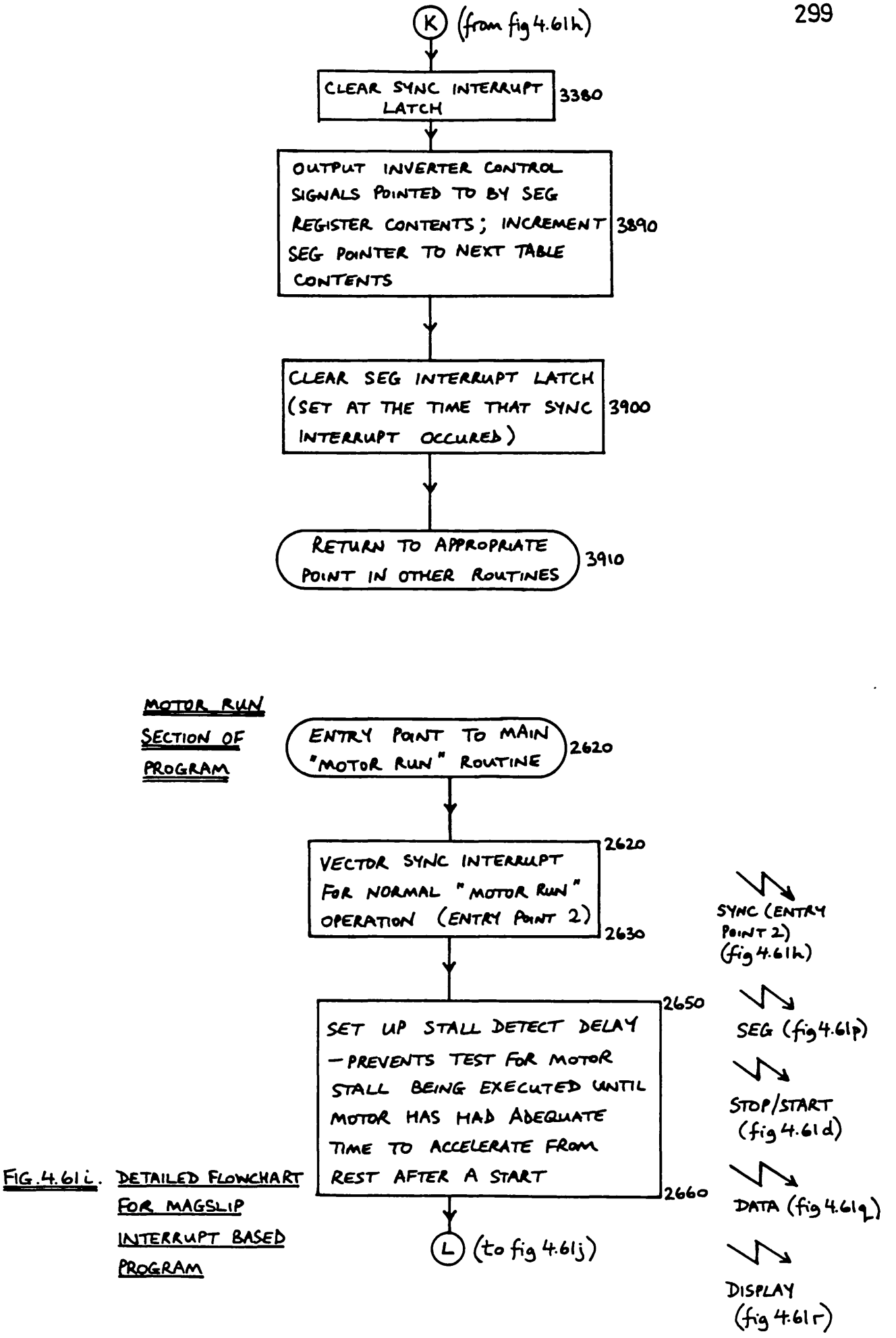


FIG. 4.61i. DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM

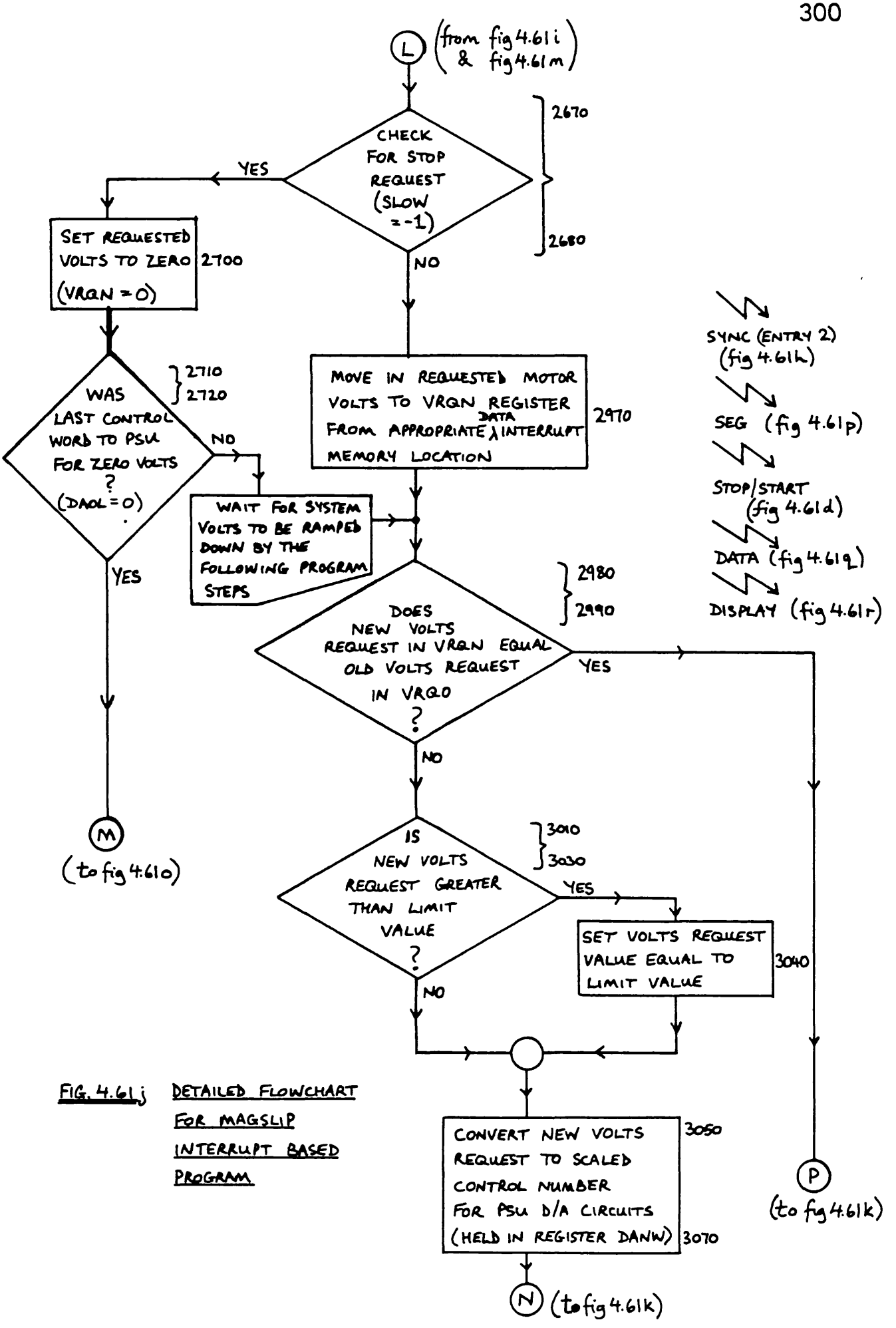


FIG. 4.61 j DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM

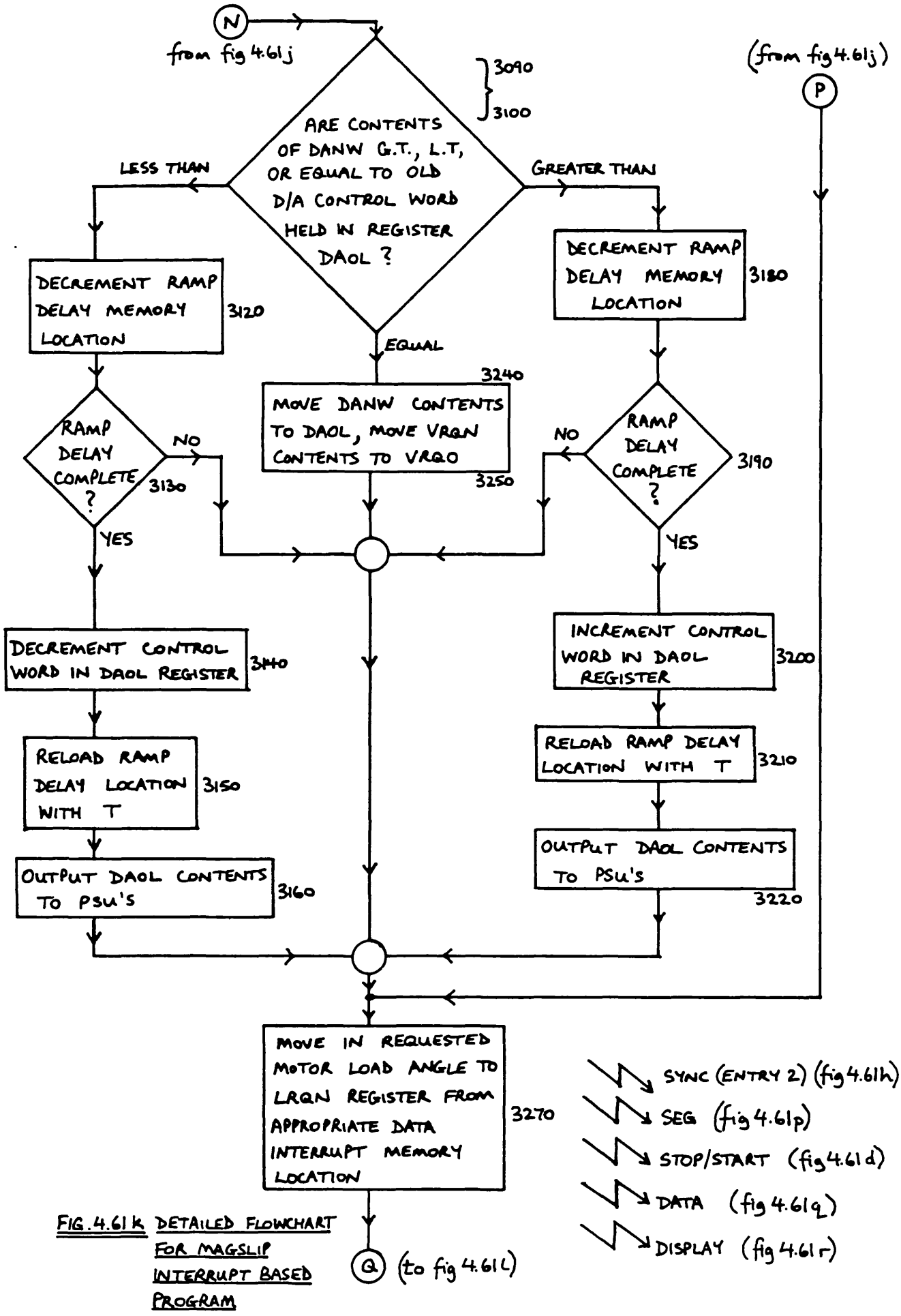


FIG. 4.61k DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM

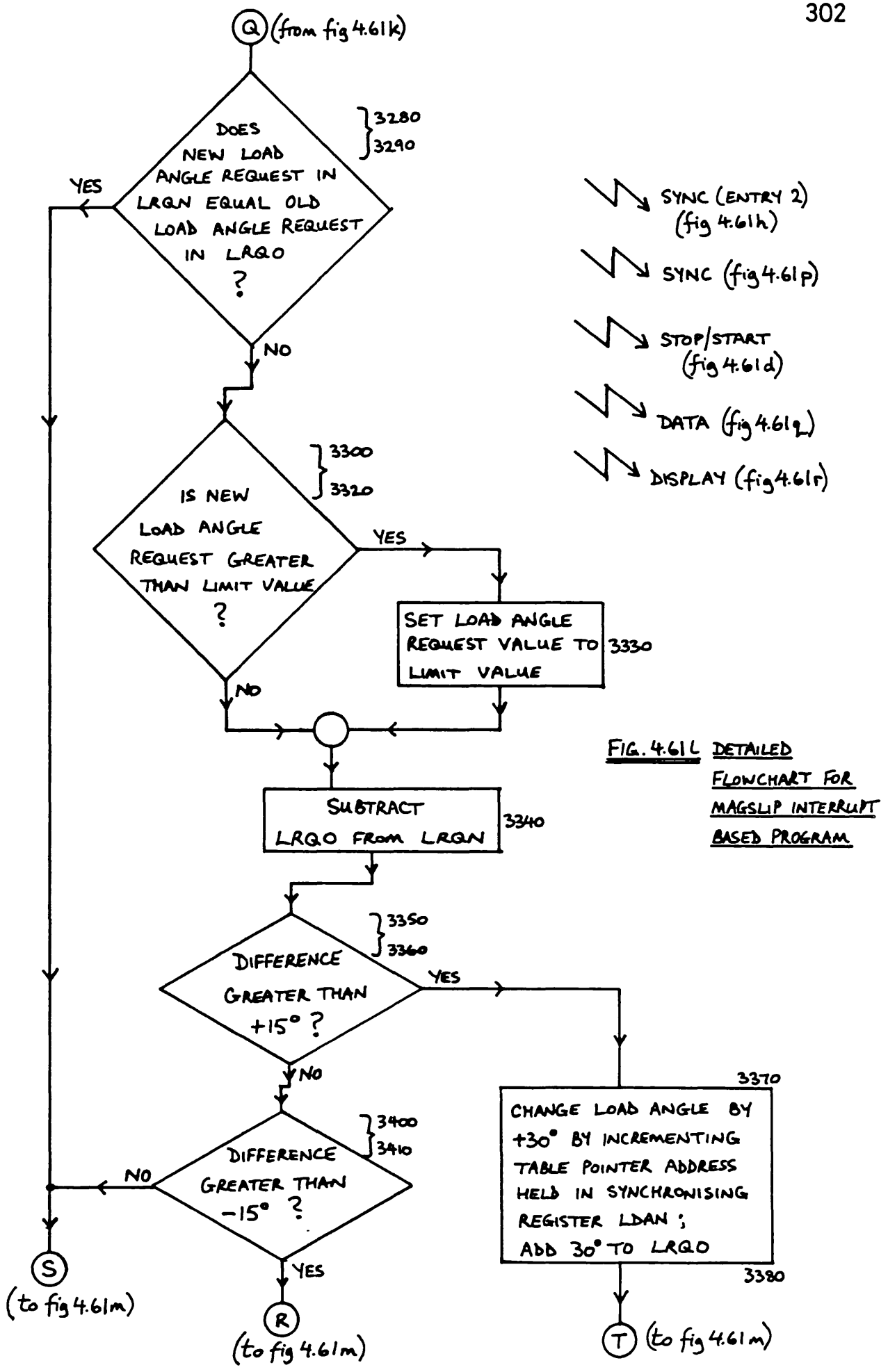


FIG. 4.61L DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM

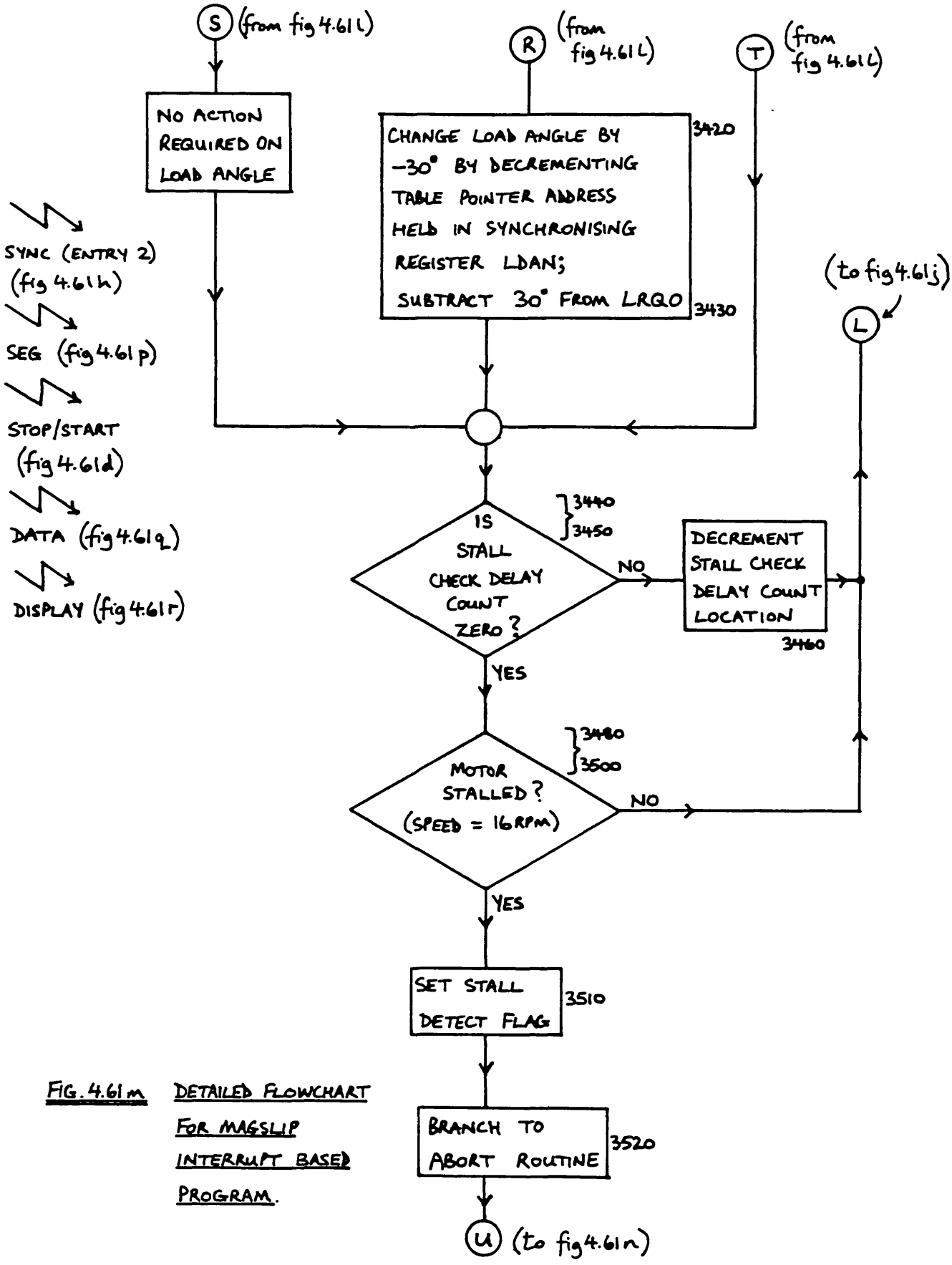


FIG. 4.61m DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM.

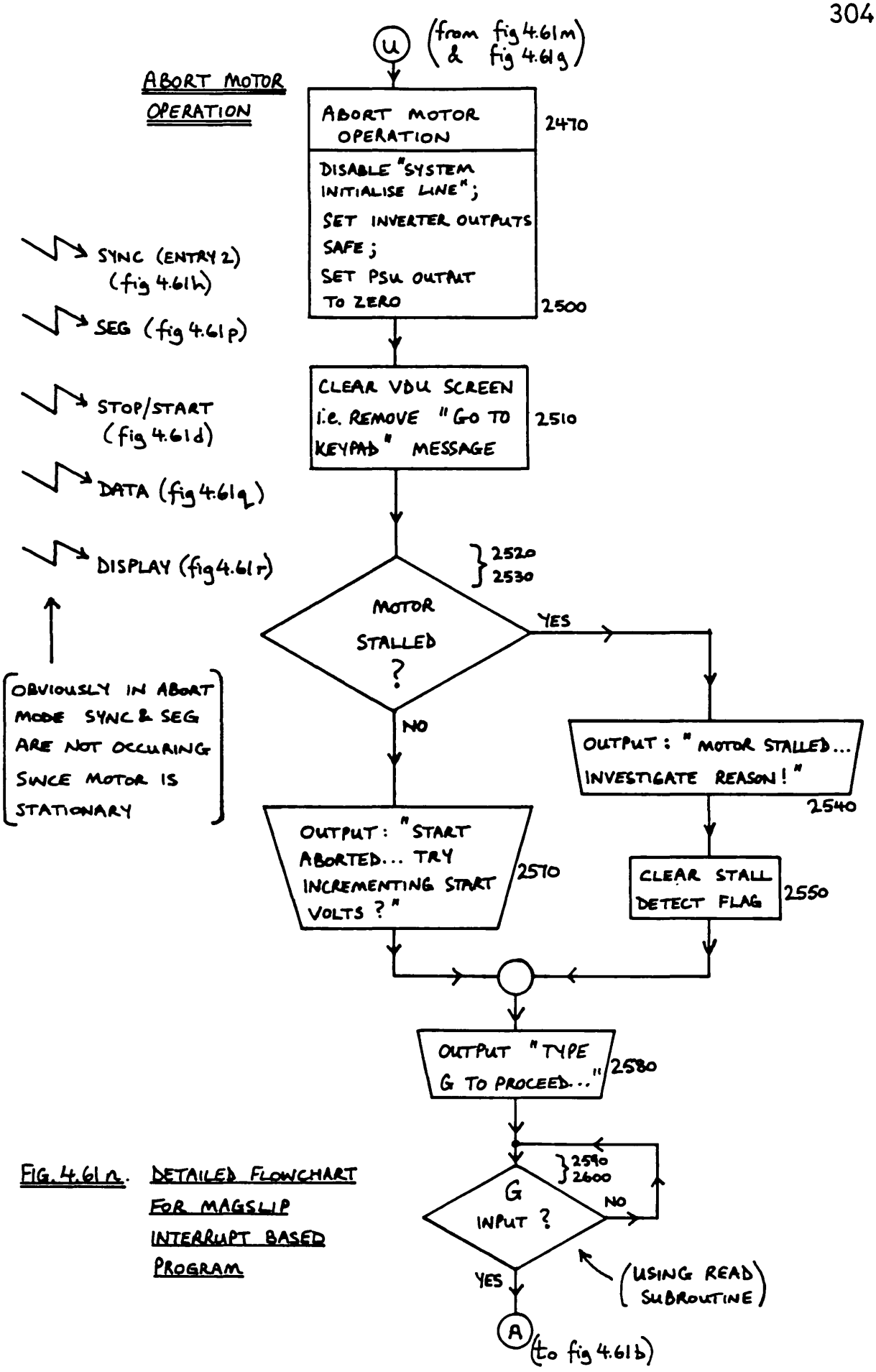


FIG. 4.61n. DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM

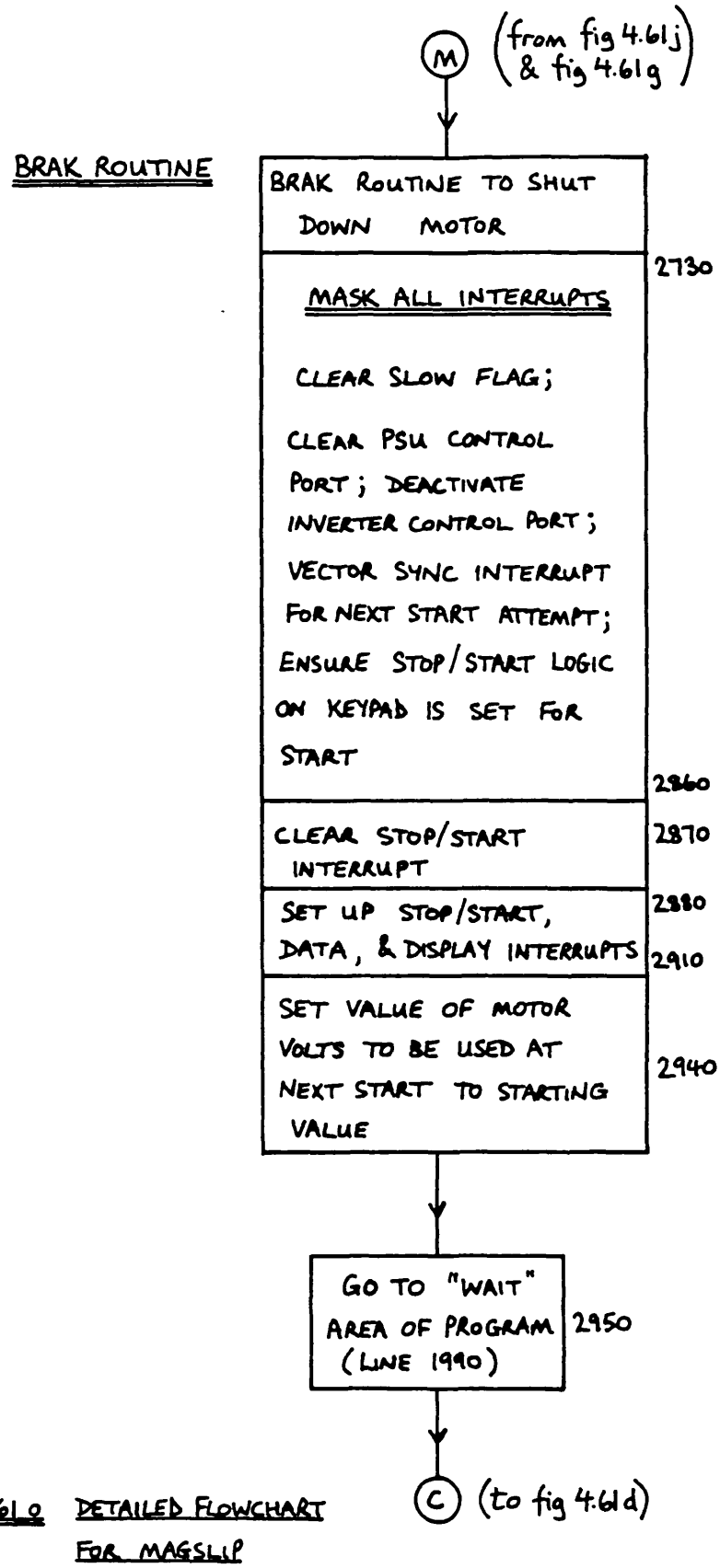


FIG. 4.61 o DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM

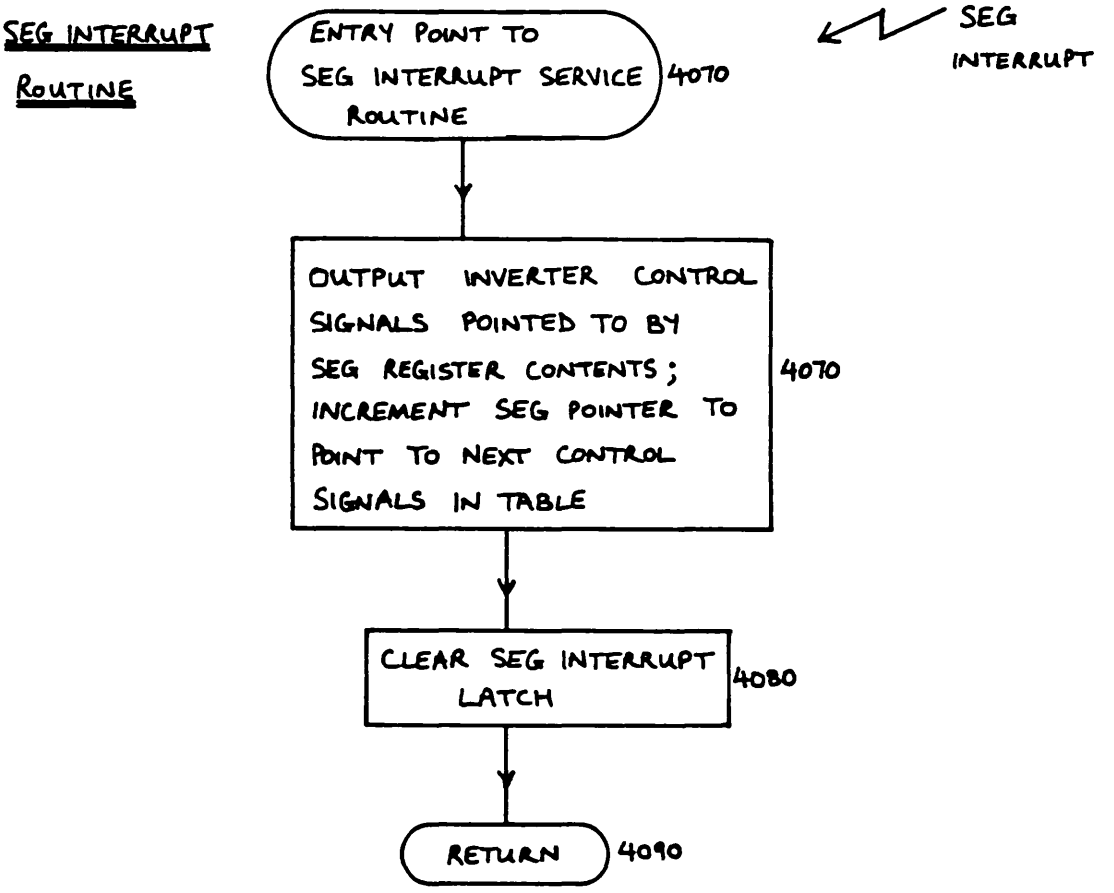


FIG. 4.61P DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM

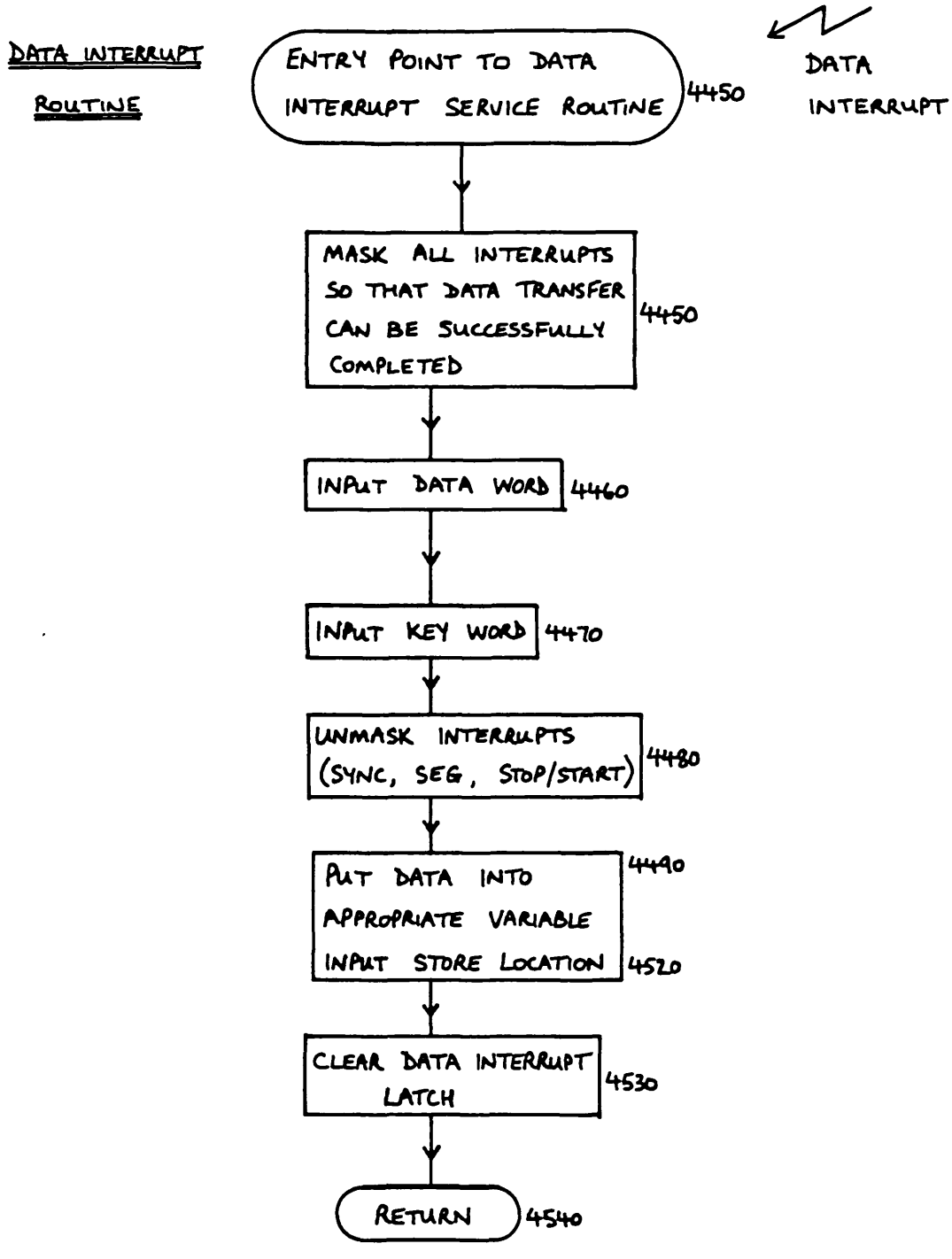


FIG. 4.61g. DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM

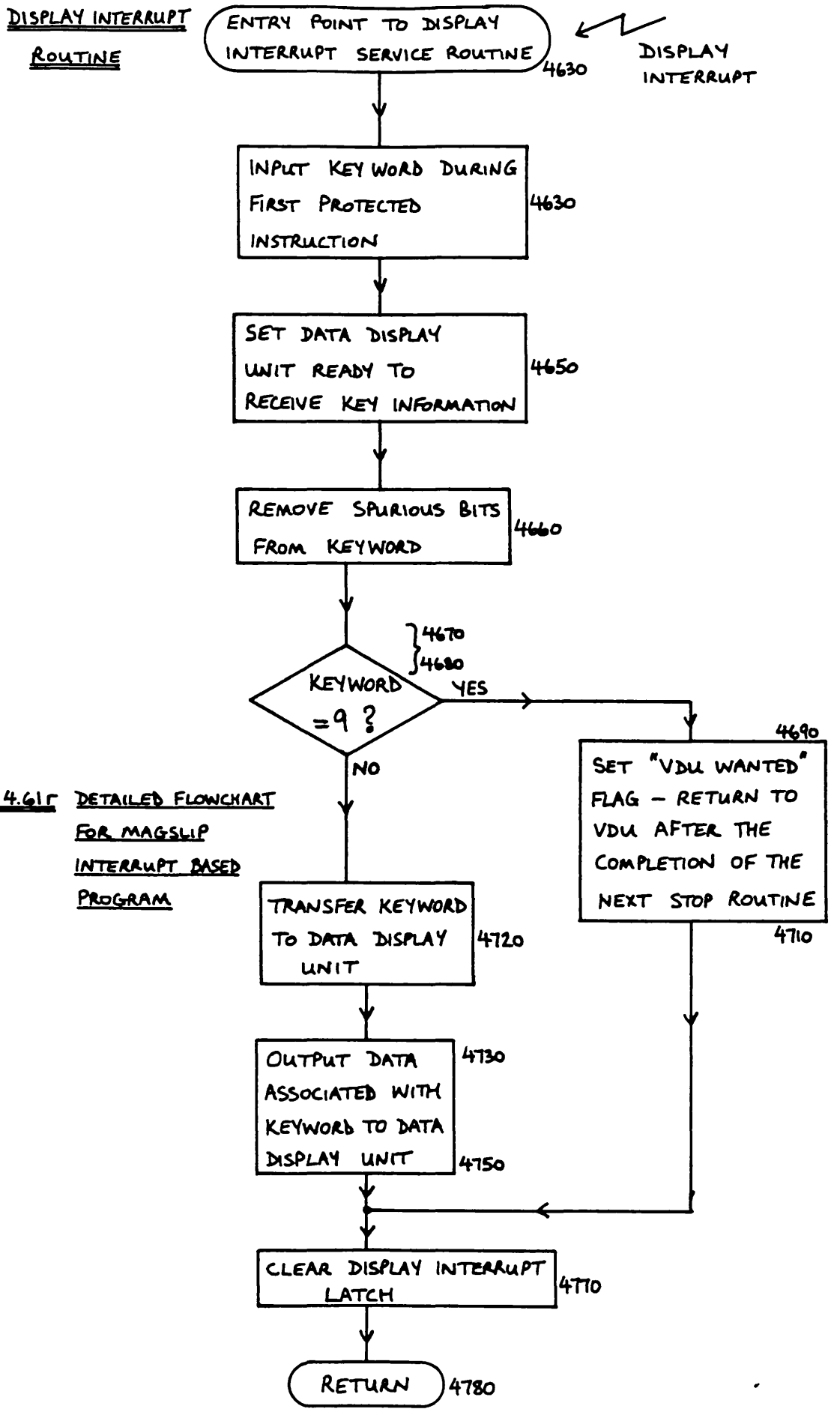


FIG 4.61r DETAILED FLOWCHART FOR MAGSLIP INTERRUPT BASED PROGRAM

at the side of the box.

Details about important sections of the magslip control program, and any special features, are covered in the following sub-sections. All line references in sub-sections 4.8.3.1 to 4.8.3.12 inclusive, refer to the program listing in Appendix 4A.

4.8.3.1 The Introductory Section of the Program

Communication with the microprocessor during the early part of the program is carried out via the VDU display and keyboard. A large introductory title (MICRO MOTOR) and a similarly sized command (GO TO KEYPAD) are displayed on the VDU screen by sub-routine BANN (line 7560 onwards). Only the relevant implementation details of this sub-routine will be discussed since it was developed for the author by a colleague. The sub-routine workspace was defined to begin at memory location 2500_{16}. The sub-routine makes use of the direct cursor control that is available on the Perkin Elmer 550B VDU. By using the appropriate escape code sequences, it is possible to position the cursor at a precise x,y co-ordinate on the screen, and hence large letters can be constructed directly. The sub-routine can be modified to write different messages. Photographs of the two large messages are shown in figs. 4.62 and 4.63.

During the introductory phase of the program, it is possible to change some of the limit values used during the "motor-run" section of the program. In addition, the addresses of the memory locations, which are selectable by the DISPLAY interrupt for output on the data display unit, can be changed. The software to perform the changes in the limit values and data display addresses is implemented by Extended Operation (XOP) sub-routines. Five XOP routines were used called INPU, WRIT, READ, MOD, and HEX. Their workspaces and basic functions are described briefly below.

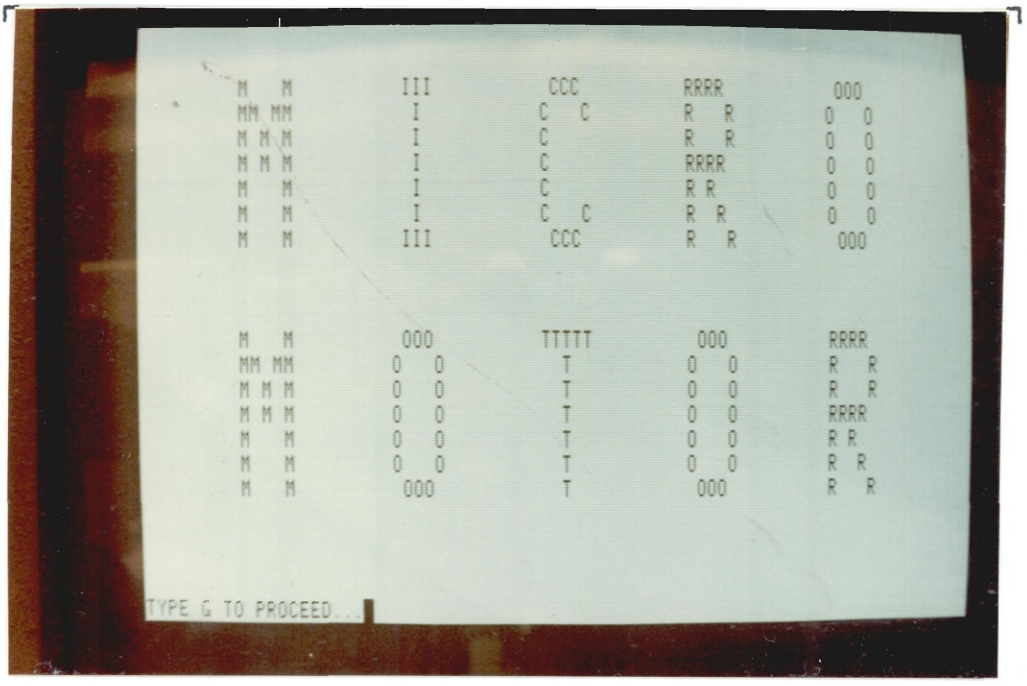


FIG. 4.62. INTRODUCTORY TITLE DISPLAYED AT START OF THE PROGRAM ON THE VDU SCREEN

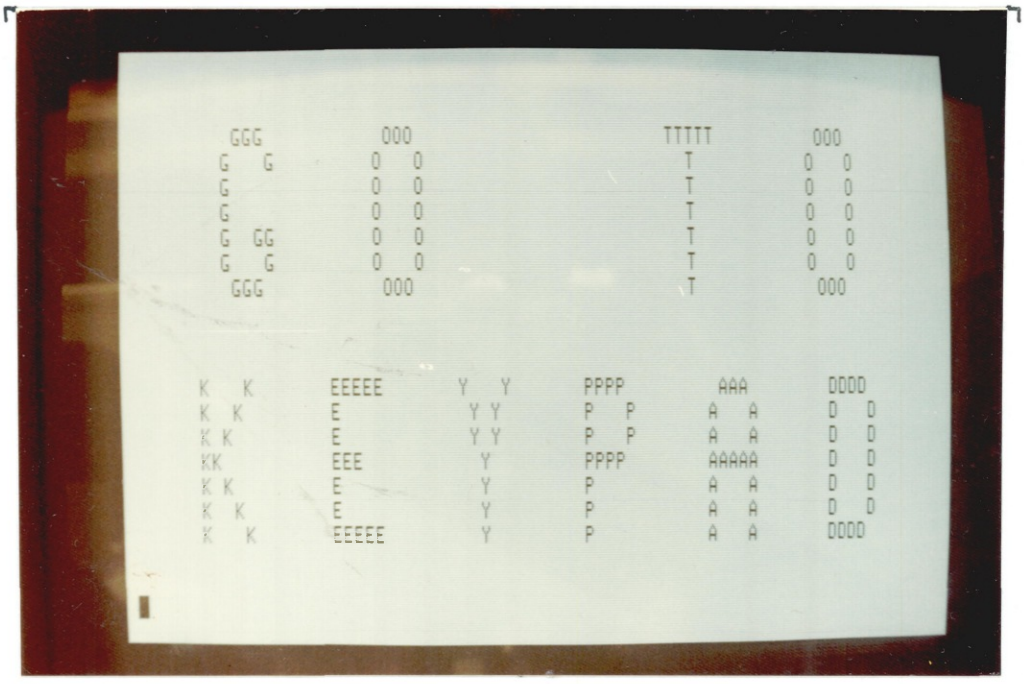


FIG. 4.63. PROGRAM COMMAND TO OPERATOR DISPLAYED ON THE VDU SCREEN

Routine INPU

This was defined as XOP4 with its workspace set at $25A0_{16}$. The routine INPU is located at IXOP (line 5020 to 5080). Routine INPU waits for a character to be received in the TMS9902 input buffer from the VDU. It then transfers the character into register 0 of the calling routine's workspace.

Routine WRIT

This was defined as XOP6 with its workspace set at 2540_{16}. The routine WRIT is located at WXOP (line 5110 to 5130). Routine WRIT outputs a string of characters to the VDU. It employs the software development board (TM990/302-1) output sub-routine located at memory address $E01C_{16}$.

Routine READ

This was defined as XOP5 with its workspace set at 2520_{16}. The routine READ is located at RXOP (line 4830 to 4990). Routine READ inputs a character from the VDU using routine INPU. It then compares the character against a list of expected characters; if a match is found the character is echoed back to the VDU using a software development board routine located at $E018_{16}$. If no match is found, the bell of the VDU is operated, and the routine waits for another character to be entered. The rejection of any irrelevant characters reduces the chance of incorrect data entry. The address of the "permitted" character matching the input character is stored at location RDAT, thus allowing other routines to access the input information.

Routine MOD

This was defined as XOP0 with its workspace set at 2560_{16}. The routine MOD is located at MXOP (line 5160 to 5930). Routine MOD is used to inspect and change the values

of constants V, B, D, S, and T (maximum applied motor voltage, starting voltage, maximum load angle, maximum speed, and timing delay-loop count value respectively) stored at location VARL (line 7020). It makes use of routine WRIT and the software development board sub-routines at $E004_{16}$, $E00C_{16}$, and $E020_{16}$. Routine MOD determines which constant is to be modified by examining the character pointed to by the address at RDAT (previously set up by routine READ). The routine then displays the current value of the selected variable and waits for either a new value or an alternative constant symbol to be entered. If data is entered which is non numerical or out of range the routine outputs "WHAT?", redisplays the old value, and waits for a new entry. Escape from the routine is achieved by entering an "E". The photograph shown in fig. 4.64 illustrates how routine MOD operates. The photograph shows that the value of B has been inspected and after an initial data entry mistake, its value has been changed from 10 to 20. The value of S has then been inspected and the final action is to escape from the routine by typing E.

Routine HEX

This was defined as XOP1 with its workspace set at 2580_{16}. The routine HEX is located at HXOP (line 5960 to 6560). Routine HEX is used to inspect and change the values of the recall memory addresses used by the DISPLAY interrupt service routine. It makes use of the READ and WRIT routines as well as the software development board sub-routine at $E010_{16}$. Ten hexadecimal memory addresses (numbered 0 to 9) stored at location MO (line 7230) can be inspected and modified by routine HEX. The photograph shown in fig. 4.65 indicates the typical sequence of events when using the routine. A recall memory address value is inspected by entering the appropriate recall memory key number (0 to 9). The photograph shows that memory recall key number 1 is associated with memory address $100C_{16}$. No change in this address is required and so carriage return (CR) is entered;

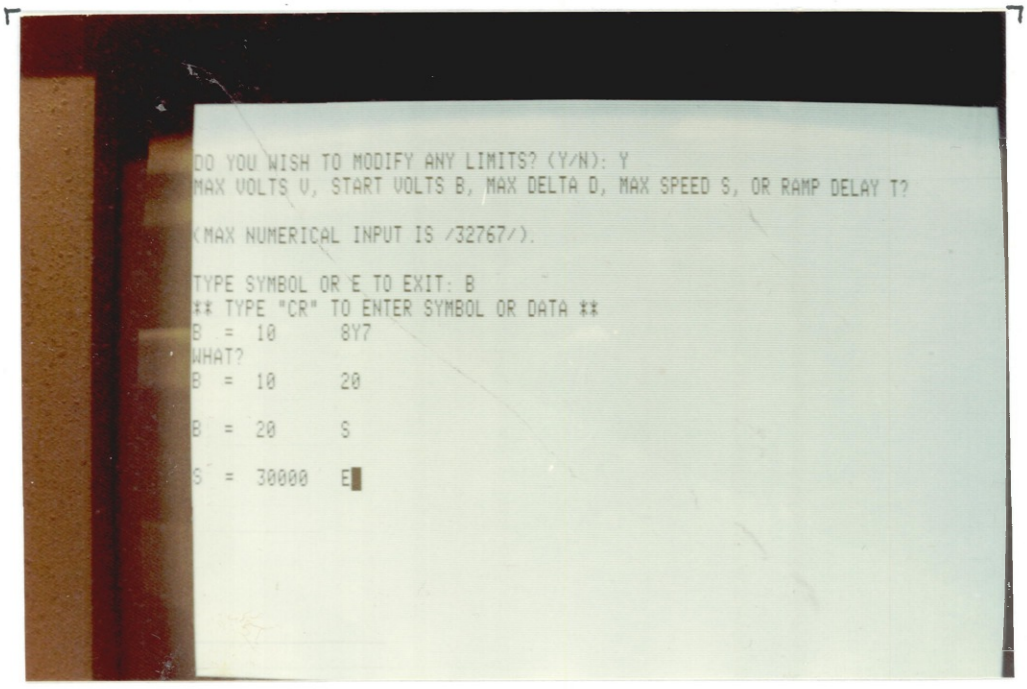


FIG. 4.64. EXAMPLE OF THE OPERATION OF ROUTINE MOD

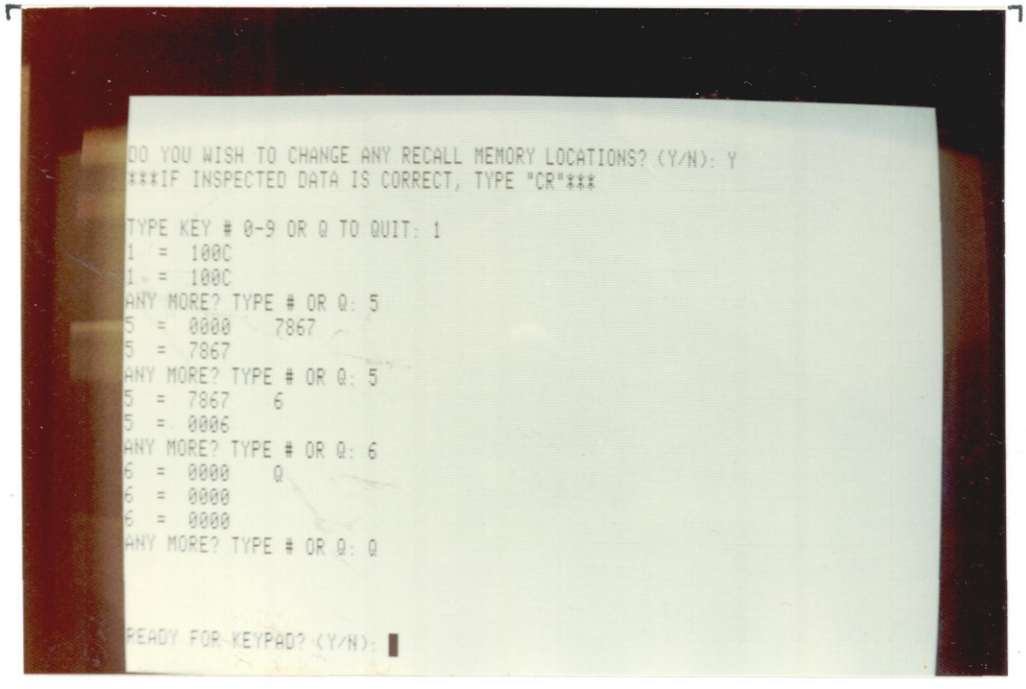


FIG. 4.65. EXAMPLE OF THE OPERATION OF ROUTINE HEX

this causes the address value to be redisplayed and the routine asks if any further changes are required. Escape from the routine is achieved by typing Q, but in the example shown the number 5 is entered. Recall memory address 5 is then changed from 0000_{16} to 7867_{16}. The example goes on to show a further change to recall memory address 5, and also illustrates how the escape command (Q) is ignored in the middle of an inspection sequence; the routine merely responds by redisplaying the recall memory address (in the example "6 = 0000"). The routine is finally brought to an end by entering Q, and the photograph shows that the microprocessors' response is to ask if the operator is ready for keypad operation. It should be noted that routine HEX only retains the last four digits entered when a modification is made. In addition if a character other than 0 to 9, A to F, or Q is entered, the routine rings the VDU bell and waits for a valid character to be typed in.

Once control has been passed to the keypad, it is possible to reestablish communication between the VDU and the microprocessor in two ways under program control.

(a) By way of the ABORT routine. If the motor fails to start in the allotted number of steps, or it stalls whilst running, the program shuts down the system and issues an appropriate message on the VDU. The operator then proceeds, if desired, through the introductory parts of the program via the VDU.

(b) By request, using the DISPLAY interrupt in conjunction with recall memory location 9. When PRINT 9 is entered on the keypad, no data is presented on the data display unit. Instead, a flag is set which directs the microprocessor back to the introductory part of the program when the motor next comes to a halt.

4.8.3.2 Start Routine of the Magslip

The microprocessor enters the start routine (lines 2120 to 2440) after detecting a start instruction via the STOP/START interrupt service routine. The workspace of the start routine was set at 1000_{16}. The main function of the start routine is to sequence the inverter through its set of output states, at a rate sufficiently slow for the magslip rotor to follow, until either a SYNC interrupt is received or the start attempt is aborted.

It is worthwhile illustrating schematically how the angular orientation of the resultant stator flux is related to the inverter control signals. Each control word for the magslip inverter is formed from the six least significant bits (D10 to D15 inclusive) of the microprocessor data bus, as explained in sub-section 4.7.8. Bits D0 to D9 of the data bus are not required to control the magslip inverter and so they are set to logic 0 (arbitrarily). The sequence of hexadecimal control words that produces continuous rotation of the magslip in one direction is shown in fig. 4.66. For each hexadecimal inverter control word, fig. 4.66 shows the corresponding inverter control signal states. In addition, fig. 4.66 indicates both the appropriate voltage polarity applied to each phase winding terminal, and the angular position of the resultant stator flux with reference to the axes of the phase windings. Fig. 4.67 shows how the voltage waveforms at the phase winding terminals are related to the hexadecimal inverter control words. The tables of inverter control words for forward and reverse motor operation begin respectively at line 3930 (table DRVF) and line 4000 (table DRVR) of the program listing in Appendix 4A.

One point which merits some discussion, is the method of preventing the microprocessor from "running off" the end of the table containing the inverter control words during a start sequence. The most significant bit (D0) of the microprocessor data bus is not used in the magslip inverter

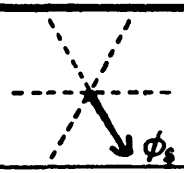
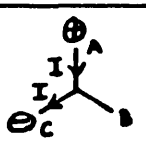
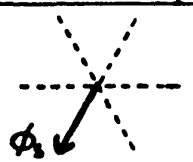
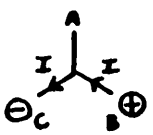
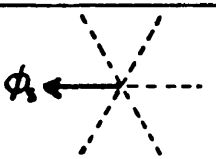
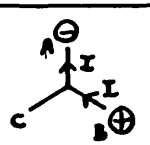
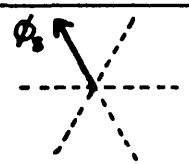
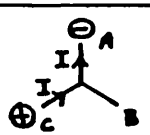
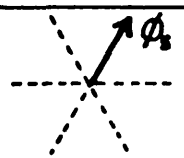
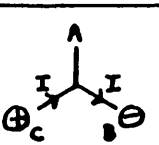
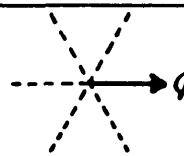
INVERTER CONTROL WORD	BITS D10 TO D15 OF INVERTER CONTROL WORD						INVERTER CONTROL SIGNALS DERIVED FROM D10 TO D15						POLARITY OF APPLIED PHASE VOLTAGES: RESULTANT CURRENTS	RESULTANT STATOR FLUX ORIENTATION
	D10	D11	D12	D13	D14	D15	A1	B1	C1	A2	B2	C2		
$001D_{16}$	0	1	1	1	0	1	1	0	0	0	1	0		
$001E_{16}$	0	1	1	1	1	0	1	0	0	0	0	1		
$002E_{16}$	1	0	1	1	1	0	0	1	0	0	0	1		
$002B_{16}$	1	0	1	0	1	1	0	1	0	1	0	0		
0033_{16}	1	1	0	0	1	1	0	0	1	1	0	0		
0035_{16}	1	1	0	1	0	1	0	0	1	0	1	0		

FIG. 4.66. RELATIONSHIP BETWEEN THE MAGSLIP INVERTER CONTROL WORDS AND THE STATOR FLUX ORIENTATION

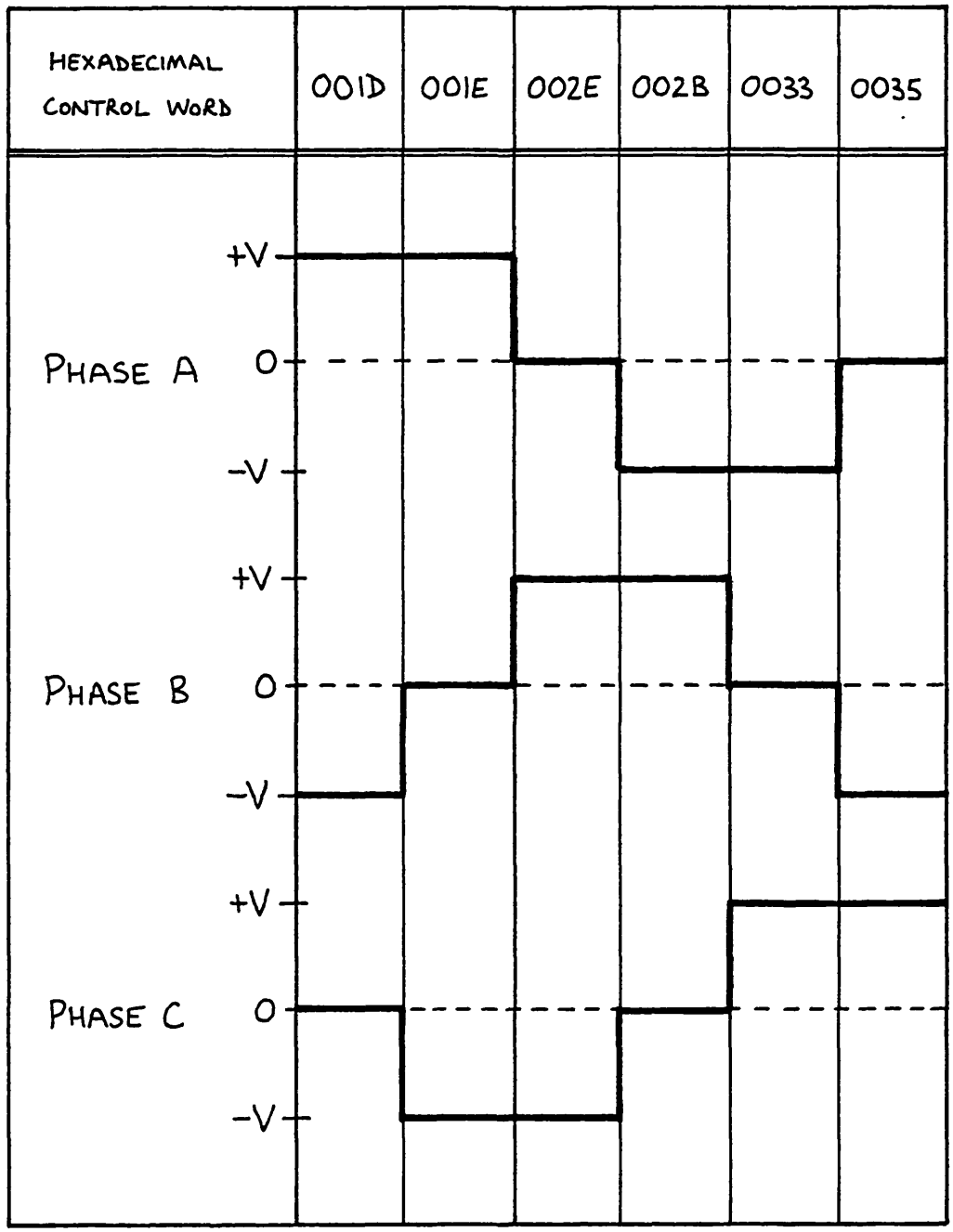


FIG. 4.67. RELATIONSHIP BETWEEN THE MAGSLIP APPLIED PHASE VOLTAGE WAVEFORMS AND THE INVERTER CONTROL WORDS

control word (nor is it needed in the control word of the 7 phase MOSFET inverter: hence what follows also applies to the 7 phase system start software). It is therefore possible to use this bit as a signal to indicate that the end of the table is approaching. The signal can be sensed either as an interrupt or as a flag.

Initially it was planned to sense the "end of table approaching" condition by using the OVFL interrupt (int4) which was hardwired to the most significant bit of the inverter control output port. However, it was realised that the task did not really warrant the use of an interrupt and so it was decided to use the signal as a flag. The simplest way of sensing the status of the flag is to test if the inverter control word that has just been output to the inverter is negative (i.e. most significant bit D0 at logic 1). If the control word is found to be negative the table pointer can be reset accordingly (see lines 2390 to 2440).

4.8.3.3 Comments on the "Main Body" of the Program

The workspace of the "main body" of the program (lines 2970 to 3520) is the same as that of the start routine. The main function of this section of the program is to execute any changes in either the applied motor phase voltages or load angle which may be requested manually via the keypad. The input variables can have integer values between 0 and 65535_{10} but the software truncates any input which exceeds the preset limits. The DATA interrupt service routine inputs new data and stores it in the appropriate input variable memory location at VARI (line 4590). The microprocessor continuously cycles through the "main body" of the program whilst the motor is running and on each loop the appropriate variables (volts and load angle) are transferred from VARI regardless of whether they have changed or not. The "new" values of demanded motor voltage and load angle are placed into registers VRQN and LRQN respectively. The old values of motor voltage and load angle are held in

registers VRQO and LRQO respectively. By comparing VRQN with VRQO the program decides if any change in voltage has been demanded. Similarly, a comparison between LRQN and LRQO determines whether a load angle change is required. Both the contents of VRQN and LRQN are checked against preset limit values; if the contents of a given register are over the relevant limit they are replaced by that limit.

If a change in load angle is required it is then achieved by incrementing or decrementing the table pointer address held in register LDAN; this address is used in the SYNC interrupt service routine to synchronise the pointer address of the inverter control signal table to the position of the rotor.

The selected voltage is actually implemented by two further registers. The 0 to ±50 volt magslip power supply is controlled by an eight bit binary number via the D/A control circuits housed in the keypad auxiliary logic unit. The new voltage value is used to select the appropriate D/A control byte from look-up table VTAB (line 3600). This new byte is stored in register DANW whilst the old control byte is stored in register DAOL. A gradual change in voltage is then achieved by arranging the value held in DAOL to increment (or decrement) towards the value held in DANW; the process is deliberately slowed down by the delays produced by decrementing the memory locations @SLOW+4 and @SLOW+6 (line 3550). The former memory location is used for delays during decrementing whilst the latter is used during incrementing. They are both loaded with a value of 200_{10} (constant T) and during a voltage change sequence the appropriate memory location is decremented towards zero. Two hundred passes through the "main body" are required to achieve this and at the end register DAOL is decremented or incremented by one. The delay is then repeated until the contents of DAOL equal those of DANW. By altering T it is possible to change the rate at which the motor applied voltage ramps up and down.

The only values of load angle available to the operator are 0°, 30°, 60°, and 90°. If the load angle requested is not equal to one of the available angles the program selects the closest available angle.

Another important function performed by the "main body" of the program is the check to determine if a STOP/START interrupt has requested the microprocessor to stop the motor. The microprocessor tests the "SLOW" flag on each loop of the program and if it has the appropriate status a run down procedure is initiated.

The main body of the program also performs one useful health check on the motor system. During each program loop the motor speed is tested to determine if it has stalled. A zero speed condition in the system is assumed whenever the digital tachometer output is 16 rpm, this being the minimum reading of the tachometer. When zero speed is detected, the system is shut down to prevent possible excessive motor currents. A simple delay loop prevents erroneous "zero speed" detection when the motor is just starting; the delay allows sufficient time for the tachometer reading to rise from 16 rpm before a stall test is made.

Further tasks that might be executed by the "main body" of the program would include speed control and automatic optimum load angle selection, but in the magslip and 7 phase motor systems they were not implemented due to lack of time.

4.8.3.4 The TMS9900 Instructions Used on the Memory Mapped I/O Ports

A variety of TMS9900 instructions can be used directly on the memory mapped input and output ports. The instructions used in the motor control program were CLR, SETO, and MOV: MOV is used to transfer data both from memory to an output port and from an input port to memory; CLR is used to clear the contents of output ports; SETO is used to set

all the bits of an output port to logic 1.

In addition CLR is used to activate input port enable line $C006_{16}$. This line is used by the "Data Display" output port $C012_{16}$ (see sub-section 4.7.4.12) to set up the key code/data transfer order. It might be expected that the CLR instruction would only activate output port enable lines since the "clear" operation basically involves writing all zeroes to the chosen memory location. However, the software architecture of the TMS9900 is such that CLR (and SETO) are grouped with instructions INV, NEG, INC, INCT, DEC, DECT, and SWPB (see page 4-95 of reference 4.11). The machine cycles of these instructions involve a memory read from the addressed memory location and hence the DBIN control line is activated. Therefore, the CLR instruction can be used to activate an input port enable line.

4.8.3.5 The Chosen TMS9900 Instruction for the Resetting of the "Memory Mapped" Interrupt Latches

The choice of a suitable TMS9900 instruction to reset the six available "memory mapped" interrupt latches was made very carefully to ensure that the reset operation should be as quick as possible. Several "fast" instructions were available including CLR, SETO, INV, INC, INCT, DEC, DECT, and SWPB, all of which take 7.326μs to execute when accessing a symbolic memory address. It was thought that CLR was the most appropriate instruction to use since it had some meaning in the context of "clearing" an interrupt latch. Hence CLR was used in the motor control program to reset the interrupt latches dealing with the STOP/START, DATA, and DISPLAY interrupts.

4.8.3.6 The Relationship between the Workspace of the "Main Body" Routine and those of the SYNC, SEG, and STOP/START Interrupt Service Routines

The program is structured so that the "main body" rou-

tine and the SYNC, SEG, and STOP/START interrupt routines are all able to operate on several registers that are common to all their workspaces. This minimises the data linkage problems that are inherent in a microprocessor using the workspace-in-memory approach for its registers. Overlapping the workspaces as shown in fig. 4.68 removes the need to transfer data between routines because it is present in all of them. However, there are two problems associated with overlapped interrupt service routine workspaces. Firstly, the "common" registers have different register numbers in each routine workspace. Secondly, sufficient registers must be left available in the higher priority interrupt service routine workspaces to store the return-from-interrupt vector information, which can accumulate in the event of a sequence of increasing priority interrupts occurring in swift succession. For example, if the STOP/START interrupt service routine is itself interrupted by the SEG interrupt, then the return-from-interrupt information for the STOP/START routine (WP_8, PC_8, ST_8) is "stored" automatically in registers 7, 8, and 9 of the SEG interrupt service routine workspace. (Note in fig. 4.68 that WP_n, PC_n, and ST_n with n = 1, 2, and 8 are respectively the workspace pointer, program counter, and status register values to be returned to at the completion of TMS9900 interrupt priority n.)

The STOP/START interrupt service routine workspace is actually the same as that used by the "main body" routine; hence registers 13, 14, and 15 of the "main body" workspace have to be reserved for the STOP/START return-from-interrupt data (WP_8, PC_8, ST_8).

4.8.3.7 The Autopiloting Interrupt Service Routines

The SYNC and SEG interrupt service routines perform the necessary autopiloting functions. The SEG routine is executed much more frequently than any other routine when the motor is running and so it limits the maximum motor

		9900 INT8		9900 INT2	9900 INT 1
MEMORY ADDRESS	"MAIN-BODY" ROUTINE WORKSPACE	STOP/START ROUTINE WORKSPACE	SPARE	SEG ROUTINE WORKSPACE	SYNC ROUTINE WORKSPACE
1000_{16}	RØ	RØ			
1002_{16}	R1 VRQN	R1			
1004_{16}	R2 VRQO	R2			
1006_{16}	R3 DANW	R3	SAVED FOR FUTURE USE		
1008_{16}	R4 DAOL	R4			
$100A_{16}$	R5 LRQN	R5			
$100C_{16}$	R6 LRQO	R6		RØ	
$100E_{16}$	R7	R7		R1	
1010_{16}	R8	R8		R2	
1012_{16}	R9	R9		R3	RØ
1014_{16}	R10 LDAN	R10		R4	R1 LDN1
1016_{16}	R11 SEG	R11		R5 SEG2	R2 SEG1
1018_{16}	R12 CRUB	R12 CRUB		R6	R3
$101A_{16}$	R13	R13 WP_8		R7 (WP_8)	R4 (WP_8)
$101C_{16}$	R14	R14 PC_8		R8 (PC_8)	R5 (PC_8)
$101E_{16}$	R15	R15 ST_8		R9 (ST_8)	R6 (ST_8)
1020_{16}				R10	R7
1022_{16}				R11	R8
1024_{16}				R12	R9
1026_{16}				R13 WP_2	R10 (WP_2)
1028_{16}				R14 PC_2	R11 (PC_2)
$102A_{16}$				R15 ST_2	R12 (ST_2)
$102C_{16}$					R13 WP_1
$102E_{16}$					R14 PC_1
1030_{16}					R15 ST_1

FIG.4.68 THE OVERLAPPED WORKSPACES ARRANGEMENT

speed. It is vital to minimise the number of instructions in the SEG routine if a high motor speed is desired. The instructions making up the routine and their execution times are:

see lines 4070 to 4090 Appendix 4A	Context Switch to SEG Interrupt	8.991μs
	MOV *SEG2+,@>C00C	12.321μs
	CKOF	4.329μs
	RTWP	5.994μs

The total execution time of the SEG interrupt routine is therefore 31.635μs (32μs say). Hence the absolute maximum number of SEG interrupts per second that could be serviced is 31250. (This would not be actually possible in practice since it would leave no time for any other functions to be performed). It is interesting to note that if the MOV instruction is removed so that no useful function is performed, the interrupt service routine would still take 19.314μs to complete; hence the useful part of the routine takes a minor portion of the total execution time, the major portion being taken by the unavoidable interrupt handling procedures.

The main action of the routine is achieved by the MOV instruction. The address of the next inverter control word is held in register SEG2; when MOV is executed the control word is transferred to the output port at address $C00C_{16}$ and the address held in SEG2 is incremented to point to the next inverter control word. The sequence of inverter control words are held in a table, as mentioned earlier in sub-section 4.8.3.2: table DRVF (line 3930) contains the correct sequence for forward motor operation; table DRVR (line 4000) the correct sequence for reverse motor operation. The signals in the control words are "active low" for the reason discussed in the hardware section of this chapter.

Once every revolution the SYNC interrupt service rou-

tine is entered. The entry point when the motor first synchronises is SYST (line 3770) but during subsequent operation the entry point is SYRN (line 3870). The instructions forming the "heart" of the routine and their execution times are:

see lines 3870 to 3910 Appendix 4A

Context Switch to SYNC Interrupt	8.991µs
MOV LDN1,SEG1	5.994µs
CKON	4.329µs
MOV *SEG1+,@>C00C	12.321µs
CKOF	4.329µs
RTWP	5.994µs

The total execution time of the SYNC interrupt routine shown above is 41.958µs (42µs). The SYNC routine performs two functions. Firstly, it sets up the required load angle. The load angle is determined by the address set up in register LDAN of the main body routine workspace. This register is called LDN1 in the SYNC routine and its contents are transferred into the inverter-control-table-pointer-register SEG1 whenever SYNC is executed, thus establishing the relationship between the inverter outputs and the rotor position for the next revolution. Secondly, the SYNC routine services the SEG interrupt that occurs simultaneously with the SYNC interrupt. The total effect of the SYNC routine is therefore to synchronise the inverter-control-table-pointer and then to output the next inverter control word. The SYNC and SEG interrupt latches are both cleared before the microprocessor leaves the routine. Servicing the SEG interrupt in the SYNC routine saves time: if it was not done in this way the microprocessor would exit from the SYNC interrupt routine (RTWP = 5.994µs) and would then be immediately vectored to deal with the pending SEG interrupt (context switch = 8.991µs). This would result in an extra 14.985µs of execution time whenever simultaneous SYNC/SEG interrupts

occurred, leading to a slight reduction in the maximum interrupt rate that could be handled, and a proportionate reduction in maximum possible motor speed.

The magslip system produces 12 SEG pulses per revolution and so at its maximum speed of 12000 rpm there is a minimum interval between SEG interrupts of 417μs. The microprocessor therefore has no difficulty in servicing the SEG interrupts.

4.8.3.8 Execution Time of the STOP/START Interrupt Service Routine

The execution time of the STOP/START interrupt service routine depends on whether a stop or start command is sensed at the beginning of the routine. If a start request occurs the sequence of instructions and execution times is:

	Context switch into STOP/START interrupt routine		8.991μs
	TB	10	4.662μs
	JNE	STRT	3.663μs
	↓		
STRT	LIMI	0	5.994μs
	CKON		4.329μs
	CKOF		4.329μs
	LI	CRUB,>100	4.995μs
	LI	R8,>702	4.995μs
	LDCR	R8,11	14.980μs
	LI	CRUB,>120	4.995μs
	LI	R14,AWAY	4.995μs
	LI	R15,10	4.995μs
	CLR	@>C01A	7.326μs
	RTWP		5.994μs

The total execution time for the start portion of the STOP/START routine is therefore 85.243μs. The arrangement of the start routine is such that it cannot be interrupted by

any system interrupt (other than INT0 reset) during its execution, and so its processing time is always the same.

If a stop request occurs the sequence of instructions and execution times is:

Context switch into STOP/START interrupt routine		8.991µs
TB	10	4.662µs
JNE	STRT	2.997µs
MOV	@SLOW+2,@SLOW	11.988µs
CLR	@VARI+2	7.326µs
LI	CRUB,>100	4.995µs
SBZ	8	4.662µs
LI	CRUB,>100	4.995µs
RTWP		5.994µs

The total execution time for the stop portion of the STOP/START routine is therefore 56.61µs. This time assumes that the routine is not interrupted during its execution by a SYNC or SEG interrupt; if this occurs the time for the STOP/START routine to be completed is obviously extended. The stop routine does not in itself stop the motor, since it basically instructs the "main body" routine to ramp down the motor supply voltage and then disable the inverter and power supply. Therefore, the actual process of stopping the motor lasts for several seconds, even though it is initiated by a routine lasting as little as 56.61µs.

4.8.3.9 The DATA Interrupt Service Routine

The DATA interrupt service routine has its workspace set at $25E0_{16}$. The sequence of instructions and their execution times are:

Context switch to DATA interrupt routine	8.991µs

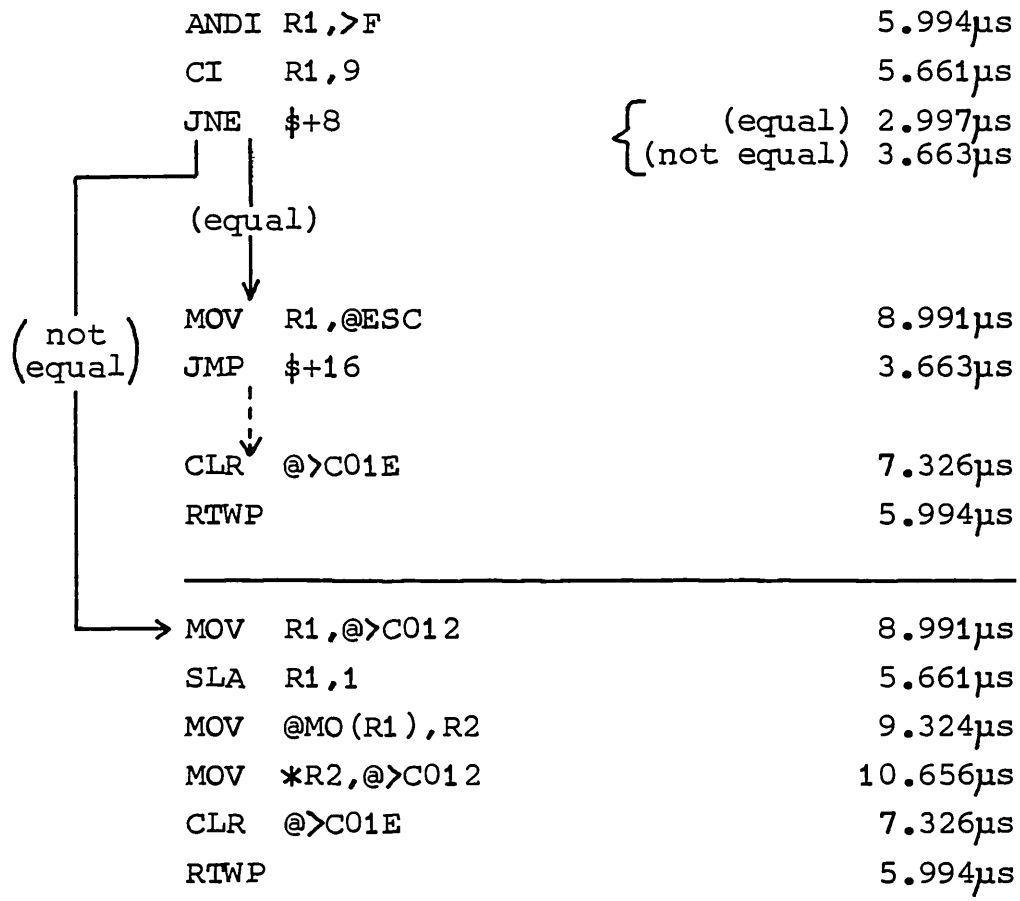
LIMI 0	5.994μs
MOV @>C002,R0	8.991μs
MOV @>C004,R1	8.991μs
LIMI 8	5.994μs
ANDI R1,>7	5.994μs
DECT R1	4.329μs
SLA R1,1	5.661μs
MOV R0,@VARI(R1)	9.324μs
CLR @>C01C	7.326μs
RTWP	5.994μs

The execution time for the DATA interrupt routine (assuming it is not interrupted by STOP/START, SEG, or SYNC) is 77.589μs. The first instruction (LIMI 0) sets the interrupt mask such that the following two MOV instructions can execute without interruption, thus ensuring that the data and keyword are successfully transferred from the keypad unit. The six variables that the keypad can alter are stored from memory location VARI to VARI+10 (see line 4590). The motor control program only uses two of these locations: the requested applied motor phase voltage is stored in VARI and the requested load angle is stored in VARI+2. These two quantities are entered by the operator via the keypad unit by selecting the "a" and "b" variable states respectively. The main body routine moves the contents of VARI and VARI+2 into registers VRQN and LRQN respectively each time the microprocessor passes through it.

4.8.3.10 The DISPLAY Interrupt Service Routine

The DISPLAY interrupt service routine has its workspace set at 2600_{16}. The sequence of instructions and their execution times are:

Context switch to DISPLAY interrupt routine	8.991μs
MOV @>C004,R1	8.991μs
CLR @>C006	7.326μs



Assuming the routine is not interrupted by DATA, STOP/START, SEG, or SYNC, the execution times for the two paths through the DISPLAY interrupt routine are 65.934 (the "equal" path) and 88.578µs (the "not equal" path).

A total of fifteen memory locations can be recalled via the DISPLAY interrupt for display on the data display unit. Nine of the locations can be changed as required during the introductory part of the motor control program. The addresses of these locations are stored from location M0 to M0+10_{16} (line 7230). They are accessed via the keypad by pressing PRINT n (n=0 to 8) and the ENTER key. Three locations (M0, M0+2, M0+4) were preset: location M0 holds the address of register VRQO; location M0+2 holds the address of register LRQO; and location M0+4 holds the address of the tachometer input port. Hence the use of PRINT 0, PRINT 1, and PRINT 2 causes the display of the current working inverter supply voltage, the current load angle, and the current speed, respectively.

The six memory locations (line 7240) following location $M0+12_{16}$ contain the addresses of memory locations VARI to $VARI+A_{16}$ (line 4590). The addresses stored by locations $M0+14_{16}$ to $M0+1E_{16}$ are set up during the introductory part of the program (see lines 0420 to 0490). The locations from VARI to $VARI+A_{16}$ contain the data entered via the DATA interrupt service routine. Appropriate use of the CONTROL and PRINT keys on the keypad (as described in Chapter 7) permits these locations to be displayed on the data display unit. Hence the last requested value of inverter supply voltage or load angle can be displayed for comparison with the actual working values, which can be observed via PRINT 0 and PRINT 1 as explained above. Therefore, if the program limits any input variables, this can be readily seen.

The PRINT 9 recall-memory-address storage location ($M0+12_{16}$) is not used. PRINT 9 is reserved for use as a flag to indicate that a return to the introductory part of the program is required. Once the flag ESC9 (line 4790) has been set, the microprocessor returns to the introductory part of the program when it next goes through the "wait" section of the program (lines 1990 to 2100).

4.8.3.11 Precautions to Ensure Correct Interrupt Servicing

In both the "start" section of the STOP/START interrupt routine and the "entry point 1" section of the SYNC interrupt routine, the return-from-interrupt program counter address (register 14 of each interrupt routine workspace) is modified. The contents of register 15 (the saved status register) may not then be appropriate for the new section of code that the microprocessor is directed to, and so it is best to clear the status bits (bits 0 to 6) of this register. However, it is necessary to set the interrupt mask section of the register (bits 12 to 15) so that if any lower priority interrupt was actually interrupted by either of the above mentioned interrupts, it can subsequently re-interrupt the microprocessor. Since the point at which such

an interrupt was interrupted is lost, the microprocessor re-executes the whole of the interrupt routine when it returns to it.

It should be noted that to fully ensure correct operation when changing the return-from-interrupt information, it is also generally necessary to set the contents of register 13 to be the workspace pointer of the new portion of program that the microprocessor returns to after the interrupt (see section 8.5 page 8-14 of reference 4.13, which deals with context switches to sections of a program). If this is not done, the microprocessor will work in the "old interrupted" workspace, which could lead to overwriting of data and other vital information. The implemented interrupt STOP/START and SYNC routines both lack this feature and ideally require the addition of the extra line: LI R13,>1000. The instruction could be added to the routines at lines 4365 and 3775 respectively. Putting the instruction at line 3775 of the SYNC routine does not reduce the maximum motor speed capability of the program.

It is fortunate that in the tests using the magslip interrupt based control program the workspace stored in R13 was always appropriate when RTWP executed.

4.8.3.12 Miscellaneous Errata in the Magslip Program

Two small mistakes are present in the magslip motor control program listing. The first concerns the label associated with the JGT instruction at line 3410. The label should be BOT (not MAIN).

The second "mistake" is concerned with the explanatory comment given with one of the CRU bit instructions. The comment on line 2120 describes CRU bit 9 as CRUBIT >129 whereas it should state either "bit 9 at CRU software base address >120" or "CRU bit address >132". There is no CRU bit address numbered >129 (in fact there are no odd software CRU bit addresses).

4.8.4 Magslip Interrupt Based Program Operation

No problems were experienced with the program once it had been debugged. The magslip optical position detector disc had six segment lines on it. Therefore, the "double-edge" circuitry of the Optical Position Detector Interface Unit was selected so that the necessary twelve SEG interrupts per revolution were generated.

The program started the magslip reliably and the magslip ran with identical performance in either direction. The speeds attained with selected load angles of 0°, 30°, 60°, and 90° were the same as those achieved with the magslip under logic circuit control (as described in Chapter 3).

Successful data display was achieved even with the magslip rotating at its maximum speed of about 11000 rpm. To ensure that the tachometer provided a correctly scaled speed, the scaling circuit (shown in fig. 4.25) was set so that it divided the SEG interrupts by a factor of two before passing them onto the tachometer (IC59 switch positions viewed from the top as follows: S2 set to right; S1, S3, and S4 set to left).

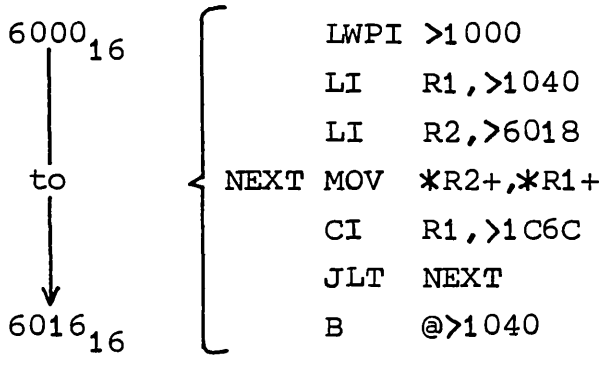
The stall detect and start abort software worked well. If the rotor was held stationary by hand during a start sequence it was possible to feel fifteen "kicks" as the stator flux rotated before the start was aborted. (The thirty starting steps set at line 2300 of the program fix the number of successive inverter control words to be output during a start sequence; reference to tables DRVF and DRVR shows that the inverter control words appear in pairs, and hence the number of inverter output changes and corresponding rotor steps are only fifteen.)

The ramping of the inverter supply voltages by the "main body" routine was such that it took about five seconds for the voltages to change from 0 to full volts and vice versa.

It is worth noting that the zero volts rail of the whole system is connected to the mains neutral via the magslip ±50 volt power supply variac, and so due care is needed during circuit adjustments, etcetera.

The program had been developed in the RAM area of the system memory and consequently variable data storage areas and flags were located at convenient points throughout the program. However, for the permanent storage of the assembled program in the system memory map, it was necessary to store the program in EPROM. To execute the program when it is located in EPROM requires the variable data stores and workspace registers to be defined in RAM areas of memory; it was felt that the modifications needed to the program would be time consuming and likely to introduce new bugs. Hence it was decided to store the complete program in EPROM and to transfer it all to the necessary area of RAM for execution.

The assembled program was stored in EPROM from address 6018_{16} to $6C42_{16}$ inclusive. The short "transfer" program shown below was stored between EPROM address 6000_{16} and 6016_{16} inclusive:-



To execute the magslip interrupt program the operator selects the debug (DP) routine of the software development board. Using the "Inspect Registers" (IR) command the program counter (PC) is set to 6000_{16} and the status register (ST) is set to zero. The program execution is then initiated by entering the "EX" command. The transfer program moves the magslip interrupt based program into the designated

area of RAM (1040_{16} to $1C6A_{16}$) and then branches to address 1040_{16} so that execution of the magslip program begins. The whole transfer operation takes a fraction of a second.

4.8.5 The 7 Phase Motor Interrupt Based Control Program

The assembly language listing of the interrupt based control program used to run the 7 phase motor is given in Appendix 4B. As in the case of the magslip program listing shown in Appendix 4A it is impractical to give the assembled object code listing because of its excessive length. In its assembled form the program occupies $6B8_{16}$ (1720_{10}) 16 bit words of memory starting from address 1040_{16}.

The program is in the main identical to the magslip program shown in Appendix 4A and so the detailed discussion of the magslip program given in sub-sections 4.8.3 through to 4.8.3.12 is of some relevance, although the line numbers quoted in those sub-sections do not necessarily correspond with the relevant sections of the 7 phase motor program listing.

There were four principal changes that had to be made to the interrupt based magslip program to make it suitable for the control of the 7 phase motor. They were the modification of:

(a) the look-up table (VTAB) used to control the inverter power supplies. The power supply for the 7 phase inverter works over a 0 to ±60 volt range instead of the 0 to ±50 volts of the magslip system;

(b) the look-up tables (DRVF and DRVR) used to provide the sequence of inverter control signals. These tables were rewritten so that they provided suitable control words for the 7 phase MOSFET inverter;

(c) the software used to implement load angle changes;

(d) the starting routine.

These changes are discussed briefly in the following sub-sections. All line references in sub-section 4.8.5.1 to 4.8.5.4 inclusive, refer to the program listing in Appendix 4B.

4.8.5.1 The 0 to ±60 volt Look-Up Table for the Interrupt Based 7 Phase Motor Program

The modified "VTAB" look-up table is located between lines 3890 and 3980 of the program listing given in Appendix 4B. The table contains the necessary power supply control bytes needed for the selection of integer voltages in the range 0 to 60 volts.

4.8.5.2 The Inverter Control Look-Up Tables for the Interrupt Based 7 Phase Motor Program

The inverter control look-up tables (DRVF beginning at line 4300, and DRVR beginning at line 4450) have to provide 14 bit control words for the 7 phase MOSFET inverter. The basic relationship between the hexadecimal inverter-control words generated by the microprocessor, and the pairs of control signals for the individual phases of the MOSFET inverter is shown in fig. 4.69. Fig. 4.69 clearly shows how bits D2 to D15 inclusive of each inverter control word, are allocated to activate the various phases of the inverter. (It should be remembered that bit D15 is the least significant bit of the hexadecimal control word.) Additional details on the bit allocations are available in Chapter 5, section 5.5.1. For each hexadecimal control word, fig. 4.69 also shows the resulting voltage polarities at the various phase outputs of the inverter. It can be seen that there are six phases on and one phase off for all the inverter control words shown in fig. 4.69. For each control word

HEXADECIMAL INVERTER CONTROL WORD (SEQUENCE AS IN DRVF TABLE IN APPENDIX 4B) ↓	RELATIONSHIP BETWEEN THE VARIOUS BITS (D2 TO D15 INCLUSIVE) OF THE INVERTER CONTROL WORD AND THE INVERTER PHASES, WITH THE RESULTANT VOLTAGE POLARITIES AT THE INVERTER PHASE OUTPUTS SHOWN																				
	PHASE 7			PHASE 6			PHASE 5			PHASE 4			PHASE 3			PHASE 2			PHASE 1		
	BIT D2	BIT D3	O/P POL	BIT D4	BIT D5	O/P POL	BIT D6	BIT D7	O/P POL	BIT D8	BIT D9	O/P POL	BIT D10	BIT D11	O/P POL	BIT D12	BIT D13	O/P POL	BIT D14	BIT D15	O/P POL
3666_{16}	1	1	NC	0	1	⊖	1	0	⊕	0	1	⊖	1	0	⊕	0	1	⊖	1	0	⊕
$1E66_{16}$	0	1	⊖	1	1	NC	1	0	⊕	0	1	⊖	1	0	⊕	0	1	⊖	1	0	⊕
$1B66_{16}$	0	1	⊖	1	0	⊕	1	1	NC	0	1	⊖	1	0	⊕	0	1	⊖	1	0	⊕
$19E6_{16}$	0	1	⊖	1	0	⊕	0	1	⊖	1	1	NC	1	0	⊕	0	1	⊖	1	0	⊕
$19B6_{16}$	0	1	⊖	1	0	⊕	0	1	⊖	1	0	⊕	1	1	NC	0	1	⊖	1	0	⊕
$199E_{16}$	0	1	⊖	1	0	⊕	0	1	⊖	1	0	⊕	0	1	⊖	1	1	NC	1	0	⊕
$199B_{16}$	0	1	⊖	1	0	⊕	0	1	⊖	1	0	⊕	0	1	⊖	1	0	⊕	1	1	NC
3999_{16}	1	1	NC	1	0	⊕	0	1	⊖	1	0	⊕	0	1	⊖	1	0	⊕	0	1	⊖
$2D99_{16}$	1	0	⊕	1	1	NC	0	1	⊖	1	0	⊕	0	1	⊖	1	0	⊕	0	1	⊖
2799_{16}	1	0	⊕	0	1	⊖	1	1	NC	1	0	⊕	0	1	⊖	1	0	⊕	0	1	⊖
$26D9_{16}$	1	0	⊕	0	1	⊖	1	0	⊕	1	1	NC	0	1	⊖	1	0	⊕	0	1	⊖
2679_{16}	1	0	⊕	0	1	⊖	1	0	⊕	0	1	⊖	1	1	NC	1	0	⊕	0	1	⊖
$266D_{16}$	1	0	⊕	0	1	⊖	1	0	⊕	0	1	⊖	1	0	⊕	1	1	NC	0	1	⊖
2667_{16}	1	0	⊕	0	1	⊖	1	0	⊕	0	1	⊖	1	0	⊕	0	1	⊖	1	1	NC

NOTE : (O/P POL) = VOLTAGE POLARITY AT PHASE OUTPUT

⊕ = POSITIVE VOLTS OUT

⊖ = NEGATIVE VOLTS OUT

NC = NO CONNECTION – I.E. ZERO VOLTS

FIG. 4.69. TRUTH TABLE FOR THE 7 PHASE MOSFET INVERTER

the relationship between the various phase voltage polarities is as required to take account of the winding configuration of the 7 phase synchronous motor (see Chapter 6). Fig. 4.70 shows schematically how the phase voltage waveforms are related to the inverter control words.

The number of 14 bit words in both the DRVF and DRVR look-up tables is determined by the number of SEG interrupts that occur per revolution. Hence the number of segments on the position sensor disc of the 7 phase motor had to be fixed before the look-up table software could be modified. The limiting number of SEG interrupts (31250) calculated in sub-section 4.8.3.6 was used to choose the number of SEG' pulses per revolution produced by the 7 phase motor position sensor. At the maximum designed speed of 30000 rpm (500rps) the position sensor can generate an absolute maximum of 31250/500 = 62.5 SEG interrupts per revolution. The nearest smaller number of SEG interrupts that was suitable for correct 7 phase motor operation was 56. Hence the chosen SEG interrupt rate was 56 per revolution, giving an angular resolution of 6.429°. Further details on the 7 phase motor position sensor disc are given in Chapter 6 (section 6.10).

The complete DRVF and DRVR tables are located from line 4300 to 4410 and from line 4450 to 4560 respectively. The approach of the end of each table is indicated by the most significant bit of the control words being set to logic 1. This is used during the motor start routine (as explained in sub-section 4.8.3.2) to prevent the microprocessor running off the end of the tables.

At a speed of 30000 rpm the position sensor produces 28000 SEG interrupts per second and so the interval between SEG interrupts is 35.71μs. This is just longer than the execution time of the SEG interrupt service routine and so it was thought that the required top speed of 30000 rpm might be just within the capabilities of the TMS9900.

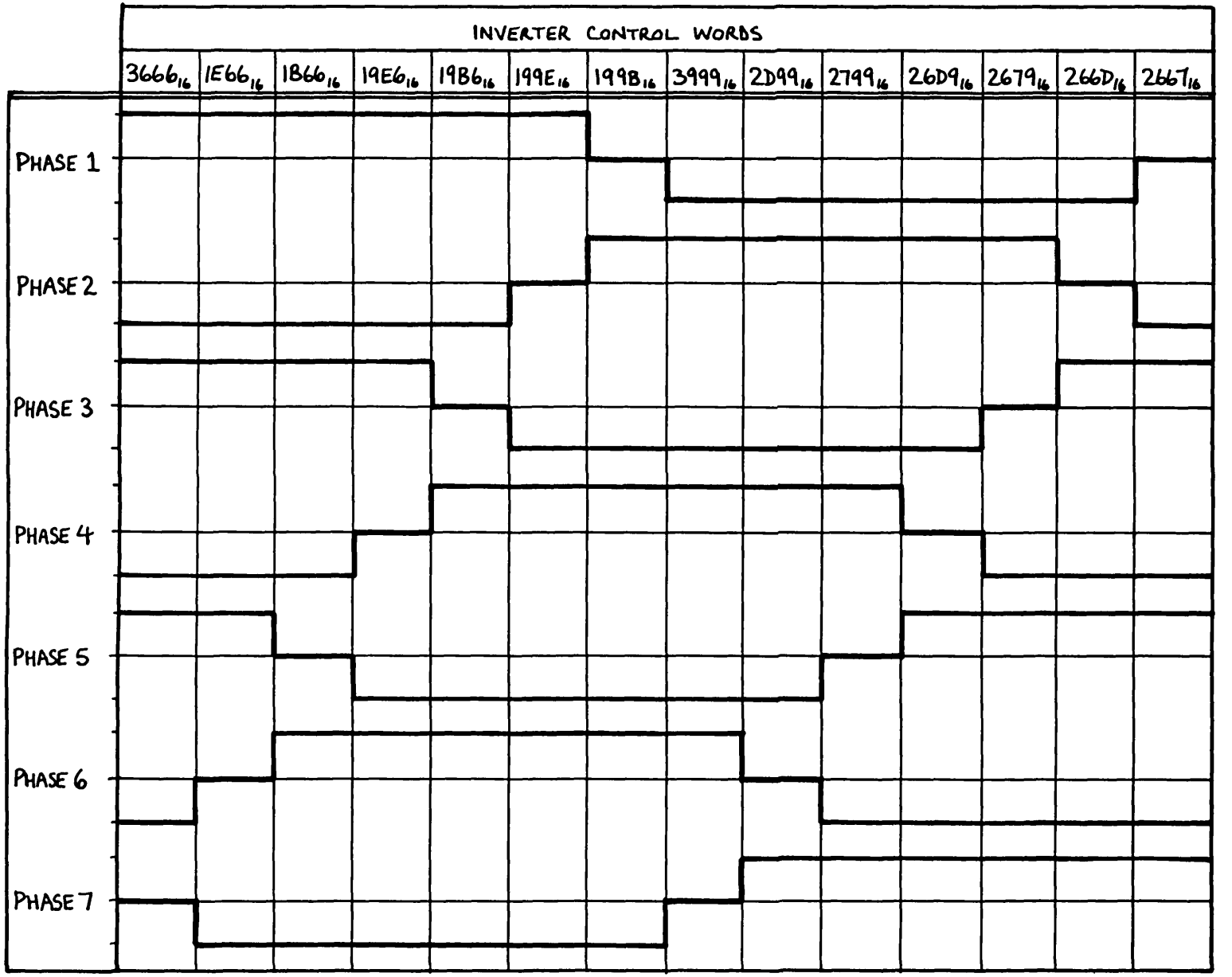


FIG. 4.70. RELATIONSHIP BETWEEN THE INVERTER CONTROL WORDS AND THE OUTPUT VOLTAGE WAVEFORMS

4.8.5.3 The Load Angle Selection Software

The load angle selection software is located from line 3370 to line 3720 of Appendix 4B. The load angles available are multiples of the basic position sensor resolution of 6.429°, whereas the load angles demanded via the keypad unit can only be integer values. Hence the software must choose the nearest available practical load angle and implement any change if necessary.

The practically available load angles are tabulated in fig. 4.71 along with a sequence of integer "trip values" which are held in look-up table LTAB (lines 4010 and 4020). The requested integer load angle is compared against the trip values stored in the LTAB look-up table in order to determine which "practical" load angle should be selected. For example, if a load angle of 17 degrees is requested, it results in a load angle of 19.29° being selected because the requested value lies between trip values 16 and 23. The rounded integer values of the available practical load angles are stored in another look-up table LVAL, and they are also shown in fig. 4.71. For the example given above, the appropriate rounded integer load angle in table LVAL is 19°. In the program the load angle selected from table LVAL is stored in the "old load angle" register LRQO. Any subsequent "new" load angle requests are compared with the contents of LRQO to determine if any action is required.

A flowchart showing the basic action of the load angle software is shown in fig. 4.72. The appropriate line numbers of the program listed in Appendix 4B are shown alongside the flowchart boxes. The routine basically compares a new request with the present implemented load angle value, and if a change is required it is implemented in 6.43° steps. On each pass through the routine, a change of 6.43° can be made and hence large changes take several passes through the routine to complete.

PRACTICALLY AVAILABLE LOAD ANGLES	TRIP VALUES IN TABLE LTAB	INTEGER WORKING LOAD ANGLES IN TABLE LVAL
	0	
0		0
	4	
6·43		6
	10	
12·86		13
	16	
19·29		19
	23	
25·71		26
	29	
32·14		32
	36	
38·57		39
	42	
45·00		45
	48	
51·43		51
	55	
57·86		58
	61	
64·29		64
	68	
70·71		71
	74	
77·14		77
	81	
83·57		84
	87	
90·00		90
	91	

FIG. 4.71. 7 PHASE MOTOR LOAD ANGLE LOOK-UP TABLE CONTENTS

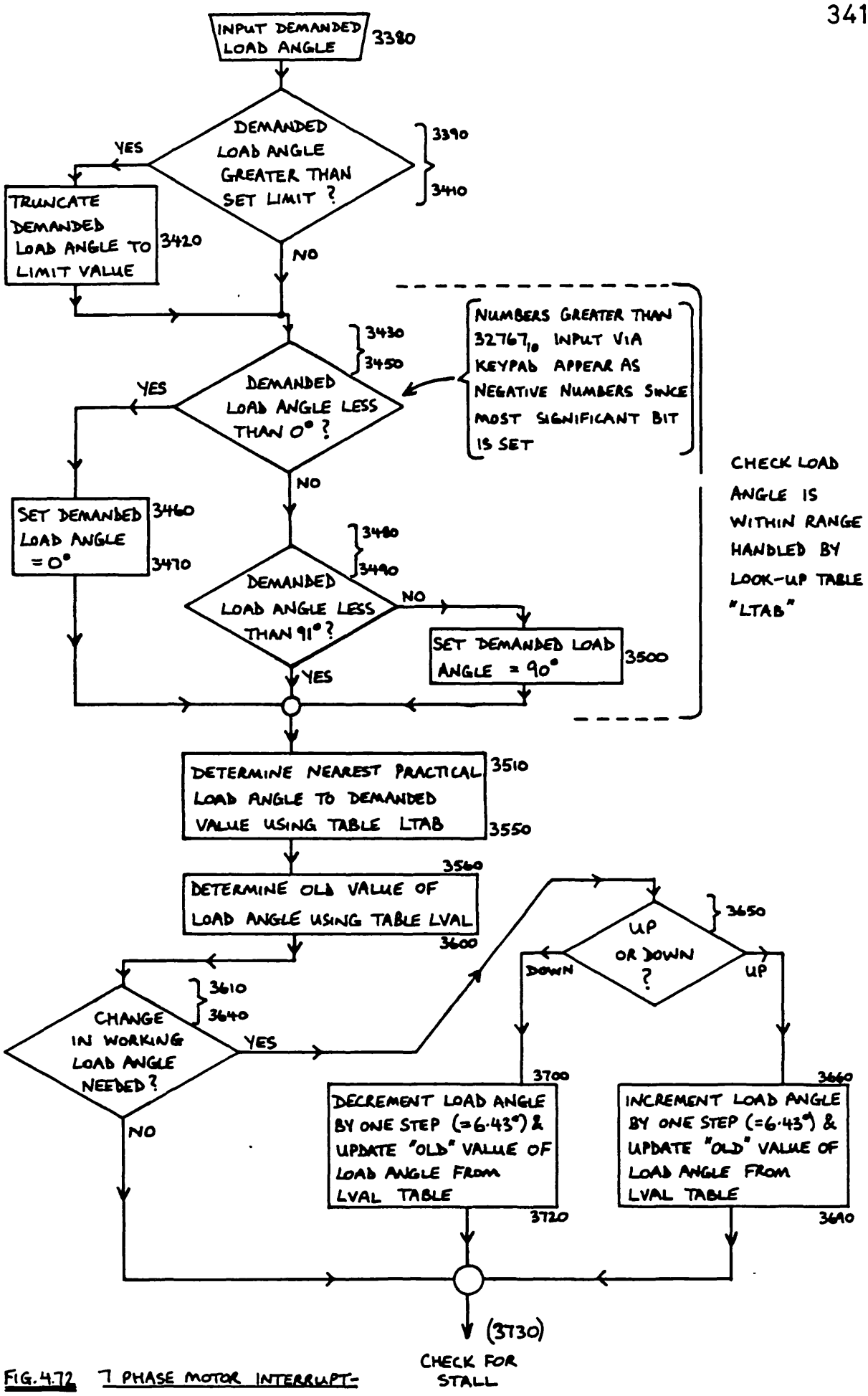


FIG. 4.72 7 PHASE MOTOR INTERRUPT-BASED-PROGRAM LOAD ANGLE SOFTWARE FLOWCHART

The 7 phase motor does not need large values of load angle for successful high speed operation, but nevertheless the routine allows load angles in the range 0° to 90° to be selected subject to the preset limit, which can be modified during the introductory part of the program.

4.8.5.4 The 7 Phase Motor Starting Routine

The starting routine is located between lines 2170 and 2530 of the listing given in Appendix 4B. It is essentially the same as the routine given in Appendix 4A for the magslip motor except that the number of starting steps has been increased to 150 and the delay time between each step has been reduced.

The delay time between each step is determined by the instructions at lines 2420 to 2440 and the delay is about 0.033 seconds. The inverter outputs actually change state after every four inverter control words (due to the control words repeating in groups of four in tables DRVF and DRVR), and so the stator flux steps on by one fourteenth of a revolution at intervals of about 0.132 seconds (four delay periods). Hence the stator flux makes one complete revolution in about 1.85 seconds, and this is sufficiently slow for the rotor to follow and for synchronism to be achieved.

The maximum number of starting steps permitted to achieve synchronisation before the start attempt is aborted was set at 150, so that just less than three revolutions of stator flux can occur (56 steps through table DRVF or DRVR being necessary per revolution of the stator flux). It was thought that this would be sufficient to ensure that the rotor would rotate around to give a SYNC interrupt; failure to receive such an interrupt within 150 steps resulting in entry to the "start abort" routine.

4.8.6 7 Phase Motor Interrupt-Based-Program Operation

No problems were experienced with the program during motor operation. The "single-edge" circuitry of the Optical Position Detector Interface Unit was selected so that the necessary 56 SEG interrupts per revolution were generated. Before any motor runs were attempted the position of the optical disc was adjusted on the motor shaft so that with a 0° load angle set up by the software, the applied phase voltage waveforms from the inverter were in phase with the motor back emf waveforms. This was achieved by rotating the motor by means of the eddy current brake and observing the relative positions of the phase 1 waveforms on an oscilloscope.

The program was stored in EPROM from memory address 8018_{16} to $86D0_{16}$ inclusive. The short routine shown below was stored from address 8000_{16} to 8016_{16} inclusive, and is used to transfer the stored program into the working area of RAM beginning at address 1040_{16}:-

```
8000₁₆      LWPI >1000
              LI   R1,>1040
              LI   R2,>8018
  to    NEXT  MOV  *R2+,*R1+
              CI   R1,>1DB0
              JLT  NEXT
8016₁₆        B    @>1040
```

The program execution is initiated in a similar manner to the magslip interrupt based program as described in subsection 4.8.4, with the obvious difference that the program counter (PC) is set to 8000_{16} prior to the EX command being entered.

To ensure that the tachometer provided a correctly scaled speed, the scaling circuit (shown in fig. 4.25) was set so that it divided the SEG interrupts by a factor of

eight before passing them onto the tachometer (IC59 switch positions viewed from the top as follows: S1 set to the left; S2, S3, and S4 set to the right).

During the 7 phase motor tests the microprocessor system successfully autopiloted the motor at speeds up to 25000 rpm. Vibration of the test rig and concern at the safety of the test rig prevented even higher motor speeds being attempted. At 25000 rpm the microprocessor was required to service SEG interrupts at an interval of only 42.857μs (a rate of 23334 interrupts per second). No problems were experienced with the operation of the lower priority STOP/START, DATA, or DISPLAY interrupts at the high test speeds, but it was obvious that at 25000 rpm the system was beginning to run into timing trouble because the applied phase voltages were suffering from continuous jitter.

Load angles of 0°, 6°, and 13° were selected via the keypad during the 7 phase motor tests, and the load angle selection routine worked as required.

A spurious jitter on the applied phase voltages (and resulting phase currents) was noted during the early tests whenever the motor speed exceeded about 16400 rpm (see Chapter 9 for photograph showing the jitter). The applied phase voltages were observed to be delayed fractionally in time with respect to the back emfs and it appeared that a -6° load angle was being set up. This effect can occur if the program misses one increment through the inverter control table. It was initially thought that this might have been due to a timing problem associated with either the SYNC and SEG interrupt service routines or the TMS9900 microprocessor interrupt handling logic. However, it was eventually realised that the jitter was a result of the optical position sensor disc being mounted on the motor shaft the wrong way round. This was causing a SEG interrupt to be generated just prior to a SYNC interrupt.

When the disc is mounted correctly the situation around the time when a SYNC' pulse is sensed is as shown by fig. 4.73, in which for clarity, only the relevant SEG' pulse is shown.

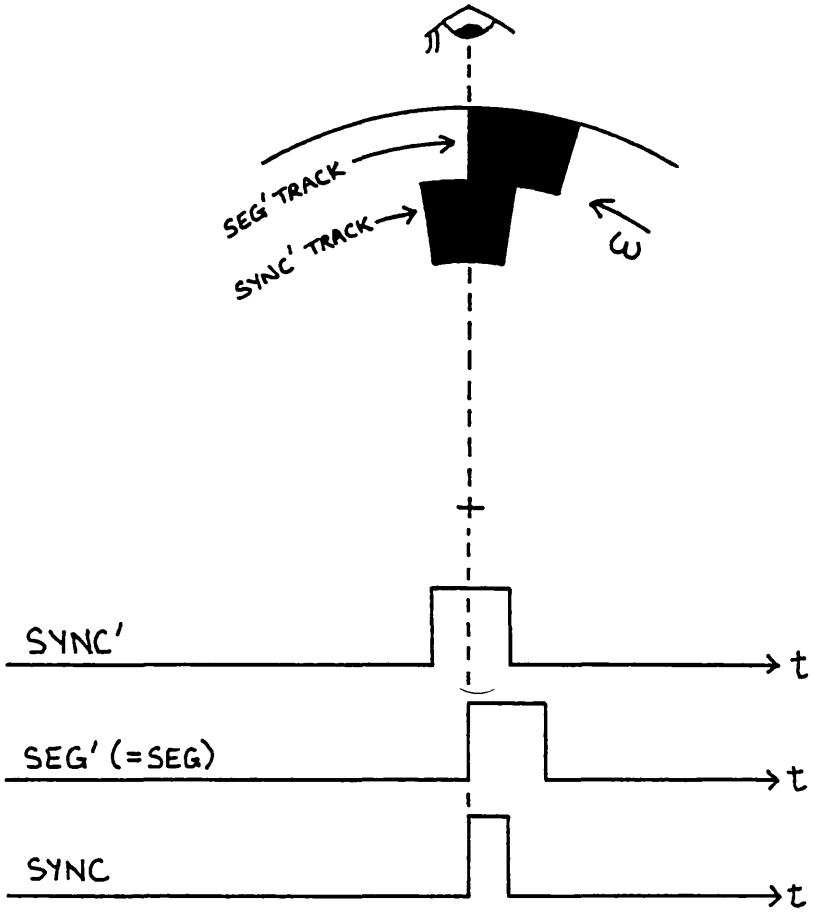


FIG. 4.73. INTERRUPT SIGNALS AT THE TIME OF A SYNC' PULSE WITH CORRECT DISC ORIENTATION

The SYNC and SEG interrupts, produced by the Optical Position Detector Interface Unit from the raw SYNC' and SEG' pulses, occur simultaneously. The SYNC interrupt is therefore serviced since it has the higher priority, and during its servicing the SEG interrupt is cancelled because the SYNC routine performs all the necessary operations for both interrupts.

When the disc is mounted incorrectly the situation around the time when a SYNC' pulse is sensed is as shown in fig. 4.74.

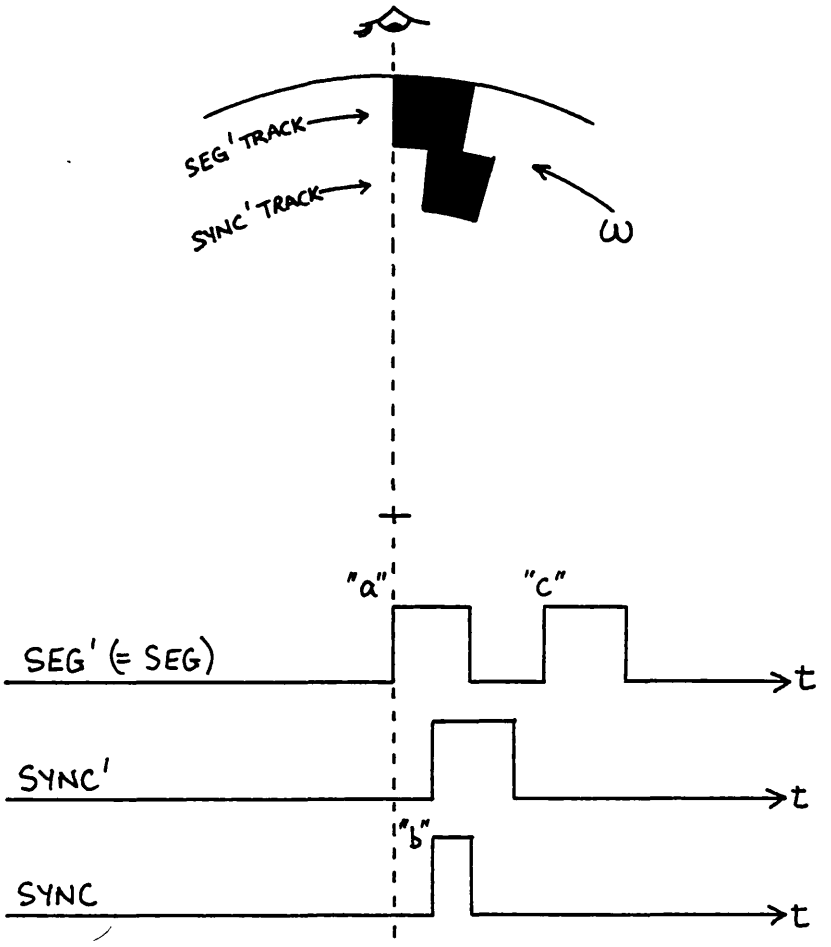


FIG.4.74. INTERRUPT SIGNALS AT THE TIME OF A SYNC' PULSE WITH INCORRECT DISC ORIENTATION

The incorrect disc orientation results in the SYNC interrupt not occurring simultaneously with a SEG interrupt. Fig. 4.74 shows interrupt edges "a", "b", and "c" which are relevant to the jitter phenomenon. There are 56 SEG' pulses per revolution and the angular span of the SYNC' pulse with respect to the SEG' pulses is indicated in fig. 4.75. (The SYNC' pulse actually subtends three quarters of the angle between adjacent SEG' pulses.)

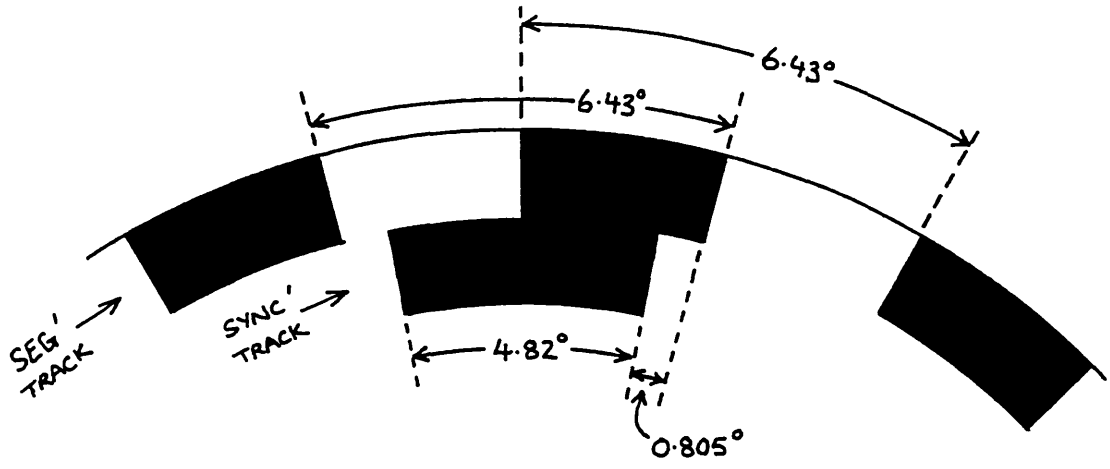


FIG. 4.75 THE ANGULAR RELATIONSHIP BETWEEN SYNC' AND SEG' PULSES

At the "onset of jitter" speed of 16400 rpm, edges "b" and "c" occur respectively 8.17μs and 65.33μs after edge "a". The microprocessor first senses the SEG interrupt caused by edge "a" and it starts to execute the SEG interrupt service routine as shown in fig. 4.76. It should be noted that the timings indicated in fig. 4.76 are with respect to edge "a" and assume that the microprocessor is immediately available to vector to the interrupt routine. The times relate to the ends of the instructions. In addition, the figures in brackets refer to the number of clock cycles per instruction.

During the context switch to the SEG interrupt service routine edge "b" generates a SYNC interrupt. Although the SYNC interrupt is of higher priority than SEG, it is not serviced until the completion of the first instruction in the SEG routine. Hence, 21.312μs after the detection of edge "a", the microprocessor performs a context switch to the SYNC interrupt service routine as shown in fig. 4.76. The SYNC interrupt routine is executed, and the micropro-

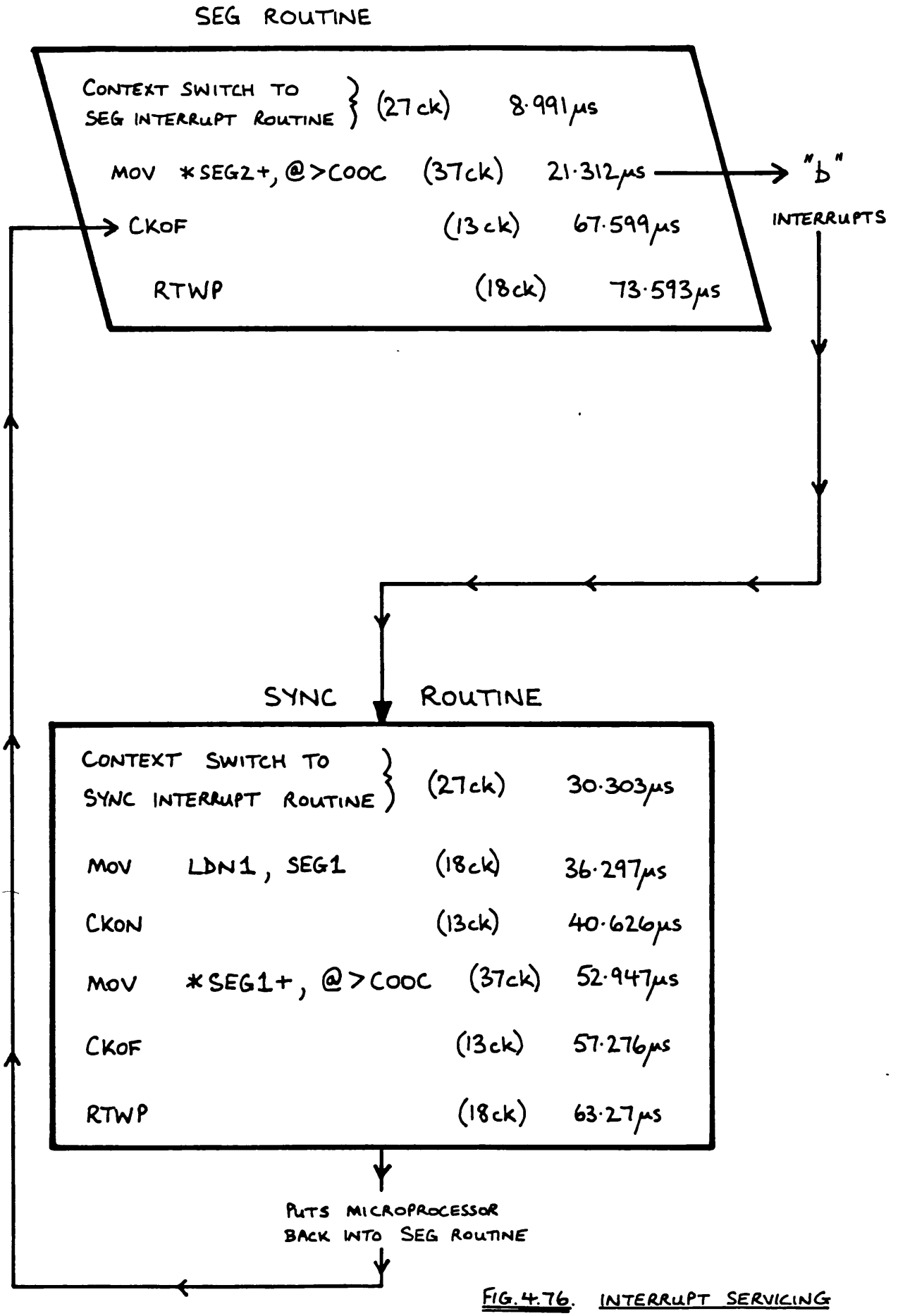


FIG. 4.76. INTERRUPT SERVICING ORDER WITH INCORRECT DISC ORIENTATION

cessor then returns to the SEG interrupt routine to complete its execution.

At speeds below 16400 rpm the contorted route taken through the SEG and SYNC routines is of no consequence to the operation of the motor. However, at a speed of 16400 rpm edge "c" occurs 65.33μs after edge "a". The microprocessor begins to execute the CKOF instruction of the SEG routine 63.27μs after edge "a", and finishes the instruction 67.599μs after edge "a". Hence at the time that the SEG interrupt latch should be registering the SEG interrupt caused by edge "c", it is forcibly being cleared by the CKOF instruction. This results in a SEG interrupt being missed, and so the program points one step behind where it should in the inverter control look-up table. This manifests itself as a load angle 6.43° less than that desired (i.e. -6° if 0° is actually wanted).

The SEG interrupt is not missed on every revolution at speeds around 16400 rpm because usually an instruction is being executed when edge "a" occurs. The resulting delay in switching to the SEG routine means that CKOF does not always coincide with edge "c", especially since the delay depends on the particular instruction being executed when "a" occurs.

After the disc orientation had been corrected, the motor was found to run up past 16400 rpm without jitter occurring. Some jitter was observed above 20000 rpm and the jitter became very bad at about 24000 rpm. Some of this jitter is due to the fact that the microprocessor can only begin to execute an interrupt service routine at the completion of its current instruction. The variable delay that occurs at successive SYNC interrupts causes some "wobble" in the front edges of the applied phase voltages with respect to the back emfs. However, the worst of the jitter is the result of a SEG interrupt being missed whilst the SYNC interrupt service routine is executing. At a speed of 24000

rpm the interval between SEG interrupts is 44.643μs. Assuming that no STOP/START, DATA, or DISPLAY interrupts are active, then the microprocessor executes instructions in the "main body" of the program inbetween servicing SEG and SYNC interrupts. The "main body" instructions have durations of between about 2.997μs (e.g. line 3100 JEQ ANG) to 11.988μs (line 3730 MOV @SLOW+8,@SLOW+8). Therefore, when a SYNC interrupt occurs there is a possible delay in vectoring to its service routine of up to about 12μs. If the delay is "t" microseconds, then the timing of the instructions following the SYNC interrupt is:

Completion of current instruction	0 to tμs
Context switch to SYNC routine	tμs to (t+8.991)μs
MOV LDN1,SEG1	(t+8.991)μs to (t+14.985)μs
CKON	(t+14.985)μs to (t+19.314)μs
MOV *SEG1+,@>C00C	(t+19.314)μs to (t+31.635)μs
CKOF	(t+31.635)μs to (t+35.964)μs
RTWP	(t+35.964)μs to (t+41.958)μs

The mid-point of the CKOF instruction occurs at (t+33.8) microseconds, and so a delay of 10.843μs results in CKOF clearing the SEG interrupt latch just as the next SEG interrupt is occurring. Hence a SEG interrupt can be missed, leading to a similar effect as that experienced when the optical disc was incorrectly orientated.

Obviously if the motor rotates even faster, the delay required to cause this trouble becomes shorter, and at 25000 rpm a delay of 9.06μs is sufficient. Since such a delay is possible several times in the "main body" routine it is not surprising that the jitter at 25000 rpm was so bad. Taking a maximum possible delay of 12μs as a limit, it is possible to predict a speed limit S_{max} below which jitter due to missed SEG interrupts should not normally occur. The equation is basically:

$$(t + 33.80 \times 10^{-6}) = 60/(56.S_{max}) \qquad 4.1$$

With a delay time of 12μs the limit speed is found to be 23394 rpm and this agrees with the observed system performance.

Methods of increasing the maximum "jitter-free" speed capability of the 7 phase motor interrupt based program are discussed in the final section of this chapter.

The minimum interval between a SEG interrupt and the output of the appropriate inverter control word is 21.312μs. At a speed of 25000 rpm, the delay of 21.312μs results in the rotor moving round by 3.2° from the interrupt point before the inverter receives its new instruction. Hence, the average value of load angle can have a minimum error of 3.2° due to the servicing delay. The error may be larger because the entry to the service routine can be delayed by up to about 12μs whilst the "currently" executing instruction is completed. With a 12μs entry delay, the load angle error is 5.0°. Therefore, in all cases the errors are less than the resolution of the system.

It is worth mentioning briefly that there is a very small finite chance of the microprocessor entering a SEG interrupt routine when a SYNC interrupt occurs. The SYNC interrupt is generated by ANDING the SEG and SYNC' signals. The TMS9901 senses the SEG and SYNC interrupts synchronously once every clock cycle. Therefore, if the sample occurs during the propagation time of the 74LS08 AND gate (typically 12ns) it is possible that the TMS9901 senses only a SEG interrupt. The microprocessor therefore starts to service the SEG interrupt. On the next clock cycle the TMS9901 senses the SYNC interrupt and a context switch to the SYNC routine occurs. The chances of this occurring are very small because of the short propagation time of the AND gate but if it does occur it could cause a SEG interrupt to be missed if the motor speed is above about 16400 rpm. However,

in practice it is not a problem. The chances of it occurring can be eliminated, if the SEG interrupt is delayed with respect to the SYNC interrupt, by passing it through a buffer gate having a suitable propagation delay.

4.8.7 Rotor-Position-Code Based Autopiloting Program Flowchart for the Magslip

The philosophy of the rotor position code based program used to control the magslip is indicated in the basic flowchart shown in fig. 4.77. The flowchart is similar to the interrupt based flowchart shown in fig. 4.55 but there is an important difference; the rotor position is sampled on two occasions during each loop through the "motor run" section of the program because SYNC and SEG interrupts are not used for position sensing. Two samples per loop are necessary to ensure that the interval between samples is sufficiently short for correct autopiloted operation of the magslip at speeds up to 12000 rpm. The absence of a SYNC interrupt also means that the start routine of the program must be different from that used by the interrupt based magslip program.

The flowchart shown in fig. 4.77 represents some operations in ways which do not occur in practice, as already mentioned with regard to fig. 4.55. For example, changes in the motor variables occur over a number of successive loops through the program, and not instantly as is suggested by the flowchart.

The basic flowcharts for the STOP/START, DATA, and DISPLAY interrupts are the same as those given for the magslip interrupt based program (see figs. 4.58, 4.59, and 4.60 respectively), with the obvious exception that references to fig. 4.55 in fig. 4.58 should be read as fig. 4.77.

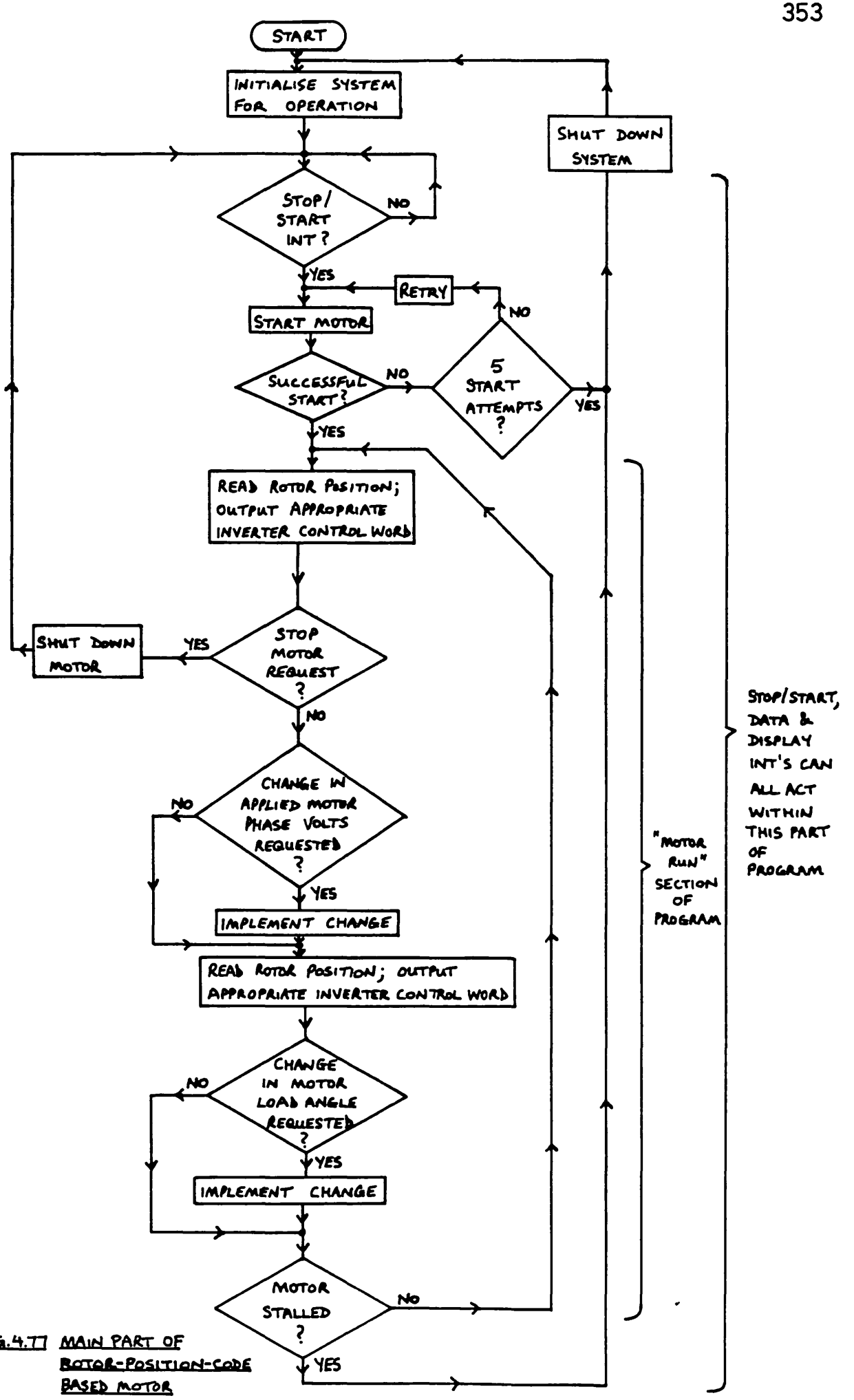


FIG. 4.77 MAIN PART OF ROTOR-POSITION-CODE BASED MOTOR CONTROL PROGRAM

4.8.8 The Magslip System Rotor-Position-Code Control Program

The magslip system TMS9900 assembly language program based on the flowcharts referenced in sub-section 4.8.7 is given in Appendix 4C. The assembled object form of the program has not been given because of its excessive length. In its assembled form the program occupies 650_{16} (1616_{10}) 16 bit words of memory starting from address 1040_{16}. Much of the program in Appendix 4C is the same as the interrupt based magslip control program given in Appendix 4A and so many of the comments made about that program in sub-sections 4.8.3 to 4.8.3.12 inclusive are also relevant to the program in Appendix 4C. However, there are some major differences and these are dealt with in the following sub-sections. All line references in sub-sections 4.8.8 to 4.8.8.5 inclusive, refer to the program listing in Appendix 4C.

4.8.8.1 Interrupts Used in the Rotor-Position-Code Based Program

The Rotor Position Code based program does not use the SYNC or SEG interrupts to perform the autopiloting task, and so their vectoring addresses are not initialised during the introductory parts of the program. However, the program does use the STOP/START, DATA, and DISPLAY interrupts, and they are serviced by the same software as that used by the "interrupt-based" autopiloting programs shown in Appendices 4A and 4B.

4.8.8.2 The Overlapped Workspaces in the Rotor-Position-Code Based Program

The STOP/START interrupt service routine workspace is the same as the workspace used by the "main body" routines of the program. The register names referred to in Appendix 4C and their respective memory addresses are shown in fig. 4.78. Register 9 of the "main body" workspace is called

MEMORY ADDRESS	"MAIN - BODY" ROUTINES WORKSPACE	'9900 INT8 STOP/START ROUTINE WORKSPACE
1000_{16}	RØ	RØ
1002_{16}	R1 VRQN	R1
1004_{16}	R2 VRQO	R2
1006_{16}	R3 DANW	R3
1008_{16}	R4 DAOL	R4
$100A_{16}$	R5 LRQN	R5
$100C_{16}$	R6 LRQO	R6
$100E_{16}$	R7	R7
1010_{16}	R8	R8
1012_{16}	R9 POIN	R9
1014_{16}	R10 LDAN	R10
1016_{16}	R11	R11
1018_{16}	R12 CRUB	R12 CRUB
$101A_{16}$	R13	R13 WP_8
$101C_{16}$	R14	R14 PC_8
$101E_{16}$	R15	R15 ST_8

FIG. 4.78. OVERLAPPED WORKSPACES IN ROTOR-POSITION-CODE BASED MAGSLIP CONTROL PROGRAM.

POIN; it is used to store the sampled rotor-position-code and it "points" to the next inverter control word in memory. Register 11 of the same workspace is reserved to store the "return from subroutine" program counter information, needed by the Branch and Link (BL) instruction. This instruction is used to call the delay subroutine (DELY), which provides a 1.51 second wait.

4.8.8.3 Comments on the Rotor-Position-Code Input Port and the Rotor-Position-Code Register "POIN"

The six bit position code produced by the Optical Position Detector Interface Unit cycles through the binary number sequences 0 to 11, 0 to 13, or 0 to 55 depending on the range that is selected. For operation with the magslip, the unit is set to the 0 to 11 count range. The count value is used to "point" to the next inverter control word in the chosen inverter control look-up table (DRVF or DRVR). There are twelve control words required for one cycle of the inverter outputs, and each position code value selects one specific control word. The code value is added to the base address of the first word in the relevant look-up table to form the required "look-up" address. However, since the memory address values of consecutive control words differ by two, it is necessary to multiply the code value by two before it is added to the base address. This multiplication could be performed by software. For example, having read the position code into register POIN, the microprocessor could perform a "shift left arithmetic" (by one bit towards the most significant end) on the contents of POIN, thus achieving the required multiplication. However, this does take up valuable program execution time. Therefore, a hardware multiplication was performed by connecting the six bit position code bus of the Position Detector Interface Unit to the rotor position code input port, such that each bit was routed to the next higher significant bit of the TMS-9900 data bus (that is: the least significant bit of the count value went to data bus bit D14, not D15; the most

significant bit went to data bus bit D9, not D10). Both the least and most significant bits of the 8 bit rotor position code input port were connected to ground, so that the least significant byte of the TMS9900 data bus received a "times two" value of the rotor position code whenever the input port was accessed.

The rotor position code input port does not control the most significant byte of the TMS9900 data bus. Hence, if a word of data is moved from the port into a memory based register, the most significant byte of that register is "all ones" and obviously must be masked if the true value of the input data is required. However, the byte containing the rotor position code can be specifically accessed by using byte instructions with memory address $C009_{16}$. The Compare Byte (CB) and Move Byte (MOVB) instructions are both used in the program, as explained in the following two sub-sections.

4.8.8.4 The Starting Procedure for the Rotor-Position-Code Based Program

The flowchart for the starting routine of the Rotor-Position-Code based program is shown in figs. 4.79(a) to 4.79(d) inclusive. The relationship between a particular box in the flowchart, and the corresponding lines of instructions in the program listing in Appendix 4C, is indicated by line numbers at the side of the box.

The actual starting procedure is arranged in two stages. The first stage is needed to synchronise the Rotor-Position-Code with the actual rotor position, because when the unit is first switched on the counter unit output is random and bears no relationship to the rotor orientation. In addition, the counter can become unsynchronised from the rotor position if the rotor oscillates as it comes to a halt during a stop procedure. The synchronisation is achieved by stepping the inverter output slowly, and it relies on the rotor

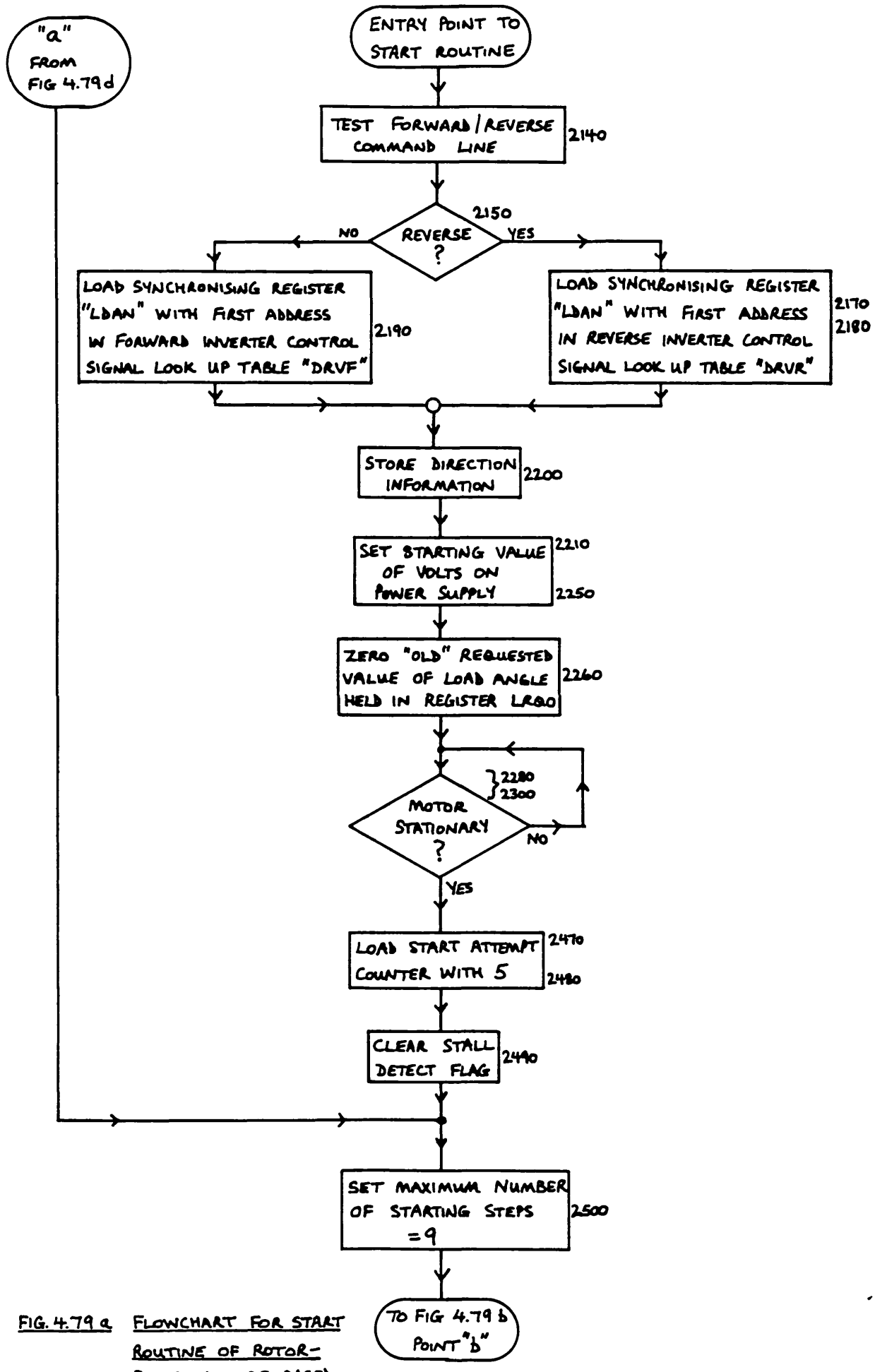


FIG. 4.79a FLOWCHART FOR START ROUTINE OF ROTOR-POSITION-CODE BASED PROGRAM

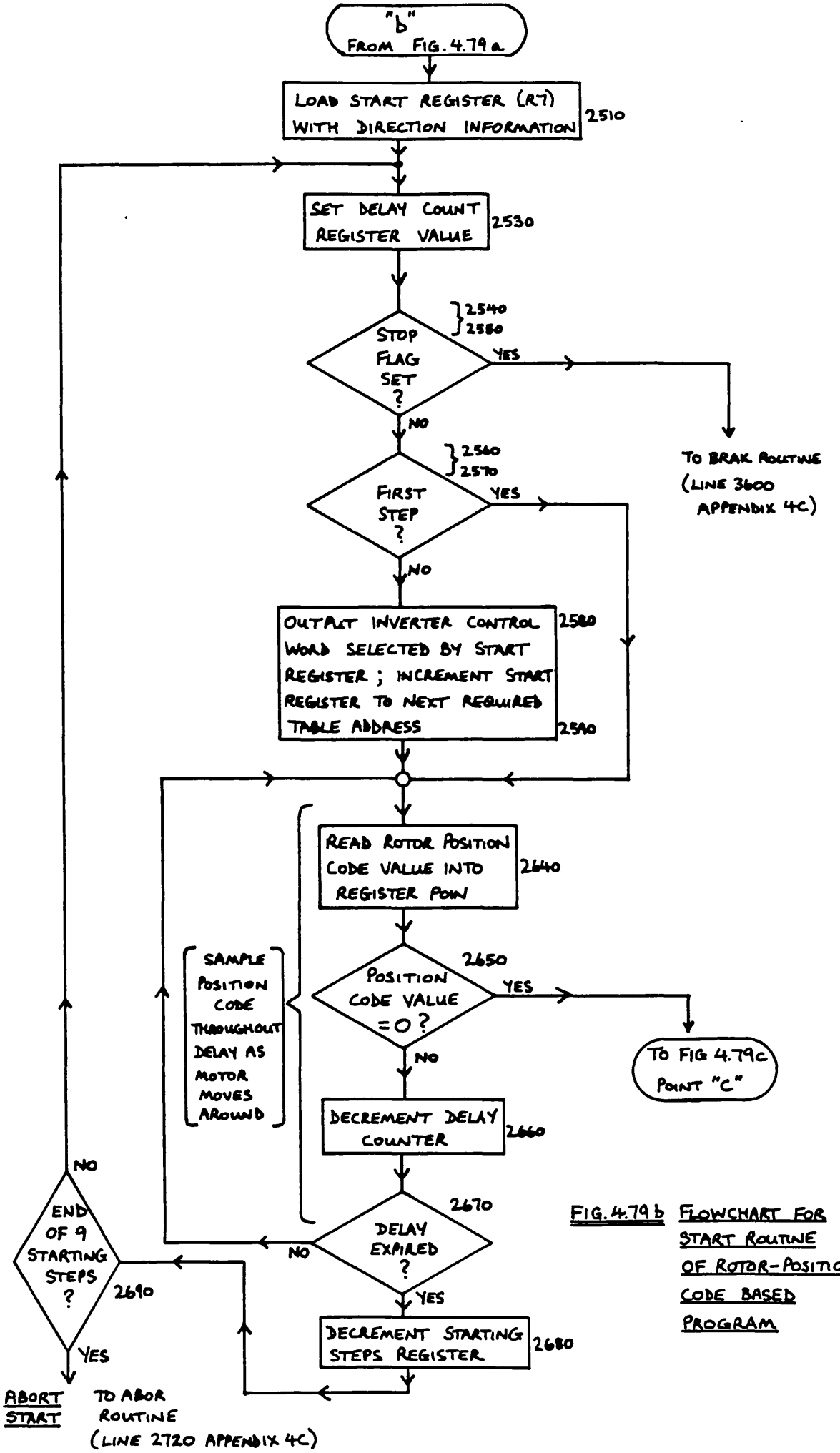


FIG. 4.79 b FLOWCHART FOR START ROUTINE OF ROTOR-POSITION-CODE BASED PROGRAM

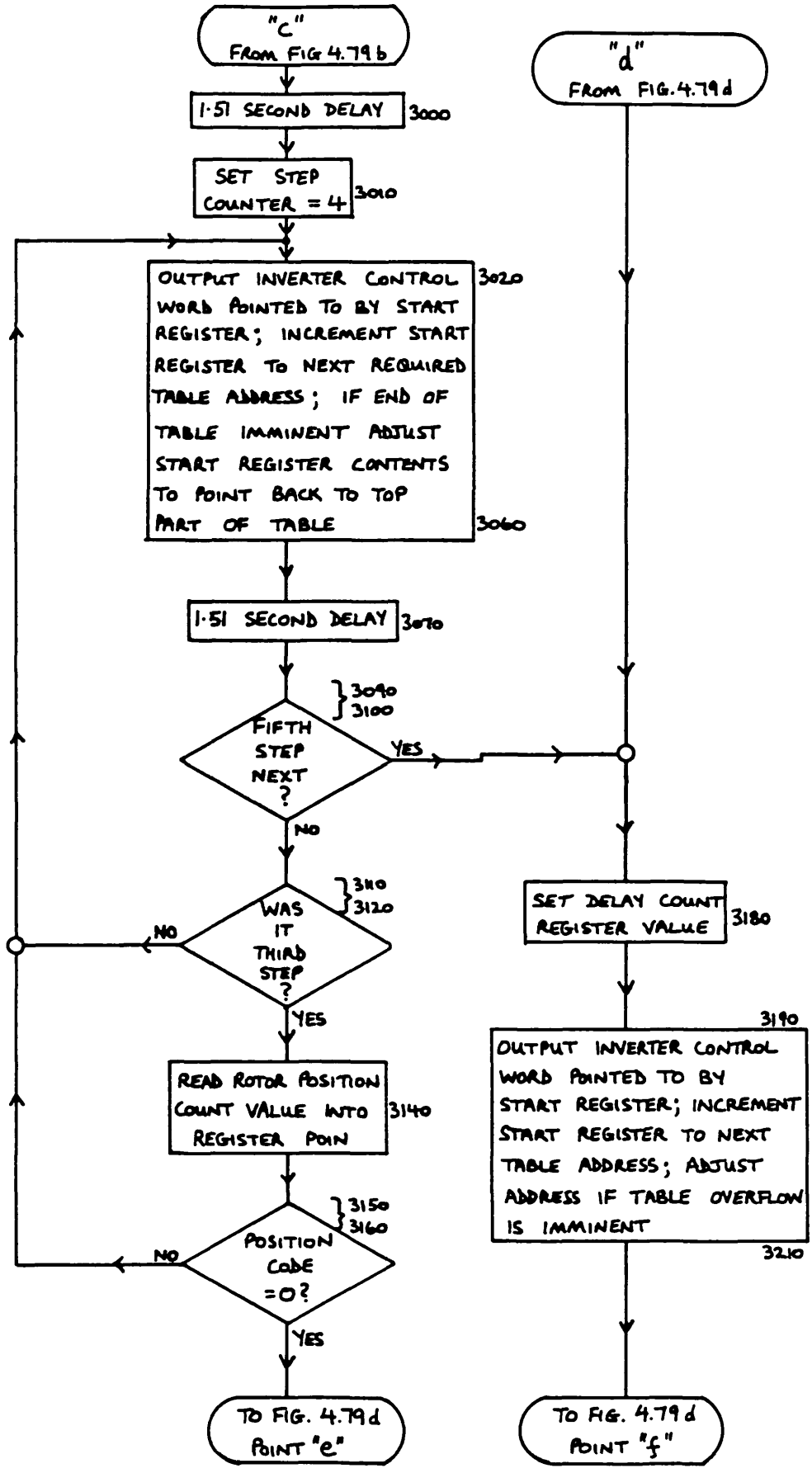


FIG. 4.79c FLOWCHART FOR START ROUTINE OF ROTOR-POSITION-CODE BASED PROGRAM

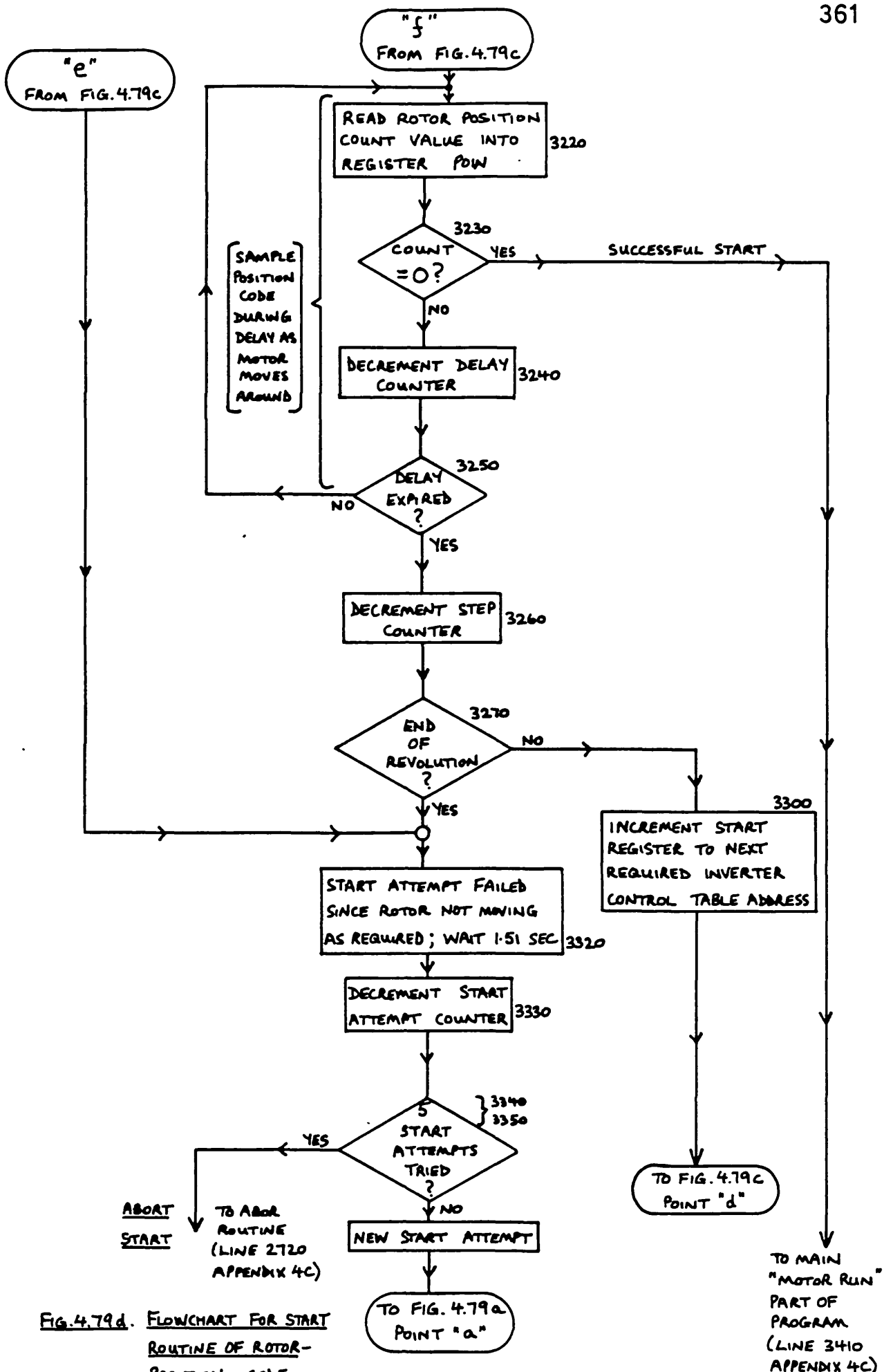


FIG. 4.79d. FLOWCHART FOR START ROUTINE OF ROTOR-POSITION-CODE BASED PROGRAM

following the slowly stepping stator field. Obviously, the system can fail if the rotor cannot freely move. The program continuously checks the rotor position code to detect when the code becomes zero; this occurs when the counter circuit is synchronised with the rotor position by the SYNC pulse. The stator field is advanced by a maximum of nine steps (that is one and a half revolutions), and so if the rotor is free to move a SYNC pulse should be generated and a zero count detected at some point during the nine steps. If a zero count is not detected by the end of the ninth step, the program assumes that the rotor is too heavily loaded for the starting torque to move it, and so the start attempt is totally abandoned. It should be noted that there is an error in the program statement at line 2560 of Appendix 4C. The statement should be "CI R0,9" and not "CI R0,10". In its correct form the statement detects when the program is about to perform the first step in the first stage of the start sequence. If this is the case, then the microprocessor first samples the rotor position code to check if it is zero. A zero value means that the counter circuit is most probably synchronised with the rotor position and so the microprocessor proceeds directly to the second stage of the start sequence.

If a zero count is detected within the permitted number of first stage starting steps, the microprocessor moves on to the second stage of the starting procedure. This checks that the rotor is moving in step with the stator field, and it is a way of ensuring that the detection of the zero count during the first stage is not just a fluke. Since the rotor tends to oscillate as it is slowly stepped around, the counter circuit output can advance faster than it should do due to multiple SEG pulses occurring during the oscillations. Hence it is not possible to check that the count increments by a fixed amount on each step. Therefore, the second stage checks that the rotor is moving in step with the stator field in a roundabout way. Three steps after the detection of a zero count in the first stage of the starting procedure,

the microprocessor samples the position code to see if it is non-zero; a zero reading indicates that the rotor is not moving as required and so the start attempt is halted. The microprocessor returns to the first stage of the start procedure to re-attempt a start procedure. However, if a non-zero code value is detected, the microprocessor continues to step the rotor and during the fifth, sixth, or seventh steps, it samples the position code at frequent intervals looking for a zero count value. A zero count value during any of these steps indicates a successful start and so the microprocessor then enters the "motor run" section of the program. If a zero count is not detected during these steps, the program assumes that the rotor has not rotated correctly in step with the stator field, and so the start attempt is halted. The microprocessor returns to the first stage of the start procedure to re-attempt a start. A zero count can occur during the fifth, sixth, or seventh steps of the starting procedure second stage because of the slightly irregular movement of the rotor as it steps. The rotor does not move round by exactly the same angle on each step, and so it is possible for a SYNC pulse to occur either right at the end of the fifth step, at any time during the sixth step, or at the beginning of the seventh step. The detection of a zero count during any of these steps indicates that the rotor has successfully rotated by one revolution during the second stage of the starting procedure.

It should be noted that in the event of a start failure in the second stage, the program permits the microprocessor to return to the first stage up to five times before the start procedure is totally abandoned.

The detection of a zero code value is achieved during the start procedure, by moving the rotor-position-code byte into register POIN using the MOVB instruction. This instruction automatically compares the byte to zero and so a subsequent conditional jump instruction can act on the result of the comparison.

The only other point worth highlighting in the start procedure is the need to increment the inverter control pointer address by two word locations between each step operation. This ensures that the inverter outputs change each time a new control word is presented to the inverter, and hence the stator field (and hopefully the rotor) move on at each "step" instruction. The required pointer increment is achieved by the use of the auto increment feature (for example see line 2580 in Appendix 4C) followed by an increment-by-two instruction (line 2590).

4.8.8.5 The Rotor Position Detection Software for Auto-piloted Operation

For magslip operation at speeds up to 12000 rpm with the load angle controlled in 30° steps (0°, 30°, 60°, 90°), it was necessary to arrange the rotor position code to be sampled at least once in each of the 30° sectors making up a revolution. With a sample at least every 30° of rotor movement, there are then a minimum of 12 samples per revolution. The minimum time interval between samples is determined by the rotor speed. With 12 samples per revolution at a rotor speed of 12000 rpm, there are 2400 samples per second and the time interval between them is 417μs.

It was decided to arrange the sampling software at suitable points in the "motor-run" section of the program, such that a sample was made at least every 400μs. Fixing the sample interval to meet the high speed requirements, does mean that far more samples than are necessary are made at lower rotor speeds, but it is a simple strategy to implement. A more involved strategy would involve changing the sample interval to suit the actual rotor speed.

There are two sample points in the "motor-run" section of the program. The first sample point is located between lines 3460 and 3530 and the second between lines 4120 and 4170. The actual maximum interval between the samples is

much less than the target of 400μs, and so good autopiloting is assured at all speeds. The basic sample and autopiloting routine is:

```
MAIN  MOV  @>C008,POIN
      CB   @>C009,@>1013
      JNE  MAIN
      ANDI POIN,>1E
      A    LDAN,POIN
      MOV  *POIN,@>C00C
```

Sample rotor position and output appropriate inverter control word.

The flowchart for the routine is shown in fig. 4.80.

The first instruction of the routine moves the rotor-position-code into register POIN. The position-code byte goes into the least significant byte of register POIN; that is, it is placed at memory address 1013_{16}.

The next instruction (CB @>C009,@>1013) compares the newly sampled byte placed at address 1013_{16} with the position code held by the Rotor Position Code input port at address $C009_{16}$. If the bytes do not agree, it is reasonable to assume that the sampled byte is not necessarily a valid position code: for example, the code may have been changing as it was sampled. Therefore, if the bytes differ the program directs the microprocessor to sample the code again. This process continues until a match is achieved.

Once the routine has accepted a "valid" position code the next instruction (ANDI POIN,>1E) masks the unwanted bits in register POIN so that POIN contains the "times two" value of the rotor position code as previously mentioned in subsection 4.8.8.3. The inverter-control look-up table base address contained in register LDAN, is then added to the rotor position code in register POIN to form the address of the required inverter control word. The required load angle is set up by adjustments made to the base address held by register LDAN; these are performed by the load angle routine

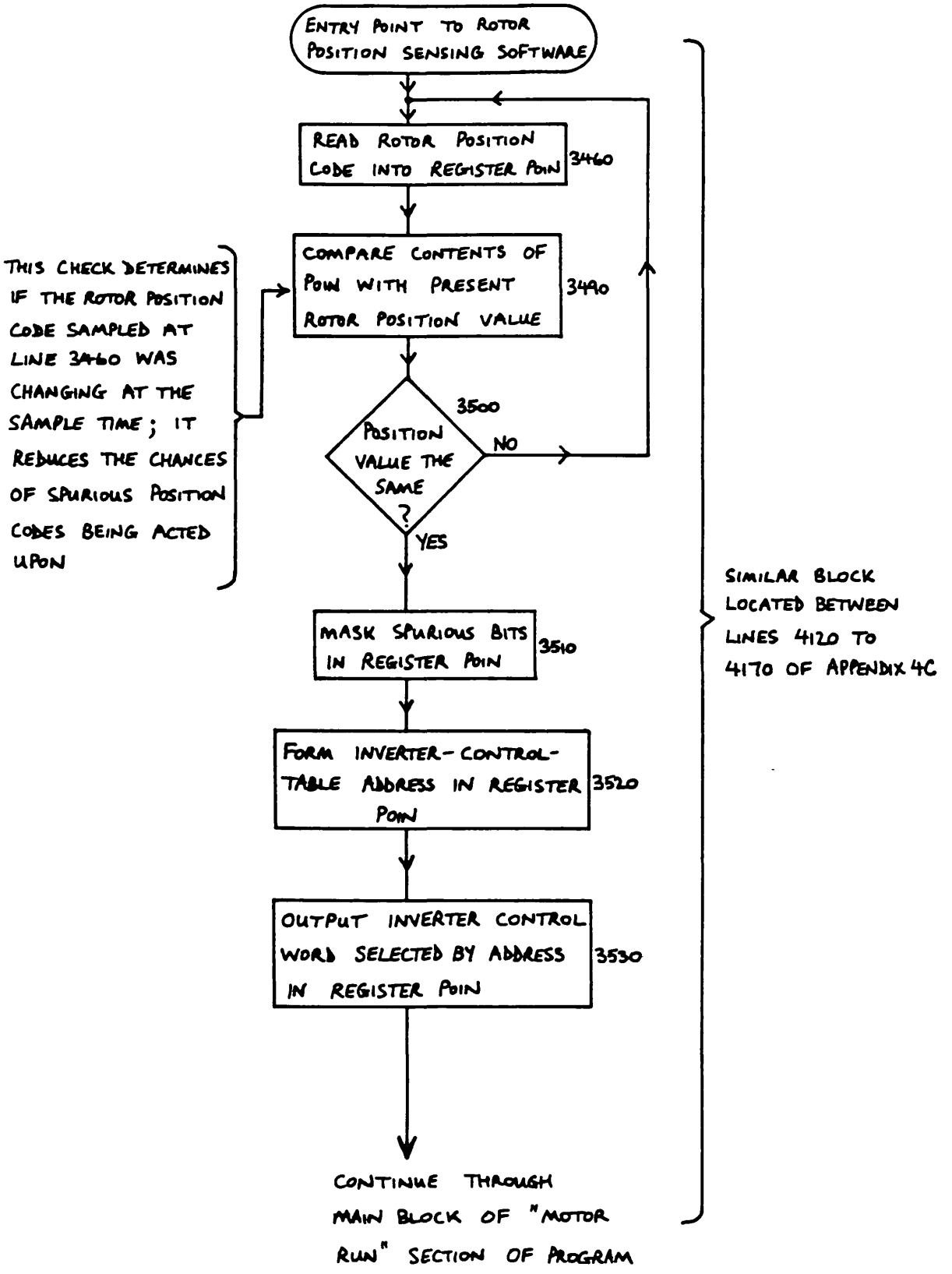


FIG. 4.80. FLOWCHART FOR THE ROTOR POSITION SENSING SOFTWARE (ACTIVE DURING "MOTOR RUN" PORTION OF PROGRAM)

located between lines 4180 and 4340.

Having formed the required inverter control word address, the final instruction of the sample and autopiloting routine (MOV *POIN,@>C00C) outputs the relevant control word to the inverter. The instruction uses workspace register indirect addressing, so that the contents of the memory address pointed to by register POIN are moved to the inverter control inputs.

The time taken for the execution of the sample and autopiloting routine (assuming that a "valid" code is sampled at the first attempt) is 46.287μs (139 clock cycles).

The interval between a position code sample and the output of the appropriate inverter control word is 37.296μs. This is slightly longer than the minimum time of 21.312μs taken by the SEG interrupt service routine to output an inverter control word following an interrupt, but it has little effect on the performance of the system. At 12000 rpm the 37.296μs delay results in the rotor moving round by 2.685° from the sample point before the inverter receives its new instruction. Hence, the average value of load angle can have a maximum error of 2.685° caused by the sample and autopiloting routine delay. The equivalent "error" in the interrupt based programs varies because of the variable delay time associated with the completion of the "current" instruction before a SYNC or SEG interrupt can be serviced.

4.8.9 Magslip Rotor-Position-Code Based Program Operation

No problems were experienced with the program during operation. The Optical Position Detector Interface Unit was set to the divide by 12 mode to provide the required position code sequence.

The program was stored in EPROM from memory address 7018_{16} to 7668_{16} inclusive. The short routine shown below

was stored from address 7000_{16} to 7016_{16} inclusive, and is used to transfer the stored program into the working area of RAM beginning at address 1040_{16}:-

7000_{16} to 7016_{16}

```
      LWPI >1000
      LI   R1,>1040
      LI   R2,>9018
NEXT  MOV  *R2+,*R1+
      CI   R1,>1DB0
      JLT  NEXT
      B    @>1040
```

The program execution is initiated in a similar manner to the magslip interrupt based program as described in subsection 4.8.4, with the difference that the program counter (PC) is set to 7000_{16} prior to the EX command being entered.

To ensure that the tachometer provided a correctly scaled speed, the scaling circuit (shown in fig. 4.25) was set so that it divided the SEG interrupts by a factor of two before passing them on to the tachometer (IC59 switch positions viewed from the top as follows: S1 and S2 set to the right; S3 and S4 set to the left).

During the practical tests the microprocessor successfully autopiloted the magslip up to a speed of about 11000 rpm. The magslip started reliably in either direction, and load angles of 0°, 30°, 60°, and 90° were successfully selected. The magslip would have rotated even faster if a higher inverter supply voltage (>50 volts) had been available.

4.9 Conclusions and Possible Programming Improvements

All of the programs described in this chapter worked well, and together they demonstrate that a microprocessor can be used very successfully as the autopiloting controller

in high speed synchronous motor drive systems. The software programmed features of the system allowed easy and convenient variable changes to be made in the introductory section of the programs. The keypad unit in conjunction with the software routines, permitted easy and relatively foolproof data entry and data recall to be achieved. The health monitoring capability of the microprocessor was demonstrated by the diagnostic messages displayed on the VDU whenever a start sequence failed or the motor stalled.

The interrupt-based and position-code-based magslip programs gave virtually identical performances, and there is little to choose between them in a three phase "medium-speed" application of the magslip type.

The interrupt-based 7 phase motor program performed very well and allowed motor speeds up to 24000 rpm to be comfortably attained. The system successfully serviced 23333 interrupts per second when the motor speed was pushed up to 25000 rpm, although waveform jitter due to missed SEG interrupts was very serious at this speed. The jitter in effect placed a speed limit of 24000 rpm on the motor, but it could be removed by a simple re-ordering of the instructions in the SYNC interrupt service routine. The jitter stems from the problem of the CKOF instruction clearing the SEG interrupt latch as a new SEG interrupt occurs, and this leads to the SEG interrupt being missed. Placing the CKOF instruction at the start of the SYNC interrupt service routine, has the effect of moving the clearing operation out of the critical time slot when the following SEG interrupt occurs. The CKOF instruction in the SEG interrupt service routine can be similarly moved to minimise the chances of jitter being caused by the SEG routine itself. The two routines therefore become:

```
SEG:    SEGI CKOF
             MOV  *SEG2+,@>C00C
             RTWP
```

```
SYNC:    SYRN CKOF
              MOV  LDN1,SEG1
              MOV  *SEG1+,@>C00C
              CKON
              RTWP
```

However, even if the jitter problem is removed at speeds around 24000 to 25000 rpm, it is reasonable to state that the microprocessor is "running out of time" at these speeds. To comfortably achieve a speed of 30000 rpm, with 55 SEG interrupts and 1 SYNC interrupt per revolution, it would be necessary to reduce the execution times of both the SYNC and SEG interrupt service routines. The execution times of the SYNC and SEG routines described earlier in this chapter are 41.958μs and 31.635μs respectively (including context switch time and assuming no delay in the start of execution). At a speed of 30000 rpm the interval between SEG interrupts is only 35.714μs (a rate of 28000 interrupts per second), and so both the SYNC and SEG routines are really too long to execute in the available time intervals. A simple way to shorten both interrupt service routines is to arrange a different method of clearing the SYNC and SEG interrupt latches, so that software instructions are not needed. For example, the memory-mapped signal "$C00C_{16}$ Enable", which clocks data into the inverter control output port, could also be used to simultaneously clear both the SYNC and SEG interrupt latches. With this arrangement the instruction "MOV *SEG1+,@>C00C" in the SYNC routine performs both the required data transfer and clears the interrupt latches. This is also true for the instruction "MOV *SEG2+,@>C00C" in the SEG routine. (In general, any instruction which activates the "$C00C_{16}$ Enable" signal during its execution, will clear the interrupt latches with this arrangement.) This solution is feasible because independent clear signals for the SYNC and SEG interrupt service routines are not really needed, and the correct execution of the software is not affected by clearing either interrupt request at the same time as the new inverter control word

is output. The elimination of the CKON and CKOF instructions from the interrupt service routines leads to a significant reduction in their execution times. The SYNC and SEG interrupt service routines are then as shown below:

```
SYNC:    SYRN MOV  LDN1,SEG1
              MOV  *SEG1+,@>C00C
              RTWP

SEG:     SEGI MOV  *SEG2+,@>C00C
              RTWP
```

The execution times for the shortened SYNC and SEG routines (including the context switches and assuming immediate vectoring to the interrupt routines) are 33.3μs and 27.306μs respectively. Therefore, they can both execute in the time between consecutive SEG interrupts at 30000 rpm. However, there is still a possibility of waveform jitter due to missed SEG interrupts (for the same reason related to the SYNC interrupt service routine as discussed earlier). The time at which the SYNC and SEG interrupts are cleared in the SYNC routine depends on the delay between the occurrence of the SYNC interrupt and the subsequent entry into the SYNC routine. If there is no delay, then the instruction causing the interrupt latches to be cleared (MOV *SEG1+, @>C00C) is active between 14.985μs and 27.306μs after the interrupt, and so there is no danger of the SEG interrupt latch being cleared as the next SEG interrupt occurs. However, if a delay of 12μs occurs before entry into the SYNC service routine, the relevant instruction is active between 26.985μs and 39.306μs after the interrupt. Hence, there is a good chance that the SEG interrupt latch might be being cleared as the next SEG interrupt occurs 35.714μs after the SYNC interrupt.

The chance of jitter can be reduced by modifying and rearranging the instructions in the SYNC routine as shown below:

```
SYRN MOV  *LDN1,@>C00C
     MOV  LDN1,SEG1
     INCT SEG1
     RTWP
```

The instruction which outputs the inverter control word and clears the interrupt latches (MOV *LDN1,@>C00C), is now at the start of the routine. Therefore, with no delay the SEG interrupt latch is cleared at some time between 8.991μs and 19.647μs after the SYNC interrupt. If a 12μs delay is incurred, the SEG interrupt latch is cleared at some time between 20.991μs and 31.647μs after the SYNC interrupt. Hence, for the usual range of delays of 0 to 12μs associated with the start of the SYNC interrupt service routine, there is no possibility of the SEG interrupt latch being cleared 35.714μs after the SYNC interrupt. The only drawback with this solution is that the extra instruction which has to be added to the routine (INCT SEG1), causes the execution time of the SYNC routine to be a minimum of 35.964μs. This is fractionally longer than the interval between SEG interrupts at 30000 rpm, but it should not cause any operational problems.

Further increases in motor speed could possibly be achieved by adopting a direct memory access strategy. When a microprocessor receives a DMA request, it enters an "idle" state after completing its present memory cycle and releases control of the system address and data buses. The DMA controlling hardware is then free to read or write into the system RAM. Data transfer by this method can save many machine cycles. A DMA function could be used to transfer data from a memory based look-up table to an output peripheral such as the inverter control port. DMA can be implemented on the TMS9900 system: the $\overline{\text{HOLD}}$ input indicates to the microprocessor that an external controller (such as a DMA device) desires to use the address and data buses; the HOLDA output indicates that the microprocessor has entered a hold state and so the address and data buses are free for

external control. Both these signals are available on the P1 connector of the TM990/100M-1 board ($\overline{\text{HOLD}}$.B at P1.92 and HOLDA.B at P1.86). However, although DMA would have been relatively easy to implement on the system, it was decided not to use DMA, because the primary aim of the microprocessor work in the project was to use "standard" programming techniques to autopilot the motors (that is, interrupt or counter-based position detection).

A rotor-position-code based program was not developed for the 7 phase motor. Having demonstrated that the technique worked successfully on the magslip, it was not felt worthwhile repeating the work on the 7 phase system. If sufficient time had been available, the first priority would have been to improve the interrupt-based software to enable 30000 rpm operation of the 7 phase motor. Subsequent work would then have been directed on the development of a system health monitoring routine, a current limiting algorithm, and a speed control loop.

The work described in this chapter has shown that autopiloted operation of the 7 phase motor at 30000 rpm is within the capability of the TMS9900 system. Real time control of such high speed machines will obviously become technically easier as faster "standard" microprocessors become commercially available. The ease by which the system could be changed from controlling the 3 phase magslip to the 7 phase motor (and vice versa), was an obvious benefit of the flexibility of the microprocessor software. It is certainly true that the autopiloting functions of both the magslip and the 7 phase motor could be implemented readily in hardware. However, many of the features of the described systems would be difficult to achieve without software, and so it is felt that the use of a microprocessor was beneficial with regard to each drive system as a whole.

APPENDIX 2A.

MUIRHEAD MAGSLIP DATA SHEET.

E-17-A/1 3″ TRANSMITTER

(CONTROL TRANSMITTER)

British Services Numbers 6547, 496A

Supply 50V 50c/s Nominal Rotor/Stator Voltages 50 50V

MECHANICAL DATA

BEARINGS	Pre-loaded single row ball journal bearings*	**ACCURACY** MAXIMUM ELECTRICAL ERROR	±0·15 degree
ROTOR CONNEXIONS	Copper graphite brushes Brass slip rings	**MOMENT OF INERTIA OF ROTOR**	5·3 oz.in² · 950gm.cm²
FRICTION TORQUE	Bearings 0·3—0·8 oz.in · 20— 60gm.cm Brushes 0·8—1·6 oz.in · 60—120gm.cm	**WEIGHT**	4 lb · 1·8Kg

Shaft fitted with hub and hub clamp

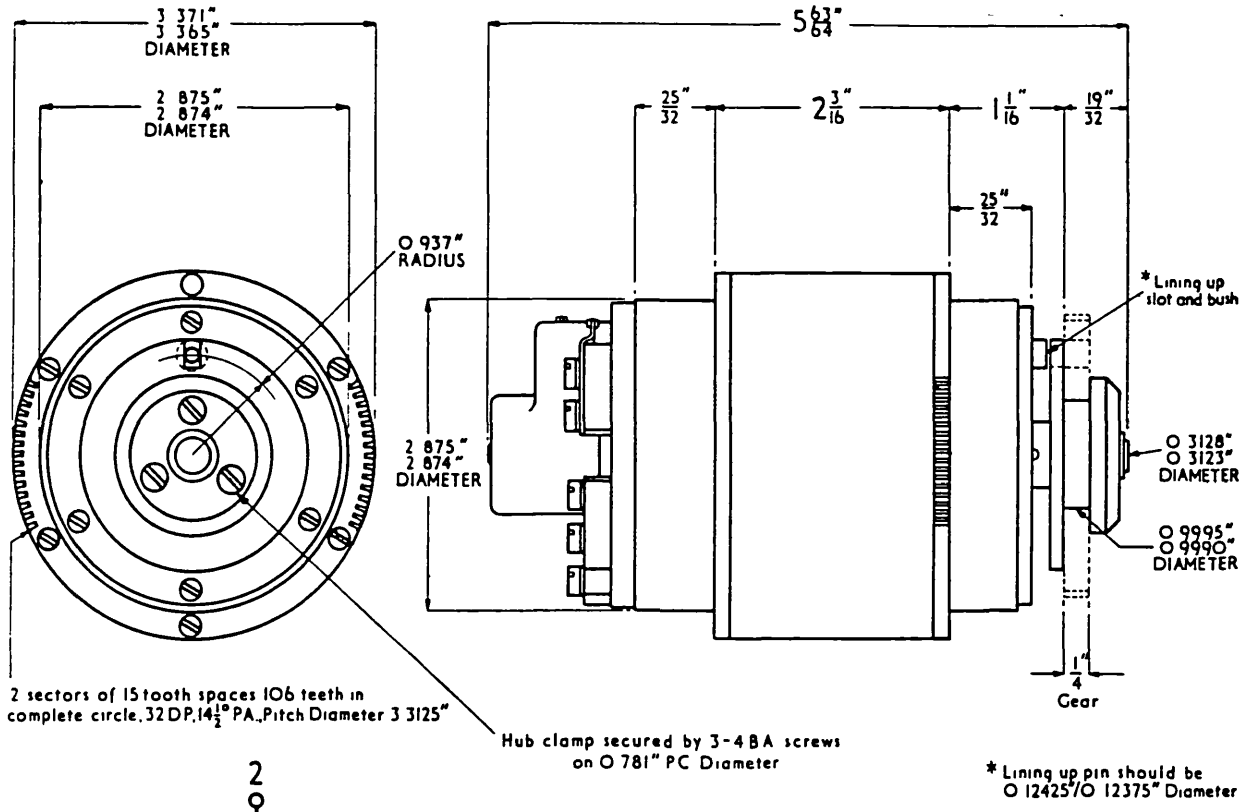
*The use of pre-loaded bearings reduces the audible hum from this instrument to a low level

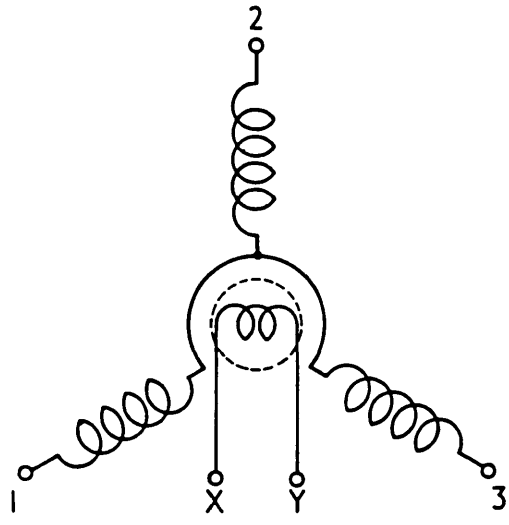
ELECTRICAL DATA

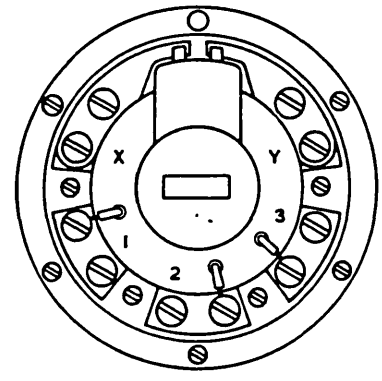
INPUT Rotor		**OUTPUT Stator**	
WINDING	Single phase with cross winding	**WINDING**	3-phase star connected sinusoidally distributed
NO LOAD CURRENT	0·5 A	**VOLTAGE BETWEEN TERMINALS** (No load)	53V maximum
NO LOAD POWER	4·6 W	**IMPEDANCE BETWEEN TERMINALS AT 50V 50c/s**	30 + *j* 140 ohms
IMPEDANCE AT 50V 50c/s	20 + *j* 100 ohms		
D.C. RESISTANCE	10 ohms	**D.C. RESISTANCE BETWEEN TERMINALS**	19 ohms

The accuracy figure above is the Muirhead rejection limit
In general, the accuracy of this instrument will be better than the figure indicates
For all other characteristics, average figures are given

APPENDIX 2A CONTINUED.







E-17-A/1

MOUNTING

The preferred method of mounting consists of inserting the spindle end-cover into a 2·876 in. diameter hole bored in a metal plate or casting. The element is held in position by a clamp ring placed over the opposite end-cover and secured to the mounting plate by four screws. The element must not be clamped around the body.

APPENDIX 2B

The Extraction of the Roots of a Quartic Polynomial

A quartic equation is of the form:

$$t^4 + a_3.t^3 + a_2.t^2 + a_1 t + a_0 = 0 \qquad 2B.1$$

Completing the square for the t^4 and t^3 terms gives:

$$\left(t^2 + \frac{a_3}{2}\, t\right)^2 = \left(\frac{a_3^2}{4} - a_2\right) t^2 - a_1.t - a_0 \qquad 2B.2$$

Adding $2(t^2 + (a_3/2)t)y + y^2$ to both sides of equation 2B.2 gives:

$$\left(t^2 + \frac{a_3}{2}\, t + y\right)^2 = \left(\frac{a_3^2}{4} - a_2 + 2y\right) t^2 + \left(a_3.y - a_1\right)t + y^2 - a_0 \qquad 2B.3$$

The RHS of 2B.3 can be factorised to the form $(\alpha.t + \beta)^2$ if y is chosen correctly. The following processes show how this may be done. Consider the identity:

$$(\alpha.t + \beta)^2 \equiv \alpha^2.t^2 + 2\alpha.\beta.t + \beta^2 \qquad 2B.4$$

Comparing coefficients of the RHS of equations 2B.3 and 2B.4 gives:

$$\alpha^2 \equiv \left(\frac{a_3^2}{4} - a_2 + 2y\right) \qquad 2B.5$$

$$2\alpha\beta \equiv (a_3.y - a_1) \qquad 2B.6$$

and

$$\beta^2 \equiv y^2 - a_0 \qquad 2B.7$$

Thus α and β can be determined providing the value of y is established. The RHS quadratic equation of 2B.3 can be

reduced to a perfect square by making its discriminant zero. Therefore, equating the discriminant to zero gives:

$$(a_3.y - a_1)^2 - 4\left(\frac{a_3^2}{4} - a_2 + 2y\right)(y^2 - a_0) = 0 \qquad 2B.8$$

i.e. $$y^3 - \left(\frac{a_2}{2}\right)y^2 + \left(\frac{a_1.a_3 - 4a_0}{4}\right)y +$$

$$\left(\frac{4a_2.a_0 - a_1^2 - a_0.a_3^2}{8}\right) = 0 \qquad 2B.9$$

This is a cubic polynomial in y which may be written in the general form:

$$y^3 + u_2.y^2 + u_1.y + u_0 = 0 \qquad 2B.10$$

and substituting

$$y = v - (u_2/3) \quad \text{gives:} \qquad 2B.11$$

$$v^3 + q.v + p = 0 \qquad 2B.12$$

where

$$q = \left(\frac{3u_1 - u_2^2}{3}\right) \qquad 2B.13$$

and $$p = \left(\frac{2u_2^3 - 9u_1.u_2 + 27u_0}{27}\right) \qquad 2B.14$$

Equation 2B.10 can be solved by use of the identity:

$$v^3 + s^3 + t^3 - 3v.s.t \equiv (v + s + t)(s + w.v + w^2.t)(s + w^2.v + w.t) \qquad 2B.15$$

where $$w = \frac{-1 + \sqrt{3}i}{2} \qquad 2B.16$$

The roots of the identity for v are:

$$\left.\begin{aligned} v &= -(s + t) \\ v &= -(w.s + w^2.t) \\ v &= -(w^2.s + w.t) \end{aligned}\right\} \qquad 2B.17$$

Equation 2B.12 can be changed into the form of the LHS of identity 2B.15 by the substitutions:

$$q = -3.s.t \qquad 2B.18$$

and $$p = s^3 + t^3 \qquad 2B.19$$

The immediate task is to find s and t in terms of p and q. Now for equation 2B.18,

$$\frac{q}{3s} = -t \qquad 2B.20$$

Substituting 2B.20 into 2B.19 gives

$$s^6 - p.s^3 - \frac{q^3}{27} = 0 \qquad 2B.21$$

Defining $p = 2p_1$ 2B.22

and $q = 3q_1$ 2B.23

Then equation 2B.21 becomes

$$s^6 - 2p_1.s^3 - q_1^3 = 0 \qquad 2B.24$$

i.e. $$s^3 = \frac{2p_1 \pm \sqrt{4p_1^2 + 4q_1^3}}{2} \qquad 2B.25$$

$$s^3 = p_1 \pm (p_1^2 + q_1^3)^{\frac{1}{2}} \qquad 2B.26$$

which gives:

$$s = \left[p_1 \pm (p_1^2 + q_1^3)^{\frac{1}{2}} \right]^{\frac{1}{3}} \qquad 2B.27$$

Equation 2B.18 can then be used to determine t, unless s is zero, when more generally equation 2B.18 can be re-arranged to give

$$-s = \frac{q}{3t} \qquad 2B.28$$

which when substituted into equation 2B.19 leads to

$$t = \left[p_1 \mp (p_1^2 + q_1^3)^{\frac{1}{2}} \right]^{\frac{1}{3}} \qquad 2B.29$$

Therefore, assuming s is the positive root and t is the negative root:

$$s = \left[p_1 + (p_1^2 + q_1^3)^{\frac{1}{2}} \right]^{\frac{1}{3}} \qquad 2B.30$$

and

$$t = \left[p_1 - (p_1^2 + q_1^3)^{\frac{1}{2}} \right]^{\frac{1}{3}} \qquad 2B.31$$

where $p_1 = p/2$ and $q_1 = q/3$ as defined by equations 2B.22 and 2B.23 respectively. Hence v can be found from any of the roots given in 2B.17 since they are cyclically related. At this stage v is known and hence y can be calculated using equation 2B.11;

i.e. $$y = v - (u_2/3) \qquad (2B.11)$$

so $$y = -(s + t) - (u_2/3) \qquad 2B.32$$

Substituting a_2 for u_2 by comparing the coefficients of y^2 in equations 2B.9 and 2B.10 gives

$$y = -(s + t) + (a_2/6) \qquad 2B.33$$

Having chosen y in accordance with equation 2B.8, the RHS of equation 2B.3 is of the form $(\alpha.t + \beta)^2$. Therefore equation 2B.5 gives:

$$\alpha^2 \equiv \left(\frac{a_3^2}{4} - a_2 + 2y\right)$$

i.e. $$\alpha = +\sqrt{\frac{a_3^2}{4} - a_2 + 2y} \qquad 2B.34$$

or $$\alpha = -\sqrt{\frac{a_3^2}{4} - a_2 + 2y} \qquad 2B.35$$

Then using equation 2B.6, $2\alpha\beta = (a_3.y - a_1)$,

i.e. $$\beta = +(a_3.y - a_1)/2\alpha \qquad 2B.36$$

or $$\beta = -(a_3.y - a_1)/2\alpha \qquad 2B.37$$

If α is zero, equation 2B.6 cannot be used and so β is then determined by using equation 2B.7, $\beta^2 = y^2 - a_0$,

i.e. $$\beta = +\sqrt{y^2 - a_0} \qquad 2B.38$$

or $$\beta = -\sqrt{y^2 - a_0} \qquad 2B.39$$

The RHS of equation 2B.3 can therefore be factorised into the form

$$\left[\pm(\alpha.t + \beta)\right]^2 \qquad 2B.40$$

Therefore, if equation 2B.3 is square rooted on both sides, a pair of equations is obtained:

$$(t^2 + \frac{a_3}{2}t + y) = +(\alpha t + \beta) \qquad 2B.41$$

or $$(t^2 + \frac{a_3}{2}t + y) = -(\alpha t + \beta) \qquad 2B.42$$

Equations 2B.41 and 2B.42 are quadratics in t and each produces two roots for t. These are the four roots of the

original quartic polynomial (2B.1). The calculation of the roots can be performed by a computer using complex arithmetic. The equations for p_1 and q_1, (2B.22) and (2B.23), are modified to contain the original coefficients of the quartic. They then have the form:

$$q_1 = \frac{3(a_1 \cdot a_3 - 4a_0) - a_2^2}{36} \qquad 2B.43$$

and

$$p_1 = \frac{9a_1 \cdot a_2 \cdot a_3 + 72a_0 \cdot a_2 - 2a_2^2 - 27a_1^2 - 27a_0 \cdot a_3^2}{432} \qquad 2B.44$$

If all the relevant equations are written in terms of the coefficients of the quartic equation, the required computer program is relatively short as illustrated by the listing in fig. 2B.1. The example program is set up to find the roots of the quartic

$$t^4 + 6t^3 - 151t^2 - 180t + 900 = 0$$

and they are 2, -3, 10, and -15 respectively.

```
      PROGRAM QUARTIC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
      COMPLEX A0,A1,A2,A3,Q,R,G,N,Y,V,W1,W2,AL,BE,S1,S2,S3,S4
C
      PRINT,"PPOGRAM TO SOLVE A QUARTIC EQUATION.",/
C
      A0=(9.0E+02,0.0E+00)
      A1=(-1.8E+02,0.0E+00)
      A2=(-1.51E+02,0.0E+00)
      A3=(6.0E+00,0.0E+00)
C
      Q=((3.0*((A1*A3)-(4.0*A0))-(A2*A2))/36.0)
      R=(((9.0*A1*A2*A3)+(72.0*A0*A2)-(2.0*A2*A2*A2)-(27.0*A1*A1)
     +   -(27.0*A0*A3*A3))/432.0)
C
C
      V=CSQRT((R*R)+(Q*Q*Q))
      W1=(R-V)
      W2=(R+V)
      Z=1.0/3.0
      IF(CABS(W1).LT.1.0E-12)GO TO 1
      G=CEXP(Z*CLOG(W1))
      GO TO 2
    1 G=(0.0E+00,0.0E+00)
    2 IF(CABS(W2).LT.1.0E-12)GO TO 3
      N=CEXP(Z*CLOG(W2))
      GO TO 4
    3 N=(0.0E+00,0.0E+00)
    4 Y=-(G+N)+(A2/6.0)
C
C
      WRITE(6,10)Q,R,G,N,Y
   10 FORMAT(1X,"Q = ",2E20.10,/,1X,"R = ",2E20.10,/,1X,"G = ",2E20.10
     +       /1X,"N = ",2E20.10,/,1X,"Y = ",2E20.10//)
C
      AL=CSQRT(((A3*A3)/4.0)-A2+(2.0*Y))
      IF(CABS(AL).LT.1.0E-12)GO TO 5
      BE=((A3*Y)-A1)/(2.0*AL))
      GO TO 6
    5 BE=CSQRT((Y*Y)-A0)
    6 WRITE(6,20)AL,BE
   20 FORMAT(1X,"AL =",2E20.10,/,1X,"BE =",2E20.10//)
C
C
      WRITE(6,25)
   25 FORMAT(1X,"SOLUTIONS ARE :",/,1X,"-------------",/)
      S1=0.5*(AL-(A3/2.0)+CSQRT(((A3/2.0)-AL)**2)-(4.0*(Y-BE))))
      S2=0.5*(AL-(A3/2.0)-CSQRT(((A3/2.0)-AL)**2)-(4.0*(Y-BE))))
      S3=0.5*(-AL-(A3/2.0)+CSQRT(((A3/2.0)+AL)**2)-(4.0*(Y+BE))))
      S4=0.5*(-AL-(A3/2.0)-CSQRT(((A3/2.0)+AL)**2)-(4.0*(Y+BE))))
      WRITE(6,30)S1,S2,S3,S4
   30 FORMAT(1X,"S1 =",2E20.10,/,1X,"S2 =",2E20.10,/,1X,"S3 =",2E20.10
     +       /1X,"S4 =",2E20.10)
      STOP
      END
```

FIG. 2B.1. EXAMPLE FORTRAN PROGRAM TO SOLVE THE QUARTIC EQUATION : $t^4 + 6t^3 - 151t^2 - 180t + 900 = 0$.

APPENDIX 3A

Magslip Inverter Snubber Design

(a) dv/dt limiting snubber (see fig. 3.36)

During the transistor turn-off period, it is assumed that the inductive load current remains constant and that it transfers from the transistor to the capacitor immediately the switching off procedure is initiated. The "worst case" capacitor value is then chosen so that the snubber capacitor voltage does not rise above the allowable collector-emitter voltage when the maximum load current ($I_{c.cont}$) is flowing. (The maximum collector-emitter voltage was chosen as the maximum rail to rail voltage of the inverter V_{rr}.)

i.e. $$C_v = \frac{I_{load}}{dv/dt} = \frac{I_{c.cont}}{V_{rr}} \times t_f \text{ Farads} \qquad 3A.1$$

where t_f is the fall time of the transistor. The fall time for both the SVT6001 and 2N3773 transistors is 1.0μs and a value of 2.0 amps was chosen for $I_{c.cont}$. The maximum inverter rail to rail voltage is 100 volts and so equation 3A.1 gives a value for C_v of 0.02μF. In practice a value of only 0.01μF was used. This does mean that the dv/dt snubbers were only really capable of protecting the transistors for currents up to 1.0 amp, but the only inverter failure during the experiments was caused by an accidental short circuit on the inverter outputs.

The capacitor C_v is discharged via resistor R_v. During the discharge of C_v the transistor current consists of the load current plus the capacitor discharge current. Since the discharge occurs when the transistor switches on, it can be assumed that following the off period associated with the 120° - 60° magslip waveforms, the current in the inductive load is small or zero. Therefore, R_v must limit

the discharge current to a value I_{dis} that the transistor can support.

$$\text{i.e.} \qquad R_v = \frac{V_{rr}}{I_{dis}} \quad \Omega \qquad \text{3A.2}$$

The power transistors in the magslip inverter can support collector currents of at least 2 amps. I_{dis} was therefore chosen as 1.0 amp to allow up to 1.0 amp of load current to flow at switch on. Equation 3A.2 then gives $R_v = 100\Omega$. The power rating of R_v is then calculated by assuming that C_v discharges its stored energy during each on period. The power P_v dissipated in R_v when the transistor switches f_s times per second is:

$$P_v = \text{Energy stored x switching frequency}$$

$$P_v = 0.5\, C_v.V_{rr}^{2}.f_s \quad \text{Watts} \qquad \text{3A.3}$$

Substituting for C_v from 3A.1:

$$P_v = 0.5.I_{c.cont}.V_{rr}.t_f.f_s \quad \text{Watts} \qquad \text{3A.4}$$

The maximum magslip speed of 12000 rpm is achieved with an inverter frequency of 200Hz. Hence for f_s = 200Hz, equation 3A.4 gives P_v = 20mW. Therefore a 100Ω 0.5W carbon resistor was chosen for R_v. The time constant $R_v.C_v$ is 1μs and so C_v can fully discharge in between the transistor switch on operations.

(b) <u>di/dt limiting snubber</u> (see fig. 3.36)

During the transistor turn on period, it is assumed that the load voltage remains constant and that the transistor voltage drops immediately to zero volts. The "worst case" inductance value is then chosen so that during the specification switching time t_r, the transistor current does not rise above the value that the transistor can sup-

port in saturation. The maximum voltage that can appear across L_i during the switch on operation is the maximum inverter rail-to-rail voltage V_{rr}. Hence:

$$L_i = \frac{V}{di/dt} = \left(\frac{V_{rr}}{I_{c.cont}}\right).\ t_r \text{ Henries} \qquad 3A.5$$

The SVT6001 and 2N3773 transistors have switch on times of about 1.0μs (SVT6001 = 0.4μs; 2N3773 = 1.5μs). Equation 3A.5 gives a value of 50μH for L_i with $I_{c.cont}$ = 2.0 amps. In practice, 9.5 turns of adequately rated wire could be comfortably wound on an RM10 ferrite core. This results in an inductance of 36μH and this was felt to be adequate.

R_i is chosen so that at switch off the extra voltage component superimposed onto the transistor collector by the decaying inductor current does not rise above a value V_{decay}.

i.e. $$R_i = \frac{V_{decay}}{I_{c.cont}} \quad \Omega \qquad 3A.6$$

A value for R_i of 1.0Ω was selected since this limits V_{decay} to only 2 volts for values of $I_{c.cont}$ up to 2.0 amps.

The power P_i dissipated in R_i when the transistor switches f_s times a second is:

P_i = Energy stored x switching frequency

$$P_i = 0.5.L_i.I_{c.cont}^{2}.f_s \text{ Watts} \qquad 3A.7$$

Substituting for L_i from 3A.5:

$$P_i = 0.5.V_{rr}.I_{c.cont}.t_r.f_s \text{ Watts} \qquad 3A.8$$

Hence for f_s = 200Hz, equation 3A.8 gives a value of 20mW

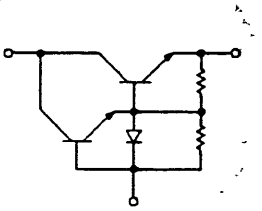
for P_i. A supply of 1Ω 2W carbon resistors were available and so were used for the R_i components. The time constant L_i/R_i is 36μs and this is short enough to ensure that the inductor current decays to zero during the transistor off time.

APPENDIX 3B.

TRANSISTOR DATA SHEET.

Monolithic Darlington

SVT6000 series -500 Volts, 20Amps, 400nsec For use in: Automotive Ignition Systems, High Gain Power Conditioning Circuits and TV Sweep Circuits



The SVT6000 Series are NPN, high-voltage, high-gain, Monolithic Darlington amplifiers for use in high-speed power circuits High I_{SB} (resistance to forward secondary breakdown) has been diffused into the device family. The units are mounted in the E I A standard TO-3 package A rapid turnoff diode and resistor network are integral in the package

Maximum Ratings (T_{CASE} = 25°C)

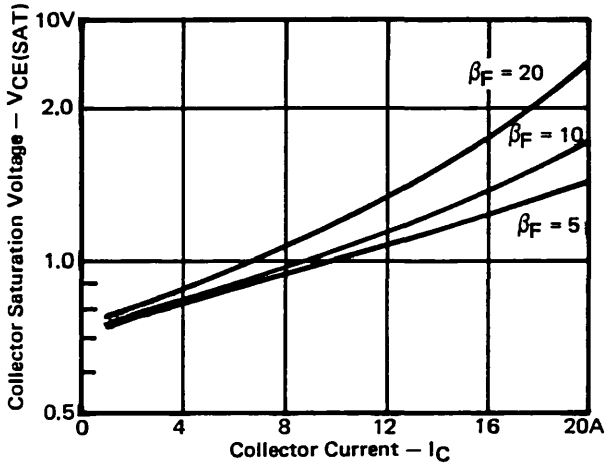
Symbol	Characteristics	SVT6000	SVT6001	SVT6002
V_{CBO}	Collector-Base Voltage	400Vdc	450Vdc	500Vdc
V_{CEO}	Collector-Emitter Voltage	400Vdc	450Vdc	500Vdc
$I_{C(CONT)}$	Collector Current	15A	15A	15A
$I_{C(PEAK)}$	Collector Current	20A	20A	20A
P_T	Power Dissipation	96W	96W	96W
E_{SB}	Energy (see Fig. 2)	70mJ	70mJ	70mJ
T_j	Junction Temperature	-50°C to +150°C		
θ_{j-c}	Thermal Resistance	1.3°C/Watt		

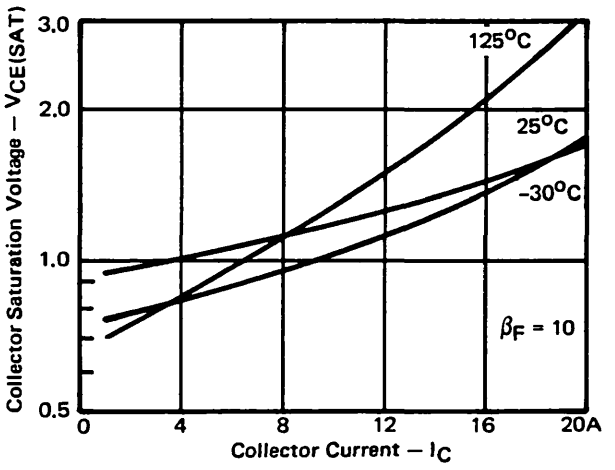
Electrical Characteristics (T_{CASE} = 25°C)

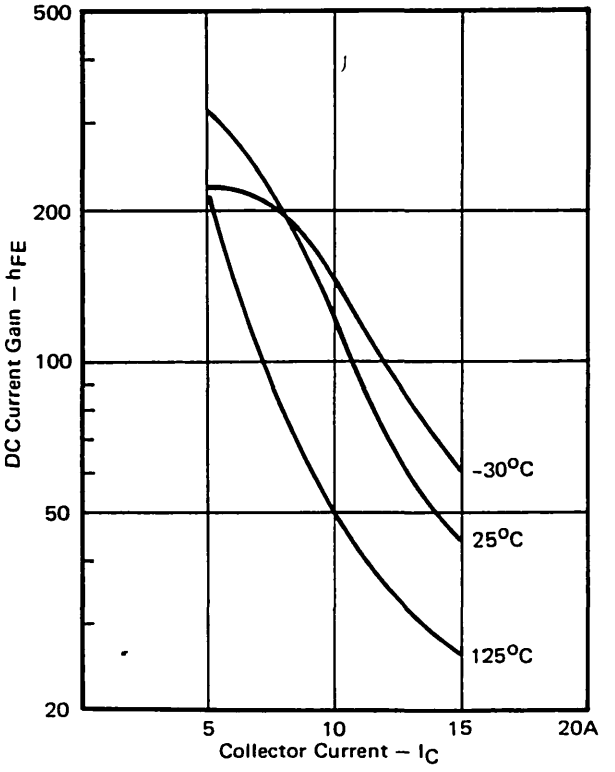
Symbol	Characteristics	Test Conditions	SVT6000		SVT6001		SVT6002	
			Min	Max	Min	Max	Min	Max
I_{CEO}	Collector-Emitter Leakage	V_{CE} = 400V		1mA				
		V_{CE} = 450V				1mA		
		V_{CE} = 500V						1mA
$V_{CEO(SUS)}$	Sustaining Voltage	I_C = 2A L = 1mH (see Fig. 2)	300V		350V		400V	
$V_{CE(SAT)}$	Collector Saturation Voltage	I_C = 15A, I_B = 1.5A I_C = 10A, I_B = 1A		2.0V 1 5V		2.0V 1.5V		2.0V 1.5V
$V_{BE(SAT)}$	Base Saturation Voltage	I_C = 10A, I_B = 1A		2.5V		2.5V		2.5V
h_{FE}	D.C. Current Gain	I_C = 5A, V_{CE} = 5V I_C = 10A, V_{CE} = 5V	150 60		150 60		150 60	
t_r	Rise Time	I_C = 10A, I_{B1} = 1.0A		0.4μS		0.4μS		0.4μS
t_s	Storage Time	I_{B2} = 1.5A		2.5μS		2 5μS		2.5μS
t_f	Fall Time	(see Fig. 1)		1.0μS		1 0μS		1.0μS

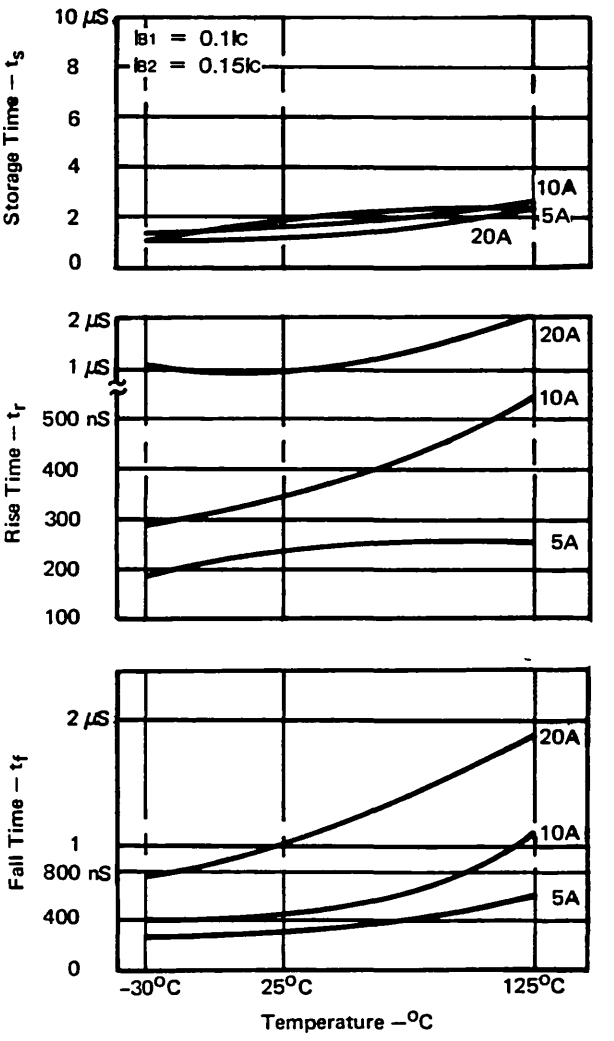
APPENDIX 3B CONTINUED.

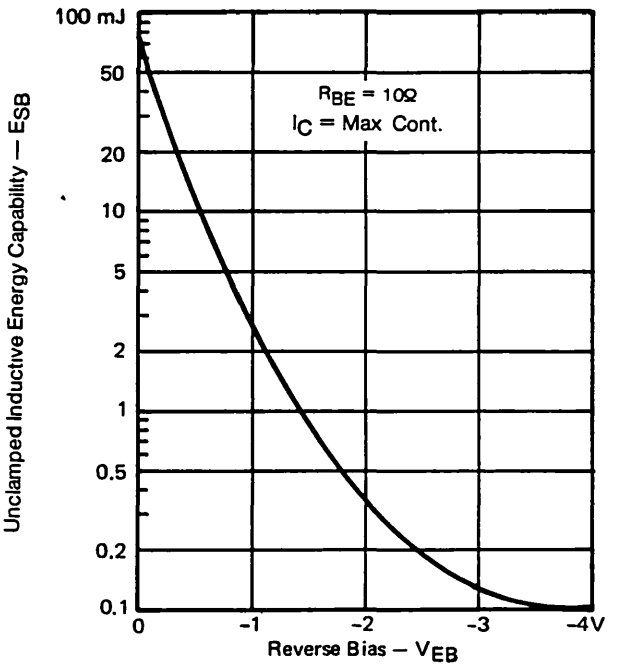
SVT 6000 SERIES



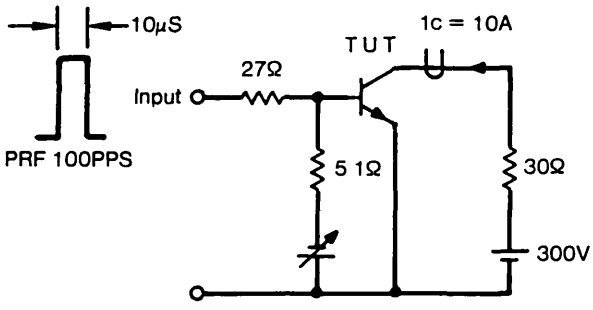








APPENDIX 3B CONTINUED.

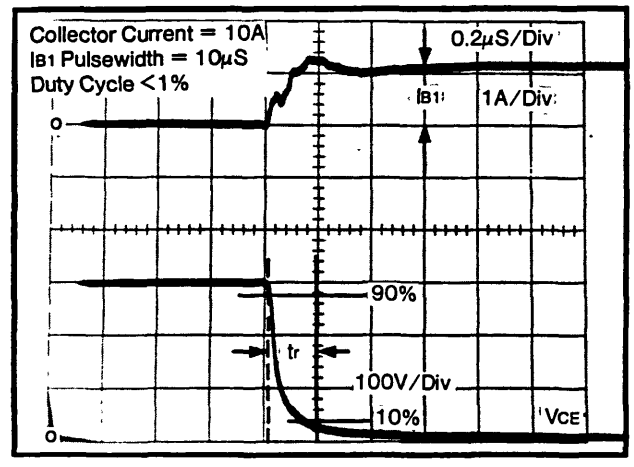


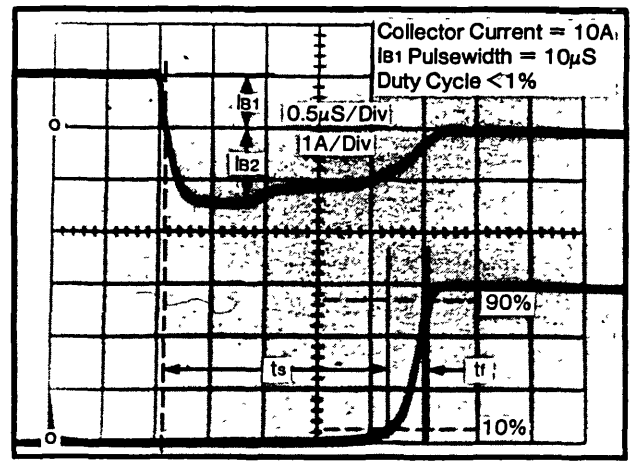
Adjust pulse amplitude to give I_{B1} = 1A

Adjust negative bias to give I_{B2} = 1 5A

Figure 1. Switching Time Test Circuit

Typical Switching Time Waveforms





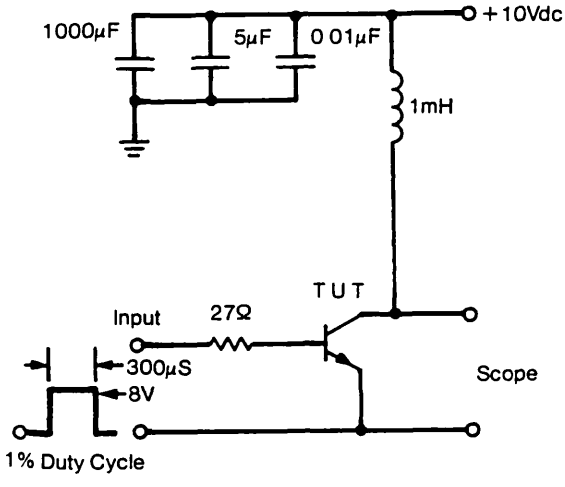
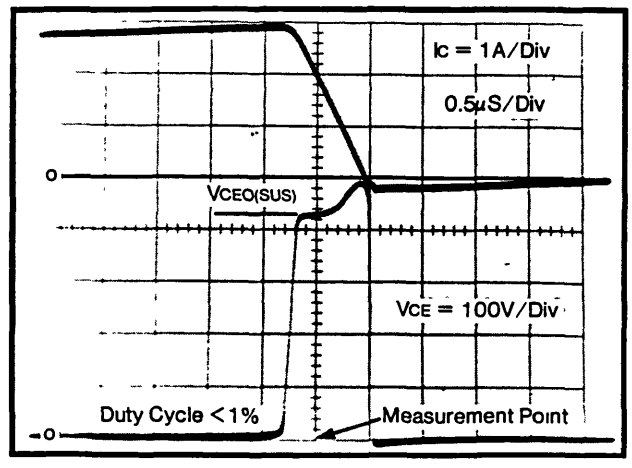


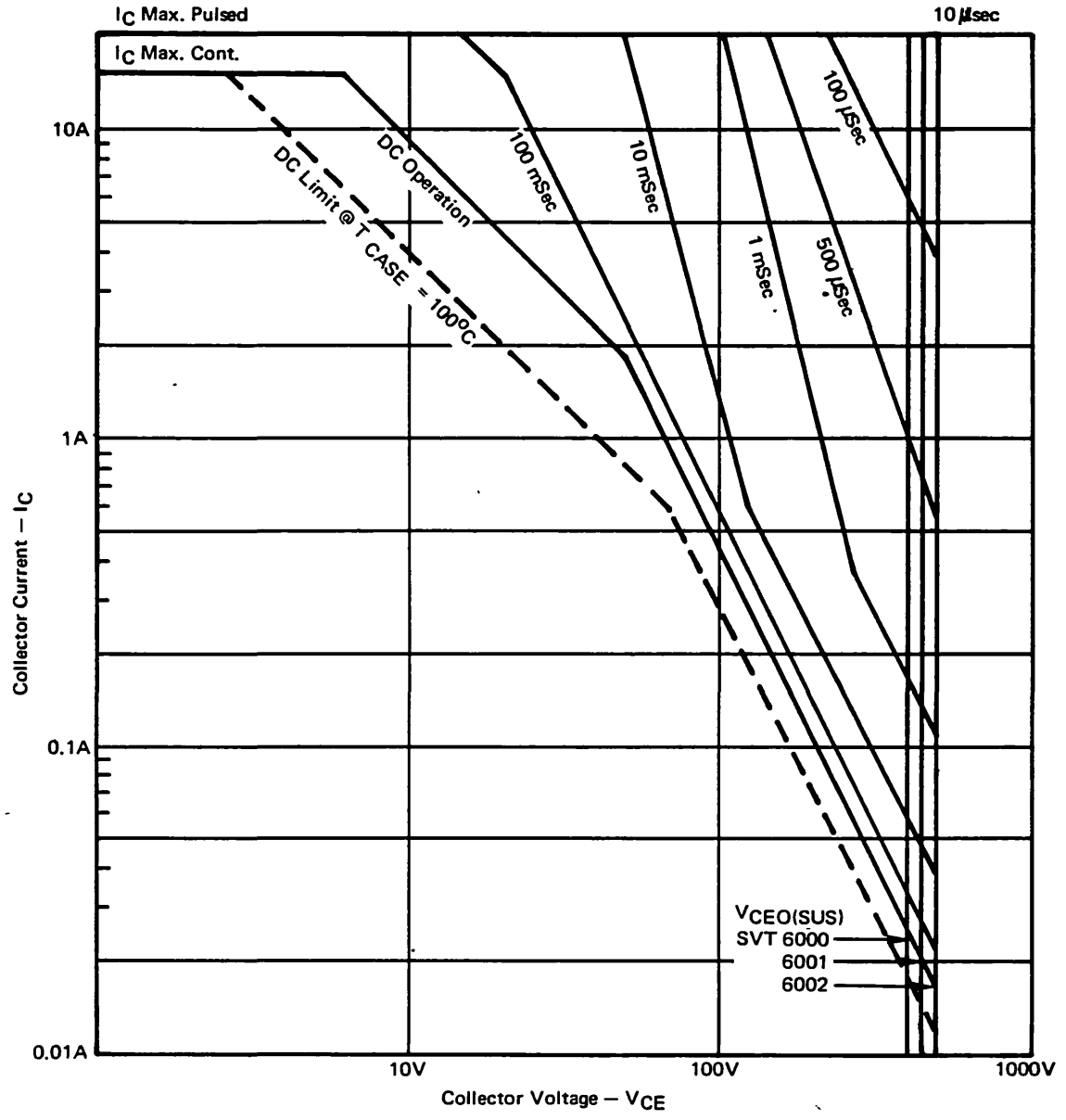
Figure 2. Sustaining Test Circuit

Typical Sustaining Waveform



APPENDIX 3B CONTINUED.

SAFE AREA CURVE (T_{CASE} = 25°C)
SVT 6000 SERIES



SAFE OPERATION

These devices have been designed for optimum performance in switching applications. The curves indicate the second breakdown capability for single non-repetitive pulses. For case temperatures above 25°C suitable derating based on predicted junction temperatures must be applied.

$$T_j = T_c + P_T \times \theta_{j\text{-}c}$$

e.g., for a device dissipation of 35W at a case temperature of 100°C

$$T_j = 100 + 35 \times 1.3$$
$$= 145.5°C$$

This is a safe operating condition since the maximum junction temperature limit is 150°C.

APPENDIX 3C.

TOSHIBA

TECHNICAL DATA

TOSHIBA TRANSISTOR

2 N 3 7 7 3

SILICON NPN TRIPLE DIFFUSED TRANSISTOR

General Purpose Power Transistor

Power Regulator, Power Switching

DC-DC Converter and Solenoid Drives

This Device Features

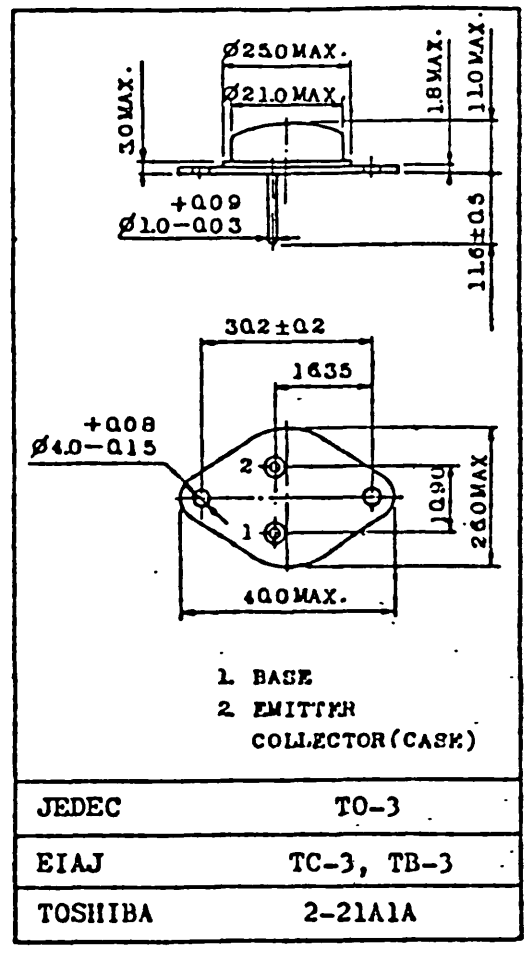
High Collector Power Dissipation

: P_C=150W (Tc=25°C)

High Collector Current : I_C=16A

Lov Saturation Voltage : $V_{CE(sat)}$=0.4V (Typ.)
(I_C=8A)

Unit in mm



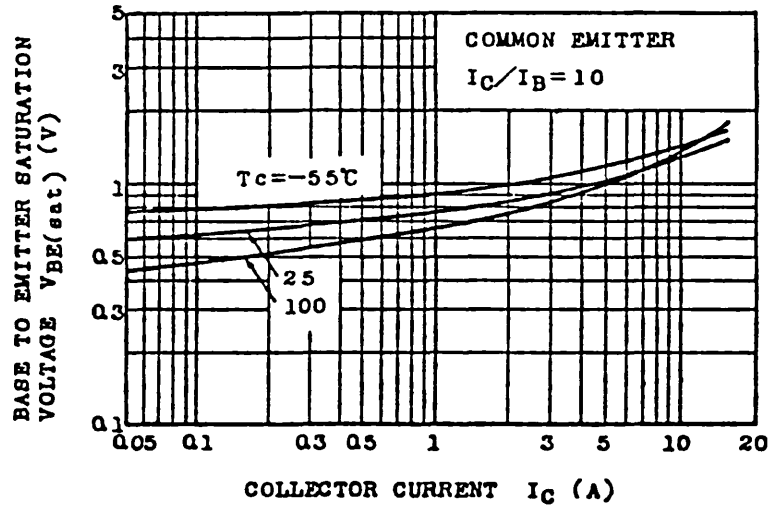
JEDEC	TO-3
EIAJ	TC-3, TB-3
TOSHIBA	2-21A1A

MAXIMUM RATINGS (Tc=25°C)

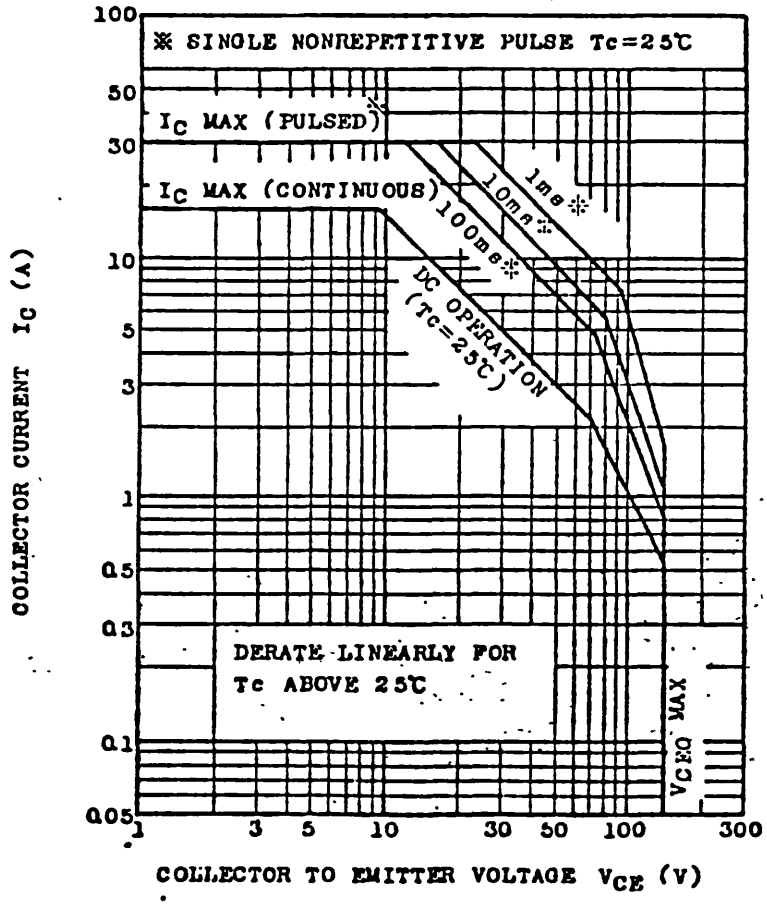
CHARACTERISTIC	SYMBOL	RATING	UNIT
Collector to Base Voltage	V_{CBO}	160	V
Collector to Emitter Voltage	V_{CEO}	140	V
Emitter to Base Voltage	V_{EBO}	7	V
Collector Current	I_C	16	A
Base Current	I_B	4	A
Collector Power Dissipation (Tc=25°C)	P_C	150	W
Junction Temperature	T_j	200	°C
Storage Temperature	T_{stg}	-65 ~ 200	°C

APPENDIX 3C CONTINUED.

$V_{BE(sat)}$ – I_C

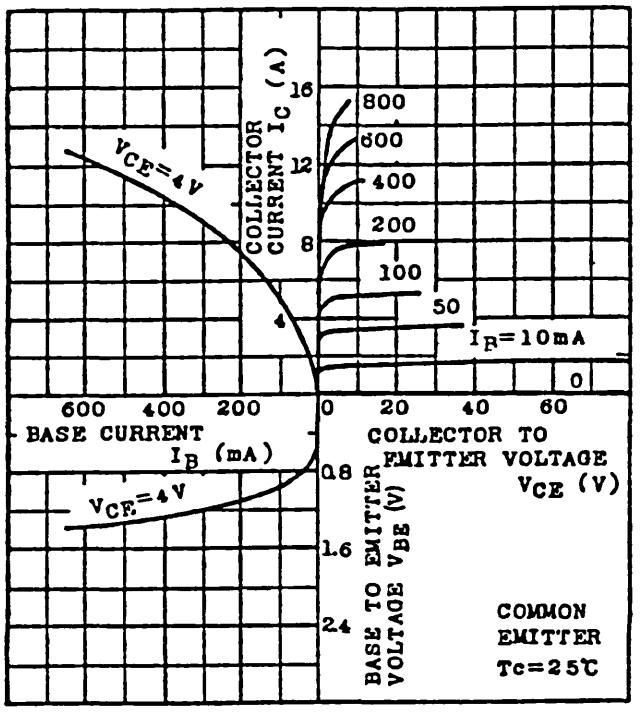


ASO

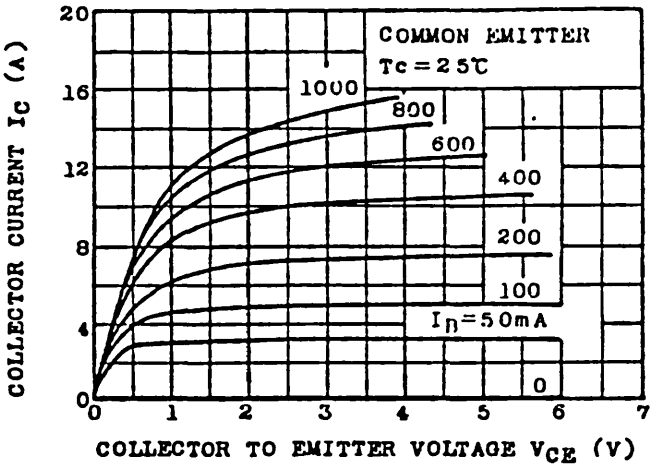


APPENDIX 3C CONTINUED.

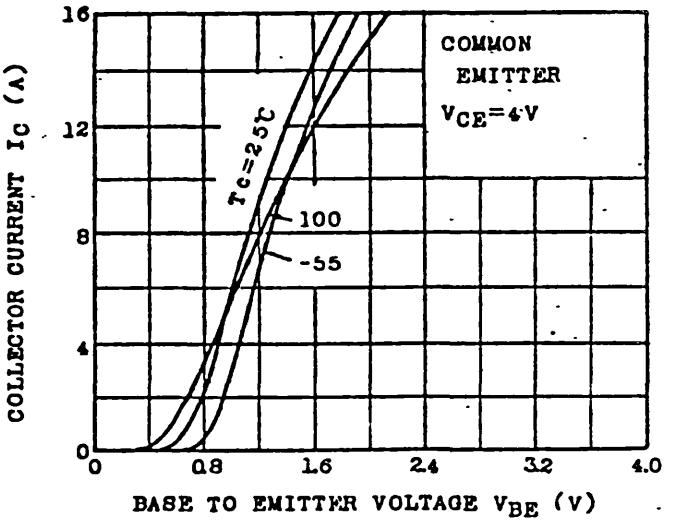
STATIC CHARACTERISTICS



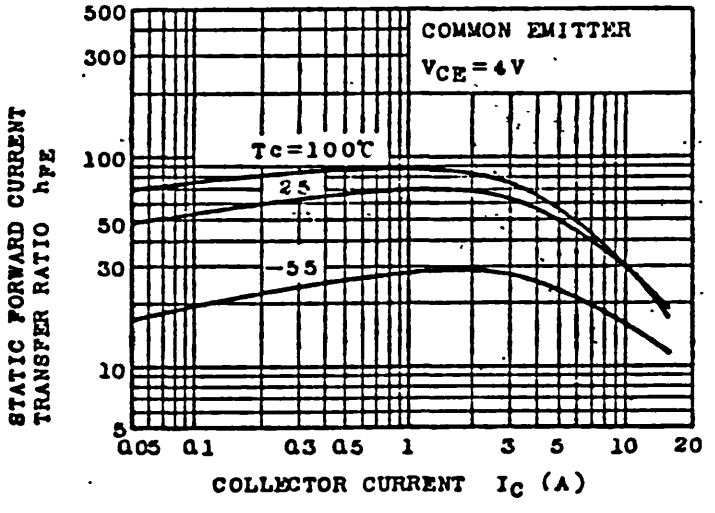
$I_C - V_{CE}$ (LOW VOLTAGE REGION)



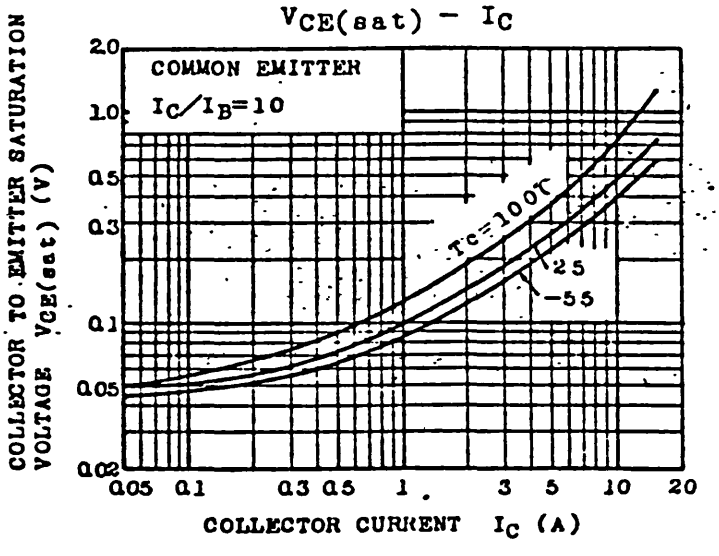
$I_C - V_{BE}$



$h_{FE} - I_C$



$V_{CE(sat)} - I_C$



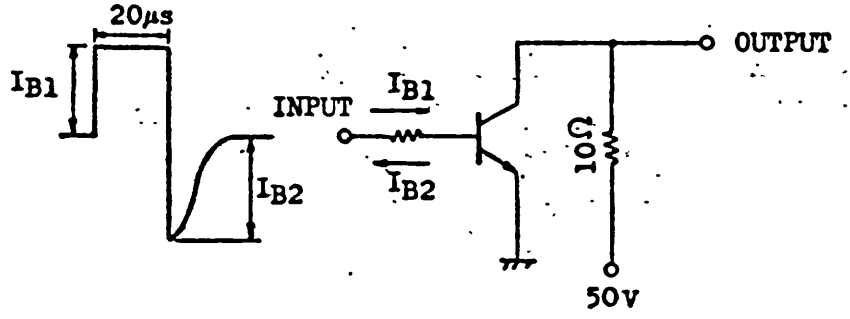
APPENDIX 3C CONTINUED.

ELECTRICAL CHARACTERISTICS (T_c=25°C)

CHARACTERISTIC		SYMBOL	CONDITION	MIN.	TYP.	MAX.	UNIT
Collector Cut-off Current		I_{CBO}	V_{CB}=140V, I_E=0	-	-	100	μA
Emitter Cut-off Current		I_{EBO}	V_{EB}=7V, I_C=0	-	-	100	μA
Collector to Emitter Breakdown Voltage		$V_{(BR)CEO}$	I_C=200mA, I_B=0	140	-	-	V
Static Forward Current Transfer Ratio		$h_{FE(1)}$	V_{CE}=4V, I_C=8A (Note)	15	-	60	
		$h_{FE(2)}$	V_{CE}=4V, I_C=16A (Note)	5	-	-	-
Collector to Emitter Saturation Voltage		$V_{CE(sat)}$	I_C=8A, I_B=0.8A (Note)	-	0.4	1.4	V
Base to Emitter Voltage		V_{BE}	V_{CE}=4V, I_C=8A (Note)	-	1.2	2.2	V
Gain Bardwidth Product		f_T	V_{CE}=4V, I_C=1A	-	3	-	MHz
Output Capacitance		C_{ob}	V_{CB}=10V, I_E=0 f=1MHz	-	350	-	pF
Switching Time	Turn-On Time	t_{on}	Fig. 1	-	1.5	-	μs
	Storage Time	t_{stg}		-	6	-	μs
	Fall Time	t_f		-	1	-	μs

Note : Pulse Test:Pulse Width≦300μs Duty Cycle≦2%

Fig.1 : SWITCHING TIME TEST CIRCUIT



I_{B1}=-I_{B2}=0.5A

DUTY CYCLE≦1%

SOURCE LISTING OF THE MAGSLIP SYSTEM INTERRUPT-BASED CONTROL PROGRAM.

```
0010 * PROGRAM TO RUN THE MAGSLIP SYSTEM USING
0020 * INTERRUPTS TO MAINTAIN SYNCHRONISM.
0030 * NOVEMBER 20, 1981. N.M. MALLINSON.
0040 *
0050 *
0060 *** THIS PROGRAM SHOULD BE RUN ON A PERKIN ELMER VDU
0070 *** (PREFERABLY WITH 9600 BAUD RATE).
0080 *
0090 *
0100        IDT  'MAGSLP'
0110        AORG >1040
0120 * DEFINE REGISTER LABELS.
0130 R0     EQU  0
0140 R1     EQU  1
0150 R2     EQU  2
0160 R3     EQU  3
0170 R4     EQU  4
0180 R5     EQU  5
0190 R6     EQU  6
0200 R7     EQU  7
0210 R8     EQU  8
0220 R9     EQU  9
0230 R10    EQU  10
0240 R11    EQU  11
0250 R12    EQU  12
0260 R13    EQU  13
0270 R14    EQU  14
0280 R15    EQU  15
0290 MAG    LWPI >1000
0300 *
0310        CLR  @ESC          **CLEAR VDU FLAG.
0320 *
0330 * SET ALL IMPORTANT OUTPUT PORTS SAFE.
0340        CLR  @>C00E        VOLTS.
0350        SETO @>C00C        INVERTER CONTROL.
0360        CLR  @>C010        INVERTER SELECT.
0370        CLR  @>C006        DISPLAY PORT.
0380        CLR  @>C012           "      "
0390        CLR  @>C012           "      "
0400 *
0410 *
0420 **VECTOR VARI DATA ADDRESSES FOR DISPLAY ROUTINE.
0430        LI   R1,M0
0440        AI   R1,20
0450        LI   R0,VARI
0460        MOV  R0,*R1+       ***DATA VECTOR
0470        INCT R0
0480        CI   R1,MORE       END OF ADDRESS BLOCK.
0490        JLT  $-8           ***DATA VECTOR
0500 *
0510 *
0520 * VECTOR INPUT, READ,WRITE,
0530 * MODIFY & HEX MOD XOP'S.
0540        DXOP INPU,4
0550        LI   R0,>25A0      XOP4 WP.
0560        MOV  R0,@>0050
0570        LI   R0,IXOP       XOP4 PC.
0580        MOV  R0,@>0052
0590        DXOP READ,5
0600        LI   R0,>2520      XOP5 WP.
0610        MOV  R0,@>0054
0620        LI   R0,RXOP       XOP5 PC.
0630        MOV  R0,@>0056
0640        DXOP WRIT,6
0650        LI   R0,>2540      XOP6 WP.
0660        MOV  R0,@>0058
0670        LI   R0,WXOP       XOP6 PC.
0680        MOV  R0,@>005A
0690        DXOP MOD,0
0700        LI   R0,>2560      XOP0 WP.
0710        MOV  R0,@>0040
0720        LI   R0,MXOP       XOP0 PC.
0730        MOV  R0,@>0042
0740        DXOP HEX,1
0750        LI   R0,>2580      XOP1 WP.
0760        MOV  R0,@>0044
0770        LI   R0,HXOP       XOP1 PC.
0780        MOV  R0,@>0046
0790 *
0800        CLR  @BANN+4       **MESSAGE IDENTIFIER.
0810        BLWP @BANN         WRITE VDU TITLE.
0820        WRIT @G+2          **GO? MESSAGE.
0830        READ @G
0840        BLWP @CLER         CLEAR SCREEN.
0850        WRIT @QUES         PERIPHERAL STATUS QUESTIONS.
0860        WRIT @CHK1
0870        READ @SYMY
0880        WRIT @CHK2
0890        READ @SYMY
0900        WRIT @CHK3
0910        READ @SYMY
0920 OPT    BLWP @CLER         CLEAR SCREEN.
0930        WRIT @VAR          IS CHANGE IN LIMIT VALUES WANTED?
0940        READ @CHOS         YES/NO CHOICE.
0950        LI   R0,CHOS
0960        C    @RDAT,R0
0970        JNE  $+22          ***CONTINUE
```

```
0980        WRIT @VTX1
0990        WRIT @NOTE
1000        WRIT @VTX2
1010        READ @VARD
1020        MOD  @RDAT
1030 * CLEAR SCREEN.
1040        BLWP @CLER          ****CONTINUE
1050        WRIT @HQES          CHANGE RECALL MEMORY LOCATIONS?
1060        READ @CHOS
1070        LI   R0,CHOS
1080        C    @RDAT,R0
1090        JNE  $+10           ****KEYPAD
1100        WRIT @HXCR
1110        HEX  @M0            MODIFY DATA AT LOCATION M0-M9.
1120        WRIT @KEYP          ****KEYPAD
1130        READ @CHOS
1140        C    @RDAT,R0
1150        JNE  OPT
1160        LI   R0,4
1170        MOV  R0,@BANN+4     **MESSAGE IDENTIFIER.
1180        BLWP @BANN
1190 *
1200 *
1210 ********************
1220 * MAIN MOTOR PROGRAM.
1230 * COMPLETE INITIALISATION OF SYSTEM.
1240 *
1250 * VECTOR 9900 INTERRUPTS 1,2,8,9,10.
1260 * INTERRUPTS ARE (9900 # IN BRACKETS):
1270 * INT1=SYNC (1).
1280 * INT2=SEGI (2).
1290 * INT6=STOP (8).
1300 * INT7=DATA (9).
1310 * INT8=DISP (10).
1320 *
1330 * THE WORKSPACES FOR THE MAIN PROGRAM, SYNC, SEG & STOP
1340 * ARE OVERLAPPED TO EASE DATA TRANSFER.
1350        LI   R1,INFO
1360        LI   R2,ADDR
1370        MOV  *R2+,R3        ****VECTOR
1380        MOV  *R1+,*R3
1390        MOV  *R1,*R1        COMPARE TO ZERO.
1400        JNE  $-6            ****VECTOR
1410        JMP  CLRI
1420 * INFORMATION ARRANGED IN FORMAT WP1;PC1;WP2;PC2; ETC..
1430 *
1440 INFO   DATA >1012,SYST,>100C,SEGI
1450        DATA >1000,STOP,>25E0,DATA,>2600,DISP
1460        DATA 0              ***END OF VECTORING.
1470 *
1480 * ADDRESSES IN FORMAT INT1 WP STORE,
1490 * INT1 PC STORE, ETC..
1500 ADDR   DATA >0004,>0006,>0008,>000A
1510        DATA >0020,>0022,>0024,>0026,>0028,>002A
1520 *
1530 CLRI   CKON                CLEAR ALL INTERRUPT LATCHES.
1540        CKOF
1550        LI   R1,>C014
1560        CLR  *R1+           ****WIPE
1570        CI   R1,>C020       STOP VALUE.
1580        JLT  $-6            ****WIPE
1590 *
1600 * SET UP INTERRUPTS.
1610 * (DO NOT ENABLE SYNC UNTIL START REQUESTED).
1620 * (DO NOT ENABLE SEG UNTIL SYNC ACHIEVED).
1630        LI   R12,>100       CRU BASE.
1640        LI   R1,>700        ENABLE 9900 INT'S 8,9,10.
1650        LDCR R1,11          SET INT MODE.
1660 *
1670 * SET DEMANDED VOLTS TO START VALUE & OTHER
1680 * VARIABLE INPUTS TO ZERO FOR SAFETY.
1690        MOV  @VARL+2,@VARI
1700        LI   R1,8
1710        CLR  @VARI+2(R1)    ****CLEAR
1720        DECT R1
1730        JLT  $+4            ****CLRFIN
1740        JMP  $-8            ****CLEAR
1750 *
1760 *
1770 * MAIN PROGRAM.
1780 VRQN   EQU  R1
1790 VRQO   EQU  R2
1800 DANW   EQU  R3
1810 DAOL   EQU  R4
1820 LRQN   EQU  R5
1830 LRQO   EQU  R6
1840 CRUB   EQU  R12
1850 LDAN   EQU  R10
1860 SEG    EQU  R11
1870 * INITIALISE KEYPAD, INVERTER & PSU CRUBIT.
1880        LI   CRUB,>120      ****CLRFIN
1890        SBZ  8
1900 *
1910        CLR  VRQN
1920        CLR  VRQO
1930        CLR  DANW
1940        CLR  DAOL
1950        CLR  LRQN
```

```
1960        CLR  @SLOW
1970        CLR  @>C00E
1980 *
1990 * NOW WAIT FOR STOP, DATA OR DISP INT.
2000 REST   C    @NINE,@ESC
2010        JNE  $+20          ****NOT VDU
2020        LIMI 0
2030 * DISABLE INVERTER.
2040        LI   CRUB,>120
2050        SBO  8
2060        WRIT @BELL         LOOK AT VDU!!!
2070        B    @MAG
2080        LIMI 10            ****NOT VDU
2090 *
2100        JMP  REST          WAIT FOR START INT.
2110 * CHECK REQUIRED DIRECTION OF ROTATION.
2120 AWAY   TB   9              CRUBIT >129.
2130        JNE  $+8           ***FWD
2140 * REVERSE SELECTED.
2150        LI   LDAN,DRVR
2160        JMP  $+6           ****HOP
2170        LI   LDAN,DRVF     ****FWD
2180        MOV  LDAN,SEG      ****HOP
2190        MOV  @VARL+2,VRQO
2200        MOVB @VTAB(VRQO),DAOL
2210        SRL  DAOL,8
2220        AI   DAOL,>6000
2230        MOV  DAOL,@>C00E   SET START VOLTS.
2240        CLR  LRQO
2250 * CHECK MOTOR IS STATIONARY PRIOR TO START.
2260        MOV  @>C000,R8     ****SPEED
2270        CI   R8,16
2280        JGT  $-8           ****SPEED
2290 *
2300        LI   R0,30         MAX # OF STARTING STEPS.
2310        DEC  R0            ****NX
2320        JEQ  ABOR          ABORT START.
2330        LI   R8,>7FFF
2340        DEC  R8            ****DELAY
2350        JGT  $-2           ****DELAY
2360 * CHECK IF STOP REQUIRED.
2370        MOV  @SLOW,@SLOW
2380        JNE  BRAK
2390        MOV  *SEG+,@>C00C
2400 * TEST FOR NEGATIVE NUMBER: IF SO MSB IS SET
2410 * AND TABLE POINTER REQUIRES RESETTING.
2420        JGT  $-24          ****NX
2430        AI   SEG,-24
2440        JMP  $-30          ****NX
2450 *
2460 ** ABORT START: DISABLE INVERTER.
2470 ABOR   LI   CRUB,>120
2480        SBO  8
2490        SETO @>C00C
2500        CLR  @>C00E
2510        BLWP @CLER
?
2520        MOV  @SLOW+10,@SLOW+10 STALL DETECT FLAG.
2530        JEQ  $+12          ****START FAILURE
2540        WRIT @STAL
2550        CLR  @SLOW+10      STALL DETECT FLAG.
2560        JMP  $+6           ****GO? MESSAGE
2570        WRIT @FAIL         ****START FAILURE
2580        WRIT @G+2          ****GO? MESSAGE
2590        READ @G
2600        B    @OPT
2610 *
2620 SYPL   LI   R8,SYRN       SYNC PULSE RECEIVED.
2630        MOV  R8,@>0006     VECTOR SYNC FOR NORMAL RUN.
2640 *
2650 MAIN   LI   R0,>FFF       STALL DETECT DELAY.
2660        MOV  R0,@SLOW+8    STALL DETECT DELAY STORE.
2670        MOV  @SLOW,@SLOW   ****STOP INT?
2680        JEQ  INPT
2690 * CLEAR VOLTS AND WAIT FOR SYSTEM TO RAMP DOWN.
2700        CLR  VRQN
2710        CI   DAOL,>6000    ZERO VOLTS.
2720        JGT  DIFF
2730 BRAK   LIMI 0
2740        CLR  @SLOW
2750        CLR  @>C00E
2760        SETO @>C00C
2770        LI   SEG,SYST
2780        MOV  SEG,@>0006    VECTOR SYST INT.
2790 *
2800 **NOTE STOP/START FLIPFLOP IN KEYPAD MAY BE
2810 **SET TO START WHILST MOTOR IS STOPPING: TO AVOID
2820 **TROUBLE WHEN KEY NEXT PRESSED (GENERATING A
2830 **STOP) CLEAR FLIPFLOP AT END OF STOP.
2840        LI   CRUB,>120
2850        SBO  8
2860        SBZ  8
2870        CLR  @>C01A        CLEAR STOP INT.
2880        LI   CRUB,>100
2890        LI   SEG,>700
2900        LDCR SEG,11        MASK SYNC & SEG INTS.
```

```
2910        LI   CRUB,>120
2920 * SET INITIAL RUNNING VOLTS TO START VALUE.
2930 * (CAN BE CHANGED THOUGH DURING STOPPED TIME).
2940        MOV  @VARL+2,@VARI
2950        JMP  REST
2960 *
2970 INPT   MOV  @VARI,VRQN   INPUT VOLTS.
2980 DIFF   C    VRQN,VRQO
2990        JEQ  ANG
3000 * CHECK IF VRQN IS IN LIMITS.
3010        C    VRQN,@VARL
3020        JLT  $+8          ****OKAY
3030        JEQ  $+6          ***OKAY
3040        MOV  @VARL,VRQN
3050        MOVB @VTAB(VRQN),DANW ****OKAY
3060        SRL  DANW,8
3070        AI   DANW,>6000
3080        C    DANW,DAOL
3090        JEQ  UPDA         UPDATE REGISTERS.
3100        JGT  INCR
3110 *DECREMENT VOLTS.
3120        DEC  @SLOW+4      ****DOWN
3130        JGT  ANG
3140        DEC  DAOL
3150        MOV  @VARL+8,@SLOW+4 ****DOWN
3160        MOV  DAOL,@>C00E  OUTPUT NEW VOLTS.
3170        JMP  ANG
3180 INCR   DEC  @SLOW+6      ****UP
3190        JGT  ANG
3200        INC  DAOL
3210        MOV  @VARL+8,@SLOW+6 ****UP
3220        MOV  DAOL,@>C00E  OUTPUT NEW VOLTS.
3230        JMP  ANG
3240 UPDA   MOV  DANW,DAOL
3250        MOV  VRQN,VRQO
3260 * LOAD ANGLE CHANGE SECTION.
3270 ANG    MOV  @VARI+2,LRQN INPUT LOAD ANGLE.
3280        C    LRQN,LRQO
3290        JEQ  BOT          CHECK FOR STALL.
3300        C    LRQN,@VARL+4 MAX LOAD ANGLE.
3310        JLT  $+8          ****FINE
3320        JEQ  $+6          ***FINE
3330        MOV  @VARL+4,LRQN
3340        S    LRQO,LRQN    ****FINE
3350        CI   LRQN,15
3360        JLT  FALL
3370        INCT LDAN
3380        AI   LRQO,30
3390        JMP  BOT          CHECK FOR STALL.
3400 FALL   CI   LRQN,-15
3410        JGT  MAIN         CHECK FOR STALL.
3420        DECT LDAN
3430        AI   LRQO,-30
3440 BOT    MOV  @SLOW+8,@SLOW+8 STALL DELAY COUNT.
3450        JEQ  $+10         ****CHECK STALL
3460        DEC  @SLOW+8
3470        B    @MAIN+8      ****STOP INT?
3480        MOV  @>C000,R0    ****CHECK STALL
3490        CI   R0,16
3500        JGT  $-12         ***STOP INT?
3510        SETO @SLOW+10     STALL DETECT FLAG.
3520        B    @ABOR
3530 *
3540 SLOW   DATA 0,-1
3550        DATA 0,0          **DOWN/UP.
3560        DATA 0            STALL DELAY-SAMPLE COUNT.
3570        DATA 0            STALL DETECT FLAG.
3580 *
3590 *
3600 VTAB   BYTE 0,5,10,15,20,26
3610        BYTE 31,36,41,46,51,56,61,66
3620        BYTE 71,77,82,87,92,97,102
3630        BYTE 107,112,117,122,128,133
3640        BYTE 138,143,148,153,158,163
3650        BYTE 168,173,179,184,189,194
3660        BYTE 199,204,209,214,219,224
3670        BYTE 230,235,240,245,250,255,0
3680 *
3690 *
3700 * INTERRUPT SERVICE ROUTINES.
3710 *
3720 * SYNC *
3730 LDN1   EQU  R1
3740 SEG1   EQU  R2
3750 *
3760 * SYNC START ROUTINE.
3770 SYST   LI   R14,SYPL     ESCAPE ADDRESS FROM START ROUTINE.
3780        LI   R15,10
3790 **ENSURES THAT IF COMING FROM STOP, DATA OR DISPLAY
3800 **INT'S WHICH HAVE NOT BEEN CLEARED, PROCESSOR
3810 **CAN BE RECALLED BY THEM TO BE CORRECTLY SERVICED.
3820        MOV  R12,R0       MUST SAVE R12 CONTENTS.
3830        LI   R12,>100
3840        LI   SEG1,>706    ENABLE ALL INTERRUPTS.
3850        LDCR SEG1,11
3860        MOV  R0,R12
```

APPENDIX 4A CONTINUED.

```
3870 SYRN   MOV  LDN1,SEG1
3880        CKON              CLEAR SYNC INT.
3890        MOV  *SEG1+,@>C00C NEW SIGNALS TO INVERTER.
3900        CKOF              CLEAR SEG INT.
3910        RTWP
3920 *
3930 DRVF   DATA >001D,>001D,>001E,>001E,>002E,>002E
3940        DATA >002B,>002B,>0033,>0033,>0035,>0035
3950        DATA >801D,>801D,>801E,>801E,>802E,>802E
3960        DATA >802B,>802B,>8033,>8033,>8035,>8035
3970 *
3980 * REVERSE TABLE SHIFTED BY 180 DEG.
3990 *
4000 DRVR   DATA >002E,>002E,>001E,>001E,>001D,>001D
4010        DATA >0035,>0035,>0033,>0033,>002B,>002B
4020        DATA >802E,>802E,>801E,>801E,>801D,>801D
4030        DATA >8035,>8035,>8033,>8033,>802B,>802B
4040 *
4050 * SEG *
4060 SEG2   EQU  R5
4070 SEG1   MOV  *SEG2+,@>C00C NEW SIGNALS TO INVERTER.
4080        CKOF              CLEAR SEG INT.
4090        RTWP
4100 *
4110 * STOP *
4120 * IN SAME WP AS MAIN PROG: IN STOPPED MODE WAITING
4130 * FOR START SIGNAL MAKE SURE VOLTS & LOAD ANGLE ARE
4140 * SET TO VSTART & ZERO RESPECTIVELY.
4150 * WHEN EXECUTING A "STOP" REQUEST ENSURE THAT A START
4160 * IS INHIBITED UNTIL "STOP" COMPLETE.
4170 STOP   TB   10            CHECK IF STOP/START.
4180        JNE  STRT
4190 * SHUT DOWN MOTOR.
4200        MOV  @SLOW+2,@SLOW SET SLOW=-1.
4210        CLR  @VARI+2      RESET INPUT LOAD ANGLE.
4220 ** DISABLE STOP INT. BUT DO NOT CLEAR.
4230        LI   CRUB,>100
4240        SBZ  8 MASK STOP INT.
4250        LI   CRUB,>120
4260 ** CLEAR STOP INT WHEN STOP COMPLETED.
4270 ** ALSO MASK SYNC & SEG AT SAME TIME.
4280 ** REVECTOR SYNC INT FOR START.
4290        RTWP
4300 STRT   LIMI 0
4310        CKON              CLEAR SYNC & SEG: (MAY HAVE
4320        CKOF              BEEN SET DURING BRAK ROUTINE).
4330        LI   CRUB,>100
4340        LI   R8,>702      (9900 INTS 1,8,9,10).
4350        LDCR R8,11
4360        LI   CRUB,>120
4370        LI   R14,AWAY     RETURN ADDRESS.
4380 * ENSURE INT MASK IS SET SO THAT IF DATA OR DISPLAY
4390 * WERE INTERRUPTED BY START, THEY CAN RECALL MICRO.
4400        LI   R15,10
4410        CLR  @>C01A       CLEAR INT.
4420        RTWP              RETURN TO AWAY.
4430 *
4440 * DATA *
4450 DATA   LIMI 0
4460        MOV  @>C002,R0    DATA WORD.
4470        MOV  @>C004,R1    KEY WORD.
4480        LIMI 8            ENABLE INT'S.
4490        ANDI R1,>7        CLEAR SPURIOUS BITS.
4500        DECT R1           CORRECT KEY #.
4510        SLA  R1,1         *2.
4520        MOV  R0,@VARI(R1) PUT DATA IN STORE.
4530        CLR  @>C01C       CLEAR DATA INT.
4540        RTWP
4550 *
4560 *
4570 * DATA INTERRUPT STORAGE AREA.
4580 ** DATA ORDER: VIN; LDIN; SPEED; ETC..
4590 VARI   DATA 0,0,0,0,0,0
4600 *
4610 *
4620 * DISP *
4630 DISP   MOV  @>C004,R1    KEY WORD.
4640 * (TRANSFER DATA IN FIRST PROTECTED INSTRUCTION).
4650        CLR  @>C006
4660        ANDI R1,>F
?
4670        CI   R1,9         RETURN TO VDU?
4680        JNE  $+8          **DISPLAY WANTED.
4690        MOV  R1,@ESC      VDU WANTED FLAG.
4700 * (ESCAPE AFTER NEXT STOP INT).
4710        JMP  $+16         ***QUIT
4720        MOV  R1,@>C012    TRANSFER KEY.
4730        SLA  R1,1         *2
4740        MOV  @M0(R1),R2
4750        MOV  *R2,@>C012
4760 * CLEAR DISP INT.
4770        CLR  @>C01E       ****QUIT
4780        RTWP
4790 ESC    DATA 0            **VDU FLAG.
4800 NINE   DATA 9            **VDU CHECK DATA.
4810 *
4820 *
```

```
4830 RXOP   MOV  R11,R1       READ XOP ROUTINE.
4840        INPU R0           READ CHARACTER.
4850        MOV  *R1+,R2      ****NXCH
4860        MOV  R2,R3
4870        ANDI R2,>FF00
4880        ANDI R3,>FF
4890        C    R0,R2
4900        JEQ  $+14         ****ECHO
4910        CI   R3,>002E     FULL STOP.
4920        JNE  $-20         ****NXCH
4930 * NO VALID CHARACTER: DO NOT ECHO; RING BELL.
4940        WRIT @BELL
4950        JMP  RXOP
4960        DECT R1           ****ECHO
4970        MOV  R1,@RDAT     CHARACTER ADDR. PUT AT RDAT.
4980        BLWP @>E018
4990        RTWP
5000 *
5010 *
5020 IXOP   LI   R12,>80      INPUT CHARACTER XOP ROUTINE.
5030        TB   21
5040        JNE  IXOP
5050        CLR  *R11
5060        STCR *R11,8
5070        SBZ  18
5080        RTWP
5090 *
5100 *
5110 WXOP   MOV  R11,R0       WRITE XOP ROUTINE.
5120        BLWP @>E01C
5130        RTWP
5140 *
5150 *
5160 MXOP   MOV  *R11,R10     MODIFY XOP ROUTINE.
5170 * R10 CONTAINS ADDRESS OF SYMBOL.
5180        MOV  *R10,R1      R1 CONTAINS SYMBOL.
5190        CI   R1,>452E     "E.".
5200        JEQ  EXIT
5210        WRIT @METH
5220 TOP    ANDI R1,>FF00
5230        AI   R1,>20       R1=SYM+SPACE.
5240        MOV  R1,@LINS+2
5250        LI   R3,VARD
5260        S    R3,R10       MOD #.
5270 SHOW   MOV  @VARL(R10),R0
5280        LI   R5,6
5290        LI   R6,>2020
5300        MOV  R6,@LIND(R5) ****LOOP
5310 * (DATA BUFFER + R5)
5320        DECT R5
5330        JLT  $+4          ****FIN
5340        JMP  $-8          ****LOOP
5350        LI   R1,LIND      ****FIN
5360 * (LIND IS BUFFER FOR BLWP @>E00C.)
5370        BLWP @>E00C       CONVERT TO DEC. ASCII.
5380 ERRT   WRIT @LINS
5390        BLWP @>E020       READ IN STRING.
5400 * TERMINATE WITH CR; THIS IS NOT ECHOED.
5410        WRIT @CRLF
5420 * CHECK IF VARD SYMBOL ENTERED.
5430        CLR  R1
5440        MOVB @>0110,R1
5450        CI   R1,>4500     "E".
5460        JEQ  EXIT
5470        LI   R5,8
5480        CB   R1,@VARD(R5) ****COMPARE
5490        JEQ  RSET
5500        DECT R5
5510        JLT  $+4          ****NOMATCH
5520        JMP  $-10         ****COMPARE
5530 * CHECK FOR VALID NUMBER INPUT.
5540        CLR  R1           ****NOMATCH
5550        MOVB @>0110,R1
5560        CI   R1,>2B00     "+".
5570        JEQ  NUM
5580        CI   R1,>2D00     "-".
5590        JEQ  NUM
5600        CI   R1,>3000     ASCII 0.
5610        JLT  ERR
5620        CI   R1,>3900     ASCII 9.
5630        JGT  ERR
5640 * CONVERT DATA AT >0110 TO BINARY.
5650 NUM    LI   R1,>0110
5660        BLWP @>E004
5670 * CHECK IF NUMBER WAS IN CONVERSION RANGE.
5680        LI   R1,NMCK
5690        BLWP @>E00C       CONVERT BACK TO ASCII.
5700        LI   R7,>0110
5710        CLR  R1
5720        MOVB *R7,R1
5730        CI   R1,>2B00     "+".
5740        JNE  $+4          ****NPOS
5750        INC  R7
5760        LI   R8,NMCK      ****NPOS
5770        CB   *R7+,*R8+    ****VALID
5780        JEQ  $-2          ****VALID
```

```
5790        DEC  R7
5800        DEC  R8
5810        CB   *R7,@CRLF
5820        JNE  ERR
5830        CB   *R8,@LIMS+4  (SPACE).
5840        JNE  ERR
5850 * NUMBERS AGREE.
5860        MOV  R0,@VARL(R10)
5870        JMP  SHOW
5880 RSET   LI   R10,VAR0
5890        A    R5,R10
5900        JMP  TOP
5910 ERR    WRIT @ER
5920        JMP  ERRT
5930 EXIT   RTWP
5940 *
5950 *
5960 * HEX XOP ROUTINE.
5970 * ROUTINE ACCEPTS HEX ASCII & KEEPS LAST
5980 * FOUR DIGITS; TERMINATE WITH CR.
5990 HXOP   CLR  R0          HEX BUFFER.
6000        LI   R2,RDAT     RXOP CHARACTER ADDRESS STORE.
6010        LI   R4,>40      HEX A-F SYMBOL CHECK WORD.
6020        WRIT @KNUM
6030 NKY    READ @NKEY       CHECK INPUT AGAINST 0-9 & Q.
6040        MOV  *R2,R9      CHARACTER ADDRESS.
6050        MOV  *R9,R6      CHARACTER.
6060        CI   R6,>512E    "Q."
6070        JEQ  LEAV
6080        MOV  R11,R10     ADDRESS OF MEMORY RECALL DATA.
6090        SLA  R6,4
6100        SRL  R6,11       EXTRACT DATA FROM ASCII & *2.
6110        A    R6,R10
6120        MOV  *R10,R0
6130        LI   R1,LIMD
6140        BLWP @>E010
6150        SLA  R6,7
6160        AI   R6,>3020
6170        MOV  R6,@LIMS+2
6180 FALS   WRIT @LIMS
6190        CLR  R8          FIRST DIGIT INDICATOR.
6200 HXIN   READ @KHEX
6210        MOV  *R2,R9
6220        MOV  *R9,R3
6230        CB   R3,@CRLF
6240        JEQ  CR
6250        CB   R3,@NKEY+20 "Q".
6260        JNE  $+6         ***NOTQ
6270        MOV  *R10,R0     REFILL R0.
6280        JMP  FALS
```

```
6290        SWPB R3          ****NOTQ
6300        COC  R4,R3
6310        JNE  NHEX
6320        AI   R3,9
6330 NHEX   ANDI R3,>F
6340        MOV  R8,R8       IF ZERO, CLEAR R0.
6350        JNE  $+6         ****NOT FIRST DIGIT
6360        CLR  R0
6370        SETO R8
6380        SLA  R0,4        ****NOT FIRST DIGIT
6390        A    R3,R0
6400        JMP  HXIN
6410 CR     LI   R5,6
6420        LI   R6,>2020
6430 CIRC   MOV  R6,@LIMD(R5)
6440        DECT R5
6450        JLT  $+4         ****OUT
6460        JMP  CIRC
6470 * PUT HEX BACK IN STORE.
6480        MOV  R0,*R10     ****OUT
6490        LI   R1,LIMD     BUFFER FOR BLWP @>E010.
6500        BLWP @>E010      CONVERT TO HEX ASCII.
6510        WRIT @LIMS
6520        WRIT @MORE       ANY MORE CHANGES.
6530        JMP  NKY
6540 LEAV   WRIT @CRLF
6550        WRIT @CRLF
6560        RTWP
6570 *
6580 *
6590 G      TEXT 'G.'
6600        DATA >0D0A,>0D0A
6610        TEXT 'TYPE G TO PROCEED...'
6620        DATA 0
6630 BELL   DATA >0700       BELL
6640 QUES   TEXT 'PERIPHERAL STATUS QUESTIONS:'
6650        TEXT ' TYPE Y TO PROCEED. '
6660        DATA 0
6670 CHK1   DATA >0D0A,>0D0A
6680        TEXT 'IS KEYPAD UNIT POWERED UP? '
6690        BYTE 0
6700 CHK2   DATA >0D0A,>0D0A
6710        TEXT 'ARE 60V PSU''S POWERED UP? '
6720        DATA 0
6730 CHK3   DATA >0D0A,>0D0A
6740        TEXT 'IS INVERTER POWERED UP & 100KHZ ON? '
6750        DATA 0
6760 SYNY   TEXT 'Y.'
```

```
6770 VAR    DATA >0D0A,>0D0A
6780        TEXT 'DO YOU WISH TO MODIFY ANY LIMITS?'
6790        TEXT ' (Y/N): '
6800        BYTE 0
6810 CHOS   TEXT 'Y,N.'
6820 VTX1   DATA >0D0A
6830        TEXT 'MAX VOLTS V, START VOLTS B, MAX DELTA D,'
6840        TEXT ' MAX SPEED S, OR RAMP DELAY T?'
6850        DATA 0
6860 NOTE   DATA >0D0A,>0D0A
6870        TEXT '(MAX NUMERICAL INPUT IS /32767/).'
6880        BYTE 0
6890        DATA 0
6900 VTX2   DATA >0D0A,>0D0A
6910        TEXT 'TYPE SYMBOL OR E TO EXIT: '
6920        DATA 0
6930 METH   DATA >0D0A
6940        TEXT '** TYPE "CR" TO ENTER SYMBOL OR DATA **'
6950        BYTE 0
6960 RDAT   DATA 0
6970 LINS   DATA >0D0A,>0
6980        TEXT ' = '
6990 LIND   DATA 0,0,0,0,0
7000 VARD   TEXT 'V,B,D,S,T,E.'
7010        DATA 0
7020 VARL   DATA 30,7,0,30000,200 LIMIT VALUES.
7030 CRLF   DATA >0D0A,0
7040 ER     TEXT 'WHAT?'
7050        BYTE 0
7060 NMCK   DATA 0,0,0,0,0
7070 HQES   DATA >0D0A,>0D0A
7080        TEXT 'DO YOU WISH TO CHANGE ANY RECALL'
7090        TEXT ' MEMORY LOCATIONS? (Y/N): '
7100        DATA 0
7110 HXCR   DATA >0D0A
7120        TEXT '***IF INSPECTED DATA IS CORRECT, TYPE '
7130        TEXT '"CR"***'
7140        BYTE 0
7150 KNUM   DATA >0D0A,>0D0A
7160        TEXT 'TYPE KEY # 0-9 OR Q TO QUIT: '
?
7170        BYTE 0
7180 KHEX   DATA >0D2C        "CR,".
7190        TEXT 'A,B,C,D,E,F,'
7200 HKEY   TEXT '0,1,2,3,4,5,6,7,8,9,Q.'
7210 * RECALL ADDRESSES M0-M9.
7220 **WORKING VOLTS: WORKING LOADANG: SPEED: ETC..
7230 M0     DATA >1004,>100C,>C000,0,0,0,0,0,0,0
```

```
7240 * DATA INTERRUPT STORAGE AREA ADDRESSES.
7250 ** ADDRESS ORDER: VIN; LDIN; SPEED; ETC..
7260        DATA 0,0,0,0,0,0  SET BY INITIALIZATION ROUTINE.
7270 MORE   DATA >0D0A
7280        TEXT 'ANY MORE? TYPE # OR Q: '
7290        BYTE 0
7300 KEYP   DATA >0D0A,>0D0A
7310        TEXT 'READY FOR KEYPAD? (Y/N): '
7320        BYTE 0
7330 FAIL   DATA >0707,>0707
7340        TEXT 'START ABORTED....TRY INCREMENTING '
7350        TEXT 'START VOLTS?'
7360        DATA 0
7370 STAL   DATA >0707,>0707
7380        TEXT 'MOTOR STALLED....INVESTIGATE REASON!'
7390        DATA 0
7400 *
7410 *
7420 *
7430 ** SUBROUTINE TO CLEAR VDU SCREEN.
7440 CLER   DATA >25C0        SUBROUTINE WP.
7450        DATA $+2          PC.
7460        LI   R0,Y1
7470        BLWP @>E01C
7480        LI   R1,20        20MS REFRESH TIME.
7490        CLR  R0
7500 NULL   BLWP @>E018
7510        DEC  R1
7520        JGT  NULL
7530        RTWP
7540 *
7550 *
7560 * SUBPROGRAM TO DRAW LARGE VDU TITLES.
7570 BANN   DATA >2500        SUBROUTINE WP.
7580        DATA $+4
7590        DATA 0            MESSAGE IDENTIFIER STORE.
7600        MOV  @BANN+4,R9
7610        BLWP @CLER        CLEAR SCREEN.
7620        MOV  R9,R0
7630        AI   R0,4
7640        MOV  R0,@CHEK
7650        MOV  @DATK(R9),R8
7660 NXT    MOV  @DATB(R9),R3
7670        INCT R9
7680        MOV  @DATK(R9),R10
7690        DECT R9
7700        MOV  @DATJ(R9),R7
7710 NXT2   A    R7,R3
7720        MOV  @DATC(R8),R2
```

```
7730        MOV  @DATN(R8),R4
7740        MOV  @DATL(R8),R5
7750        BL   @CHS
7760        INCT R8
7770        C    R8,R10
7780        JLT  NXT2
7790        INCT R9
7800        C    R9,@CHEK
7810        JLT  NXT
7820        MOV  @DATB+8,R1
7830        LI   R2,>2000
7840        BL   @DRAW
7850        RTWP                RETURN TO MAIN PROG.
7860 CHS    CLR  R6
7870        MOV  R11,R12
7880 NXT1   MOV  R3,R1
7890        A    *R5+,R1
7900        BL   @DRAW
7910        INC  R6
7920        C    R6,R4
7930        JLT  NXT1
7940        B    *R12
7950 *
7960 * SUBROUTINE.
7970 DRAW   MOVB R1,@X
7980        MOV  R1,@Y
7990        MOVB @Z,@Y
8000        MOVB R2,@Y+2
8010        MOVB @Z+1,@Y+3
8020        LI   R0,DAT1
8030        BLWP @>E01C
8040        RT
8050 *
8060 *
```

```
8070 CHEK   DATA 0
8080 DAT1   DATA >1B58
8090 X      DATA >1B
8100 Y      BYTE 0,0,0,0,0
8110 Y1     BYTE >1B,>4B,0
8120 Z      BYTE >59,>08
8130 *
8140 **DAT2
8150        DATA >0705,>0604,>0503,>0402,>0403,>0404,>0305
8160        DATA >0205,>0104,>0103,>0102
8170 *
8180 **DAT3
8190        DATA >0701,>0601,>0501,>0401,>0301,>0201,>0101
8200        DATA >0202,>0303,>0403,>0204,>0105,>0205,>0305
8210        DATA >0405,>0505,>0605,>0705
8220 *
8230 **DAT4
8240        DATA >0605,>0704,>0703,>0702,>0601,>0501,>0401
8250        DATA >0301,>0201,>0102,>0103,>0104,>0205,>0305
8260        DATA >0405,>0505
8270 *
8280 **DAT5
8290        DATA >0101,>0105
8300 *
8310 **DAT6
8320        DATA >0104,>0103,>0102,>0203,>0303,>0403,>0503
8330        DATA >0603,>0703,>0704,>0702
8340 *
8350 **DAT7
8360        DATA >0705,>0604,>0503,>0402,>0303,>0204,>0105
8370 *
8380 **DAT8
8390        DATA >0701,>0601,>0501,>0401,>0301,>0201,>0101
8400        DATA >0102,>0103,>0104,>0105,>0402,>0403,>0702
8410        DATA >0703,>0704,>0705
```

```
8420 *
8430 **DAT9
8440         DATA >0703,>0603,>0503,>0403,>0304,>0204,>0105
8450         DATA >0302,>0202,>0101
8460 *
8470 **DAT0
8480         DATA >0402,>0403,>0404
8490 *
8500 **DATD
8510         DATA >0701,>0601,>0501,>0401,>0301,>0201,>0101
8520         DATA >0102,>0103,>0104,>0205,>0305,>0405,>0505
8530         DATA >0605,>0704,>0703,>0702
8540 *
8550 **DAT
8560         DATA >0701,>0601,>0501,>0401,>0301,>0201,>0102
8570         DATA >0103,>0104,>0205,>0305,>0405,>0505,>0605
8580         DATA >0705,>0402,>0403,>0404
8590 *
8600 **DATG
8610         DATA >0205,>0104,>0103,>0102,>0201,>0301,>0401
8620         DATA >0501,>0601,>0702,>0703,>0704,>0605,>0505
8630         DATA >0504
8640 *
8650 **DATS
8660         DATA >0101
8670 *
8680 DATK    DATA 0,10,20,30,42
8690 DATJ    DATA 14,14,14,12
8700 DATN    DATA 18,11,13,18,16
8710         DATA 18,16,11,16,18
8720         DATA 15,16,1,11,16
8730         DATA 14,17,10,15,18,18
8740 DATC    DATA >4D00,>4900,>4300,>5200,>4F00
8750         DATA >4D00,>4F00,>5400,>4F00,>5200
8760         DATA >4700,>4F00,>2000,>5400,>4F00
8770         DATA >4B00,>4500,>5900,>5000,>4100,>4400
8780 *
8790 **(DAT3,DAT6,DAT4,DAT2,DAT4)
8800 DATL    DATA $-374,$-302,$-338,$-396,$-338
8810 *
8820 **(DAT3,DAT4,DAT5,DAT4,DAT2)
8830         DATA $-384,$-348,$-316,$-348,$-406
8840 *
8850 **(DATG,DAT4,DATS,DAT5,DAT4)
8860         DATA $-154,$-358,$-124,$-326,$-358
8870 *
8880 **(DAT7,DAT8,DAT9,DAT0,DAT,DATD)
8890         DATA $-310,$-296,$-262,$-242,$-200,$-236
8900 *
8910 DATB    DATA >221A,>2C1A
8920         DATA >221A,>2C1A,>3621
8930         END  MAG
```

```
0010 * PROGRAM TO RUN THE 7 PHASE MOTOR USING
0020 * INTERRUPTS TO MAINTAIN SYNCHRONISM.
0030 * THE MOTOR IS DRIVEN BY THE MOSFET INVERTER
0040 * POWERED BY THE 60V PSU'S.
0050 * DECEMBER 3, 1981. N.M. MALLINSON.
0060 *
0070 *
0080 *** THIS PROGRAM SHOULD BE RUN ON A PERKIN ELMER VDU
0090 *** (PREFERABLY WITH 9600 BAUD RATE).
0100 *
0110 ** DATA & COMMANDS ARE ENTERED VIA A KEYPAD
0120 ** DURING MOTOR OPERATION.
0130 *
0140        IDT  'SQ7PH'
0150        AORG >1040
0160 * DEFINE REGISTER LABELS.
0170 R0     EQU  0
0180 R1     EQU  1
0190 R2     EQU  2
0200 R3     EQU  3
0210 R4     EQU  4
0220 R5     EQU  5
0230 R6     EQU  6
0240 R7     EQU  7
0250 R8     EQU  8
0260 R9     EQU  9
0270 R10    EQU  10
0280 R11    EQU  11
0290 R12    EQU  12
0300 R13    EQU  13
0310 R14    EQU  14
0320 R15    EQU  15
0330 SQ7P   LWPI >1000
0340 *
0350        CLR  @ESC          **CLEAR VDU FLAG.
0360 *
0370 * SET ALL IMPORTANT OUTPUT PORTS SAFE.
0380        CLR  @>C00E        VOLTS.
0390        SETO @>C00C        INVERTER CONTROL.
0400        CLR  @>C010        INVERTER SELECT.
0410        CLR  @>C006        DISPLAY PORT.
0420        CLR  @>C012           "      "
0430        CLR  @>C012           "      "
0440 *
0450 *
0460 **VECTOR VARI DATA ADDRESSES FOR DISPLAY ROUTINE.
0470        LI   R1,M0
0480        AI   R1,20
0490        LI   R0,VARI
0500        MOV  R0,*R1+       ****DATA VECTOR
0510        INCT R0
0520        CI   R1,MORE       END OF ADDRESS BLOCK.
0530        JLT  $-8           ****DATA VECTOR
0540 *
0550 *
0560 * VECTOR INPUT, READ,WRITE,
0570 * MODIFY & HEX MOD XOP'S.
0580        DXOP INPU,4
0590        LI   R0,>25A0      XOP4 WP.
0600        MOV  R0,@>0050
0610        LI   R0,IXOP       XOP4 PC.
0620        MOV  R0,@>0052
0630        DXOP READ,5
0640        LI   R0,>2520      XOP5 WP.
0650        MOV  R0,@>0054
0660        LI   R0,RXOP       XOP5 PC.
0670        MOV  R0,@>0056
0680        DXOP WRIT,6
0690        LI   R0,>2540      XOP6 WP.
0700        MOV  R0,@>0058
0710        LI   R0,WXOP       XOP6 PC.
0720        MOV  R0,@>005A
0730        DXOP MOD,0
0740        LI   R0,>2560      XOP0 WP.
0750        MOV  R0,@>0040
0760        LI   R0,MXOP       XOP0 PC.
0770        MOV  R0,@>0042
0780        DXOP HEX,1
0790        LI   R0,>2580      XOP1 WP.
0800        MOV  R0,@>0044
0810        LI   R0,HXOP       XOP1 PC.
0820        MOV  R0,@>0046
0830 *
0840        CLR  @BANN+4       **MESSAGE IDENTIFIER.
0850        BLWP @BANN         WRITE VDU TITLE.
0860        WRIT @G+2          **GO? MESSAGE.
0870        READ @G
0880        BLWP @CLER         CLEAR SCREEN.
0890        WRIT @QUES         PERIPHERAL STATUS QUESTIONS.
0900        WRIT @CHK1
0910        READ @SYNY
0920        WRIT @CHK2
0930        READ @SYNY
0940        WRIT @CHK3
0950        READ @SYNY
0960 OPT    BLWP @CLER         CLEAR SCREEN.
0970        WRIT @VAR          IS CHANGE IN LIMIT VALUES WANTED?
0980        READ @CHOS         YES/NO CHOICE.
```

```
0990        LI   R0,CHOS
1000        C    @RDAT,R0
1010        JNE  $+22          ****CONTINUE
1020        WRIT @VTX1
1030        WRIT @NOTE
1040        WRIT @VTX2
1050        READ @VARD
1060        MOD  @RDAT
1070 * CLEAR SCREEN.
1080        BLWP @CLER         ****CONTINUE
1090        WRIT @HQES         CHANGE RECALL MEMORY LOCATIONS?
1100        READ @CHOS
1110        LI   R0,CHOS
1120        C    @RDAT,R0
1130        JNE  $+10          ****KEYPAD
1140        WRIT @HXCR
1150        HEX  @M0           MODIFY DATA AT LOCATION M0-M9.
1160        WRIT @KEYP         ****KEYPAD
1170        READ @CHOS
1180        C    @RDAT,R0
1190        JNE  OPT
1200        LI   R0,4
1210        MOV  R0,@BANN+4    **MESSAGE IDENTIFIER.
1220        BLWP @BANN
1230 *
1240 *
1250 ********************
1260 * MAIN MOTOR PROGRAM.
1270 * COMPLETE INITIALISATION OF SYSTEM.
1280 *
1290 * VECTOR 9900 INTERRUPTS 1,2,8,9,10.
1300 * INTERRUPTS ARE (9900 # IN BRACKETS):
1310 * INT1=SYNC (1).
1320 * INT2=SEGI (2).
1330 * INT6=STOP (8).
1340 * INT7=DATA (9).
1350 * INT8=DISP (10).
1360 *
1370 * THE WORKSPACES FOR THE MAIN PROGRAM, SYNC, SEG & STOP
1380 * ARE OVERLAPPED TO EASE DATA TRANSFER.
1390        LI   R1,INFO
1400        LI   R2,ADDR
1410        MOV  *R2+,R3       ****VECTOR
1420        MOV  *R1+,*R3
1430        MOV  *R1,*R1       COMPARE TO ZERO.
1440        JNE  $-6           ****VECTOR
1450        JMP  CLRI
1460 * INFORMATION ARRANGED IN FORMAT WP1;PC1;WP2;PC2; ETC..
1470 *
1480 INFO   DATA >1012,SYST,>100C,SEGI
1490        DATA >1000,STOP,>25E0,DATA,>2600,DISP
1500        DATA 0             ***END OF VECTORING.
1510 *
1520 * ADDRESSES IN FORMAT INT1 WP STORE,
1530 * INT1 PC STORE, ETC..
1540 ADDR   DATA >0004,>0006,>0008,>000A
1550        DATA >0020,>0022,>0024,>0026,>0028,>002A
1560 *
1570 CLRI   CKON               CLEAR ALL INTERRUPT LATCHES.
1580        CKOF
1590        LI   R1,>C014
1600        CLR  *R1+          ****WIPE
1610        CI   R1,>C020      STOP VALUE.
1620        JLT  $-6           ****WIPE
1630 *
1640 * SET UP INTERRUPTS.
1650 * (DO NOT ENABLE SYNC UNTIL START REQUESTED).
1660 * (DO NOT ENABLE SEG UNTIL SYNC ACHIEVED).
1670        LI   R12,>100      CRU BASE.
1680        LI   R1,>700       ENABLE 9900 INT'S 8,9,10.
1690        LDCR R1,11         SET INT MODE.
1700 *
1710 * SET DEMANDED VOLTS TO START VALUE & OTHER
1720 * VARIABLE INPUTS TO ZERO FOR SAFETY.
1730        MOV  @VARL+2,@VARI
1740        LI   R1,8
1750        CLR  @VARI+2(R1)   ****CLEAR
1760        DECT R1
1770        JLT  $+4           ****CLRFIN
1780        JMP  $-8           ****CLEAR
1790 *
1800 *
1810 * MAIN PROGRAM.
1820 VRQN   EQU  R1
1830 VRQD   EQU  R2
1840 DANW   EQU  R3
1850 DAQL   EQU  R4
1860 LRQN   EQU  R5
1870 LRQD   EQU  R6
1880 CRUB   EQU  R12
1890 LDAN   EQU  R10
1900 SEG    EQU  R11
1910 * INITIALISE KEYPAD, INVERTER & PSU CRUBIT.
1920        LI   CRUB,>120     ****CLRFIN
1930        SBZ  8
```

```
1940 *
1950        CLR  VRQN
1960        CLR  VRQO
1970        CLR  DANW
1980        CLR  DAOL
1990        CLR  LRQN
2000        CLR  @SLOW
2010 * ENSURE OUTPUT VOLTS ARE STILL ZERO.
2020        CLR  @>C00E
2030 *
2040 * NOW WAIT FOR STOP, DATA OR DISP INT.
2050 REST   C    @NINE,@ESC
2060        JNE  $+20          ****NOT VDU
2070        LIMI 0
2080 * DISABLE INVERTER.
2090        LI   CRUB,>120
2100        SBO  8
2110        WRIT @BELL         LOOK AT VDU!!!
2120        B    @SQ7P
2130        LIMI 10            ****NOT VDU
2140 *
2150        JMP  REST          WAIT FOR START INT.
2160 * CHECK REQUIRED DIRECTION OF ROTATION.
2170 AWAY   TB   9             CRUBIT >129.
2180        JNE  $+8           ***FWD
2190 * REVERSE SELECTED.
2200        LI   LDAN,DRVR
2210        JMP  $+6           ****HOP
2220        LI   LDAN,DRVF     ****FWD
2230        MOV  LDAN,SEG      ****HOP
2240 * ENABLE INVERTER SIGNALS.
2250        LI   R0,>1234
2260        MOV  R0,@>C010
2270 *
2280        MOV  @VARL+2,VRQO
2290        MOVB @VTAB(VRQO),DAOL
2300        SRL  DAOL,8
2310        AI   DAOL,>6000
2320        MOV  DAOL,@>C00E   SET START VOLTS.
2330        CLR  LRQO
2340 * CHECK MOTOR IS STATIONARY PRIOR TO START.
2350        MOV  @>C000,R8     ****SPEED
2360        CI   R8,16
2370        JNE  $-8           ****SPEED
2380 *
2390        LI   R0,150        MAX # OF STARTING STEPS.
2400        DEC  R0            ****NX
2410        JEQ  ABOR          ABORT START.
2420        LI   R8,>FFF
2430        DEC  R8            ****DELAY
2440        JGT  $-2           ****DELAY
2450 * CHECK IF STOP REQUIRED.
2460        MOV  @SLOW,@SLOW
2470        JNE  BRAK
2480        MOV  *SEG+,@>C00C
2490 * TEST FOR NEGATIVE NUMBER: IF SO MSB IS SET
2500 * AND TABLE POINTER REQUIRES RESETTING.
2510        JGT  $-24          ****NX
2520        AI   SEG,-112
2530        JMP  $-30          ****NX
2540 *
2550 ** ABORT START: DISABLE INVERTER.
2560 ABOR   LI   CRUB,>120
2570        SBO  8
2580        SETO @>C00C
2590        CLR  @>C010        DISABLE INVERTER.
2600        CLR  @>C00E
2610        BLWP @CLER
2620        MOV  @SLOW+10,@SLOW+10 STALL DETECT FLAG.
2630        JEQ  $+12          ****START FAILURE
2640        WRIT @STAL
2650        CLR  @SLOW+10      STALL DETECT FLAG.
2660        JMP  $+6           ****GO? MESSAGE
2670        WRIT @FAIL         ****START FAILURE
2680        WRIT @G+2          ****GO? MESSAGE
2690        READ @G
2700        B    @OPT
2710 *
2720 SYPL   LI   R8,SYRN       SYNC PULSE RECEIVED.
2730        MOV  R8,@>0006     VECTOR SYNC FOR NORMAL RUN.
2740 *
2750 MAIN   LI   R0,>4FFF      STALL DETECT DELAY.
2760        MOV  R0,@SLOW+8    STALL DETECT DELAY STORE.
2770        MOV  @SLOW,@SLOW   ****STOP INT?
2780        JEQ  INPT
2790 * CLEAR VOLTS AND WAIT FOR SYSTEM TO RAMP DOWN.
2800        CLR  VRQN
2810        CI   DAOL,>6000    ZERO VOLTS.
2820        JGT  DIFF
2830 BRAK   LIMI 0
2840        CLR  @SLOW
2850        CLR  @>C00E
2860        SETO @>C00C
2870        CLR  @>C010        DISABLE INVERTER.
2880        LI   SEG,SYST
2890        MOV  SEG,@>0006    VECTOR SYST INT.
```

```
2900 *
2910 **NOTE STOP/START FLIPFLOP IN KEYPAD MAY BE
2920 **SET TO START WHILST MOTOR IS STOPPING: TO AVOID
2930 **TROUBLE WHEN KEY NEXT PRESSED (GENERATING A
2940 **STOP) CLEAR FLIPFLOP AT END OF STOP.
2950        LI   CRUB,>120
2960        SBO  8
2970        SBZ  8
2980        CLR  @>C01A        CLEAR STOP INT.
2990        LI   CRUB,>100
3000        LI   SEG,>700
3010        LDCR SEG,11        MASK SYNC & SEG INTS.
3020        LI   CRUB,>120
3030 * SET INITIAL RUNNING VOLTS TO START VALUE.
3040 * (CAN BE CHANGED THOUGH DURING STOPPED TIME).
3050        MOV  @VARL+2,@VARI
3060        B    @REST
3070 *
3080 INPT   MOV  @VARI,VRQN    INPUT VOLTS.
3090 DIFF   C    VRQN,VRQO
3100        JEQ  ANG
3110 * CHECK IF VRQN IS IN LIMITS.
3120        C    VRQN,@VARL
3130        JLT  $+8           ****OKAY
3140        JEQ  $+6           ***OKAY
3150        MOV  @VARL,VRQN
3160        MOVB @VTAB(VRQN),DANW ****OKAY
3170        SRL  DANW,8
3180        AI   DANW,>6000
3190        C    DANW,DAOL
3200        JEQ  UPDA          UPDATE REGISTERS.
3210        JGT  INCR
3220 *DECREMENT VOLTS.
3230        DEC  @SLOW+4       ****DOWN
3240        JGT  ANG
3250        DEC  DAOL
3260        MOV  @VARL+8,@SLOW+4 ****DOWN
3270        MOV  DAOL,@>C00E   OUTPUT NEW VOLTS.
3280        JMP  ANG
3290 INCR   DEC  @SLOW+6       ****UP
3300        JGT  ANG
3310        INC  DAOL
3320        MOV  @VARL+8,@SLOW+6 ****UP
3330        MOV  DAOL,@>C00E   OUTPUT NEW VOLTS.
3340        JMP  ANG
3350 UPDA   MOV  DANW,DAOL
3360        MOV  VRQN,VRQO
3370 * LOAD ANGLE CHANGE SECTION.
3380 ANG    MOV  @VARI+2,LRQN INPUT LOAD ANGLE.
3390        C    LRQN,@VARL+4 MAX LOAD ANGLE.
3400        JLT  $+8           ****FINE
3410        JEQ  $+6           ***FINE
3420        MOV  @VARL+4,LRQN
3430        CI   LRQN,0        ***FINE
3440        JGT  $+8           ****91
3450        JEQ  $+6           ***91
3460        CLR  LRQN
3470        JMP  $+12          ****CLR R8
3480        CI   LRQN,91       ****91
3490        JLT  $+6           ***CLR R8
3500        LI   LRQN,90
3510        CLR  R8            ****CLR R8
3520        C    LRQN,@LTAB(R8) ****DELTA
3530        JLT  $+6           ****DELTA OLD
3540        INCT R8
3550        JMP  $-8           ****DELTA
3560        CLR  R7            ****DELTA OLD
3570        C    LRQO,@LTAB(R7) ****DELCOMP
3580        JLT  $+6           ****CHOICE
3590        INCT R7
3600        JMP  $-8           ****DELCOMP
3610        C    R7,R8         ****CHOICE
3620        JNE  ACT           ACTION REQUIRED.
3630        MOV  @LVAL(R7),LRQO
3640        JMP  BOT
3650 ACT    JGT  DELD          DECREASE DELTA.
3660        INCT LDAN          INCREASE DELTA.
3670        INCT R7
3680        MOV  @LVAL(R7),LRQO
3690        JMP  BOT
3700 DELD   DECT LDAN
3710        DECT R7
3720        MOV  @LVAL(R7),LRQO
3730 BOT    MOV  @SLOW+8,@SLOW+8 STALL DELAY COUNT.
3740        JEQ  $+10          ****CHECK STALL
3750        DEC  @SLOW+8
3760        B    @MAIN+8       ****STOP INT?
3770        MOV  @>C000,R0     ****CHECK STALL
3780        CI   R0,16
3790        JNE  $-12          ****STOP INT?
3800        SETO @SLOW+10      STALL DETECT FLAG.
3810        B    @ABOR
3820 *
3830 SLOW   DATA 0,-1
3840        DATA 0,0           **DOWN/UP.
3850        DATA 0             STALL DELAY-SAMPLE COUNT.
3860        DATA 0             STALL DETECT FLAG.
```

```
3870 *
3880 *
3890 VTAB   BYTE 0,4,9,13,17,21,26,30,34
3900        BYTE 38,43,48,51,55,60,64,68
3910        BYTE 72,77,81,85,89,94,98,102
3920        BYTE 106,111,115,119,123,128
3930        BYTE 132,136,140,145,149,153
3940        BYTE 157,161,166,170,174,179
3950        BYTE 183,187,191,196,200,204
3960        BYTE 208,213,217,221,225,230
3970        BYTE 234,238,242,247,251,255
3980        BYTE 0
3990 *
4000 * TABLE OF LOAD ANGLE COMPARISON POINTS.
4010 LTAB   DATA 4,10,16,23,29,36,42
4020        DATA 48,55,61,68,74,81,87,91
4030 LVAL   DATA 0,6,13,19,26,32,39,45
4040        DATA 51,58,64,71,77,84,90
4050 *
4060 *
4070 * INTERRUPT SERVICE ROUTINES.
4080 *
4090 * SYNC *
4100 LDN1   EQU  R1
4110 SEG1   EQU  R2
4120 *
4130 * SYNC START ROUTINE.
4140 SYST   LI   R14,SYPL     ESCAPE ADDRESS FROM START ROUTINE.
4150        LI   R15,10
4160 **ENSURES THAT IF COMING FROM STOP, DATA OR DISPLAY
4170 **INT'S WHICH HAVE NOT BEEN CLEARED, PROCESSOR
4180 **CAN BE RECALLED BY THEM TO BE CORRECTLY SERVICED.
4190        MOV  R12,R0       MUST SAVE R12 CONTENTS.
4200        LI   R12,>100
4210        LI   SEG1,>706    ENABLE ALL INTERRUPTS.
4220        LDCR SEG1,11
4230        MOV  R0,R12
4240 SYRN   MOV  LDN1,SEG1
4250        CKON              CLEAR SYNC INT.
4260        MOV  *SEG1+,@>C00C NEW SIGNALS TO INVERTER.
4270        CKOF              CLEAR SEG INT.
4280        RTWP
4290 *
4300 DRVF   DATA >3666,>3666,>3666,>3666,>1E66,>1E66
4310        DATA >1E66,>1E66,>1B66,>1B66,>1B66,>1B66
4320        DATA >19E6,>19E6,>19E6,>19E6,>19B6,>19B6
4330        DATA >19B6,>19B6,>199E,>199E,>199E,>199E
4340        DATA >199B,>199B,>199B,>199B,>3999,>3999
4350        DATA >3999,>3999,>2D99,>2D99,>2D99,>2D99
4360        DATA >2799,>2799,>2799,>2799,>26D9,>26D9
4370        DATA >26D9,>26D9,>2679,>2679,>2679,>2679
4380        DATA >266D,>266D,>266D,>266D,>2667,>2667
4390        DATA >2667,>2667,>B666,>B666,>B666,>B666
4400        DATA >9E66,>9E66,>9E66,>9E66,>9B66,>9B66
4410        DATA >9B66,>9B66,>99E6,>99E6,>99E6,>99E6
4420 *
4430 * REVERSE TABLE SHIFTED BY 180 DEG.
4440 *
4450 DRVR   DATA >199B,>199B,>199B,>199B,>199E,>199E
4460        DATA >199E,>199E,>19B6,>19B6,>19B6,>19B6
4470        DATA >19E6,>19E6,>19E6,>19E6,>1B66,>1B66
4480        DATA >1B66,>1B66,>1E66,>1E66,>1E66,>1E66
4490        DATA >3666,>3666,>3666,>3666,>2667,>2667
4500        DATA >2667,>2667,>266D,>266D,>266D,>266D
4510        DATA >2679,>2679,>2679,>2679,>26D9,>26D9
4520        DATA >26D9,>26D9,>2799,>2799,>2799,>2799
4530        DATA >2D99,>2D99,>2D99,>2D99,>3999,>3999
4540        DATA >3999,>3999,>999B,>999B,>999B,>999B
4550        DATA >999E,>999E,>999E,>999E,>99B6,>99B6
4560        DATA >99B6,>99B6,>9B66,>9B66,>9B66,>9B66
4570 *
4580 * SEG *
?
4590 SEG2   EQU  R5
4600 SEGI   MOV  *SEG2+,@>C00C NEW SIGNALS TO INVERTER.
4610        CKOF              CLEAR SEG INT.
4620        RTWP
4630 *
4640 * STOP *
4650 * IN SAME WP AS MAIN PROG: IN STOPPED MODE WAITING
4660 * FOR START SIGNAL MAKE SURE VOLTS & LOAD ANGLE ARE
4670 * SET TO VSTART & ZERO RESPECTIVELY.
4680 * WHEN EXECUTING A "STOP" REQUEST ENSURE THAT A START
4690 * IS INHIBITED UNTIL "STOP" COMPLETE.
4700 STOP   TB   10            CHECK IF STOP/START.
4710        JNE  STRT
4720 * SHUT DOWN MOTOR.
4730        MOV  @SLOW+2,@SLOW SET SLOW=-1.
4740        CLR  @VAR1+2      RESET INPUT LOAD ANGLE.
4750 ** DISABLE STOP INT. BUT DO NOT CLEAR.
4760        LI   CRUB,>100
4770        SBZ  8 MASK STOP INT.
4780        LI   CRUB,>120
4790 ** CLEAR STOP INT WHEN STOP COMPLETED.
4800 ** ALSO MASK SYNC & SEG AT SAME TIME.
4810 ** REVECTOR SYNC INT FOR START.
4820        RTWP
```

```
4830 STRT   LIMI 0
4840        CKON                CLEAR SYNC & SEG: (MAY HAVE
4850        CKOF                BEEN SET DURING BRAK ROUTINE).
4860        LI   CRUB,>100
4870        LI   R8,>702        (9900 INTS 1,8,9,10).
4880        LDCR R8,11
4890        LI   CRUB,>120
4900        LI   R14,AWAY       RETURN ADDRESS.
4910 * ENSURE INT MASK IS SET SO THAT IF DATA OR DISPLAY
4920 * WERE INTERRUPTED BY START, THEY CAN RECALL MICRO.
4930        LI   R15,10
4940        CLR  @>C01A         CLEAR INT.
4950        RTWP                RETURN TO AWAY.
4960 *
4970 * DATA *
4980 DATA   LIMI 0
4990        MOV  @>C002,R0      DATA WORD.
5000        MOV  @>C004,R1      KEY WORD.
5010        LIMI 8              ENABLE INT'S.
5020        ANDI R1,>7          CLEAR SPURIOUS BITS.
5030        DECT R1             CORRECT KEY #.
5040        SLA  R1,1           *2.
5050        MOV  R0,@VARI(R1)   PUT DATA IN STORE.
5060        CLR  @>C01C         CLEAR DATA INT.
5070        RTWP
5080 *
5090 *
5100 * DATA INTERRUPT STORAGE AREA.
5110 ** DATA ORDER: VIN; LDIN; SPEED; ETC..
5120 VARI   DATA 0,0,0,0,0,0
5130 *
5140 *
5150 * DISP *
5160 DISP   MOV  @>C004,R1      KEY WORD.
5170 * (TRANSFER DATA IN FIRST PROTECTED INSTRUCTION).
5180        CLR  @>C006
5190        ANDI R1,>F
5200        CI   R1,9           RETURN TO VDU?
5210        JNE  $+8            **DISPLAY WANTED.
5220        MOV  R1,@ESC        VDU WANTED FLAG.
5230 * (ESCAPE AFTER NEXT STOP INT).
5240        JMP  $+16           ***QUIT
5250        MOV  R1,@>C012      TRANSFER KEY.
5260        SLA  R1,1           *2
5270        MOV  @M0(R1),R2
5280        MOV  *R2,@>C012
5290 * CLEAR DISP INT.
5300        CLR  @>C01E         ****QUIT
5310        RTWP
```

```
5320 ESC    DATA 0              **VDU FLAG.
5330 NINE   DATA 9              **VDU CHECK DATA.
5340 *
5350 *
5360 RXOP   MOV  R11,R1         READ XOP ROUTINE.
5370        INPU R0             READ CHARACTER.
5380        MOV  *R1+,R2        ****NXCH
5390        MOV  R2,R3
5400        ANDI R2,>FF00
5410        ANDI R3,>FF
5420        C    R0,R2
5430        JEQ  $+14           ****ECHO
5440        CI   R3,>002E       FULL STOP.
5450        JNE  $-20           ****NXCH
5460 * NO VALID CHARACTER: DO NOT ECHO; RING BELL.
5470        WRIT @BELL
5480        JMP  RXOP
5490        DECT R1             ****ECHO
5500        MOV  R1,@RDAT       CHARACTER ADDR. PUT AT RDAT.
5510        BLWP @>E018
5520        RTWP
5530 *
5540 *
5550 IXOP   LI   R12,>80        INPUT CHARACTER XOP ROUTINE.
5560        TB   21
5570        JNE  IXOP
5580        CLR  *R11
5590        STCR *R11,8
5600        SBZ  18
5610        RTWP
5620 *
5630 *
5640 WXOP   MOV  R11,R0         WRITE XOP ROUTINE.
5650        BLWP @>E01C
5660        RTWP
5670 *
5680 *
5690 MXOP   MOV  *R11,R10       MODIFY XOP ROUTINE.
5700 * R10 CONTAINS ADDRESS OF SYMBOL.
5710        MOV  *R10,R1        R1 CONTAINS SYMBOL.
5720        CI   R1,>452E       "E.".
5730        JEQ  EXIT
5740        WRIT @METH
5750 TOP    ANDI R1,>FF00
5760        AI   R1,>20         R1=SYM+SPACE.
5770        MOV  R1,@LINS+2
5780        LI   R3,VARD
5790        S    R3,R10         MOD #.
5800 SHOW   MOV  @VARL(R10),R0
```

```
5810        LI   R5,6
5820        LI   R6,>2020
5830        MOV  R6,@LIND(R5) ****LOOP
5840 * (DATA BUFFER + R5)
5850        DECT R5
5860        JLT  $+4          ****FIN
5870        JMP  $-8          ****LOOP
5880        LI   R1,LIND      ****FIN
5890 * (LIND IS BUFFER FOR BLWP @>E00C.)
5900        BLWP @>E00C       CONVERT TO DEC. ASCII.
5910 ERRT   WRIT @LINS
5920        BLWP @>E020       READ IN STRING.
5930 * TERMINATE WITH CR; THIS IS NOT ECHOED.
5940        WRIT @CRLF
5950 * CHECK IF VARD SYMBOL ENTERED.
5960        CLR  R1
5970        MOVB @>0110,R1
5980        CI   R1,>4500     "E".
5990        JEQ  EXIT
6000        LI   R5,8
6010        CB   R1,@VARD(R5) ****COMPARE
6020        JEQ  RSET
6030        DECT R5
6040        JLT  $+4          ****NOMATCH
6050        JMP  $-10         ****COMPARE
6060 * CHECK FOR VALID NUMBER INPUT.
6070        CLR  R1           ****NOMATCH
6080        MOVB @>0110,R1
6090        CI   R1,>2B00     "+".
6100        JEQ  NUM
6110        CI   R1,>2D00     "-".
6120        JEQ  NUM
6130        CI   R1,>3000     ASCII 0.
6140        JLT  ERR
6150        CI   R1,>3900     ASCII 9.
6160        JGT  ERR
6170 * CONVERT DATA AT >0110 TO BINARY.
6180 NUM    LI   R1,>0110
6190        BLWP @>E004
6200 * CHECK IF NUMBER WAS IN CONVERSION RANGE.
6210        LI   R1,NMCK
6220        BLWP @>E00C       CONVERT BACK TO ASCII.
6230        LI   R7,>0110
6240        CLR  R1
6250        MOVB *R7,R1
6260        CI   R1,>2B00     "+".
6270        JNE  $+4          ****NPOS
6280        INC  R7
6290        LI   R8,NMCK      ****NPOS
6300        CB   *R7+,*R8+    ****VALID
6310        JEQ  $-2          ****VALID
6320        DEC  R7
6330        DEC  R8
6340        CB   *R7,@CRLF
6350        JNE  ERR
6360        CB   *R8,@LINS+4  (SPACE).
6370        JNE  ERR
6380 * NUMBERS AGREE.
6390        MOV  R0,@VARL(R10)
6400        JMP  SHOW
6410 RSET   LI   R10,VARD
6420        A    R5,R10
6430        JMP  TOP
6440 ERR    WRIT @ER
6450        JMP  ERRT
6460 EXIT   RTWP
6470 *
6480 *
6490 * HEX XOP ROUTINE.
6500 * ROUTINE ACCEPTS HEX ASCII & KEEPS LAST
6510 * FOUR DIGITS; TERMINATE WITH CR.
6520 HXOP   CLR  R0            HEX BUFFER.
6530        LI   R2,RDAT       RXOP CHARACTER ADDRESS STORE.
6540        LI   R4,>40        HEX A-F SYMBOL CHECK WORD.
6550        WRIT @KNUM
6560 NKY    READ @NKEY         CHECK INPUT AGAINST 0-9 & Q.
6570        MOV  *R2,R9        CHARACTER ADDRESS.
6580        MOV  *R9,R6        CHARACTER.
6590        CI   R6,>512E      "Q."
6600        JEQ  LEAV
6610        MOV  R11,R10       ADDRESS OF MEMORY RECALL DATA.
6620        SLA  R6,4
6630        SRL  R6,11         EXTRACT DATA FROM ASCII & *2.
6640        A    R6,R10
6650        MOV  *R10,R0
6660        LI   R1,LIND
6670        BLWP @>E010
6680        SLA  R6,7
6690        AI   R6,>3020
6700        MOV  R6,@LINS+2
6710 FALS   WRIT @LINS
6720        CLR  R8            FIRST DIGIT INDICATOR.
6730 HXIN   READ @KHEX
6740        MOV  *R2,R9
6750        MOV  *R9,R3
```

```
6760         CB   R3,@CRLF
6770         JEQ  CR
6780         CB   R3,@MKEY+20  "Q".
6790         JNE  $+6          ****NOTQ
6800         MOV  *R10,R0      REFILL R0.
6810         JMP  FALS
6820         SWPB R3           ****NOTQ
6830         COC  R4,R3
6840         JNE  NHEX
6850         AI   R3,9
6860 NHEX    ANDI R3,>F
6870         MOV  R8,R8        IF ZERO, CLEAR R0.
6880         JNE  $+6          ****NOT FIRST DIGIT
6890         CLR  R0
6900         SETO R8
6910         SLA  R0,4         ****NOT FIRST DIGIT
6920         A    R3,R0
6930         JMP  HXIN
6940 CR      LI   R5,6
6950         LI   R6,>2020
6960 CIRC    MOV  R6,@LIND(R5)
6970         DECT R5
6980         JLT  $+4          ****OUT
6990         JMP  CIRC
7000 * PUT HEX BACK IN STORE.
7010         MOV  R0,*R10      ****OUT
7020         LI   R1,LIND      BUFFER FOR BLWP @>E010.
?
7030         BLWP @>E010       CONVERT TO HEX ASCII.
7040         WRIT @LINS
7050         WRIT @MORE        ANY MORE CHANGES.
7060         JMP  MKY
7070 LEAV    WRIT @CRLF
7080         WRIT @CRLF
7090         RTWP
7100 *
7110 *
7120 G       TEXT 'G.'
7130         DATA >0D0A,>0D0A
7140         TEXT 'TYPE G TO PROCEED...'
7150         DATA 0
7160 BELL    DATA >0700        BELL
7170 QUES    TEXT 'PERIPHERAL STATUS QUESTIONS:'
7180         TEXT ' TYPE Y TO PROCEED. '
7190         DATA 0
7200 CHK1    DATA >0D0A,>0D0A
7210         TEXT 'IS KEYPAD UNIT POWERED UP? '
7220         BYTE 0
7230 CHK2    DATA >0D0A,>0D0A
7240         TEXT 'ARE 60V PSU''S POWERED UP? '
7250         DATA 0
7260 CHK3    DATA >0D0A,>0D0A
7270         TEXT 'IS INVERTER POWERED UP & 100KHZ ON? '
7280         DATA 0
7290 SYNY    TEXT 'Y.'
7300 VAR     DATA >0D0A,>0D0A
7310         TEXT 'DO YOU WISH TO MODIFY ANY LIMITS?'
7320         TEXT ' (Y/N): '
7330         BYTE 0
7340 CHOS    TEXT 'Y,N.'
7350 VTX1    DATA >0D0A
7360         TEXT 'MAX VOLTS V, START VOLTS B, MAX DELTA D,'
7370         TEXT ' MAX SPEED S, OR RAMP DELAY T?'
7380         DATA 0
7390 NOTE    DATA >0D0A,>0D0A
7400         TEXT '(MAX NUMERICAL INPUT IS /32767/).'
7410         BYTE 0
7420         DATA 0
7430 VTX2    DATA >0D0A,>0D0A
7440         TEXT 'TYPE SYMBOL OR E TO EXIT: '
7450         DATA 0
7460 METH    DATA >0D0A
7470         TEXT '** TYPE "CR" TO ENTER SYMBOL OR DATA **'
7480         BYTE 0
7490 RDAT    DATA 0
7500 LINS    DATA >0D0A,>0
7510         TEXT ' = '
7520 LIND    DATA 0,0,0,0,0
7530 VARD    TEXT 'V,B,D,S,T,E.'
7540         DATA 0
7550 VARL    DATA 30,7,0,30000,200 LIMIT VALUES.
7560 CRLF    DATA >0D0A,0
7570 ER      TEXT 'WHAT?'
7580         BYTE 0
7590 NMCK    DATA 0,0,0,0,0
7600 HQES    DATA >0D0A,>0D0A
7610         TEXT 'DO YOU WISH TO CHANGE ANY RECALL'
7620         TEXT ' MEMORY LOCATIONS? (Y/N): '
7630         DATA 0
7640 HXCR    DATA >0D0A
7650         TEXT '***IF INSPECTED DATA IS CORRECT, TYPE '
7660         TEXT '"CR"***'
7670         BYTE 0
7680 KNUM    DATA >0D0A,>0D0A
7690         TEXT 'TYPE KEY # 0-9 OR Q TO QUIT: '
7700         BYTE 0
```

```
7710 KHEX   DATA >0D2C        "CR,".
7720        TEXT 'A,B,C,D,E,F,'
7730 MKEY   TEXT '0,1,2,3,4,5,6,7,8,9,Q.'
7740 * RECALL ADDRESSES M0-M9.
7750 **WORKING VOLTS: WORKING LOADAMG: SPEED: ETC..
7760 M0     DATA >1004,>100C,>C000,0,0,0,0,0,0,0
7770 * DATA INTERRUPT STORAGE AREA ADDRESSES.
7780 ** ADDRESS ORDER: VIN; LDIN; SPEED; ETC..
7790        DATA 0,0,0,0,0,0  SET BY INITIALIZATION ROUTINE.
7800 MORE   DATA >0D0A
7810        TEXT 'ANY MORE? TYPE # OR Q: '
7820        BYTE 0
7830 KEYP   DATA >0D0A,>0D0A
7840        TEXT 'READY FOR KEYPAD? (Y/N): '
7850        BYTE 0
7860 FAIL   DATA >0707,>0707
7870        TEXT 'START ABORTED....TRY INCREMENTING '
7880        TEXT 'START VOLTS?'
7890        DATA 0
7900 STAL   DATA >0707,>0707
7910        TEXT 'MOTOR STALLED....INVESTIGATE REASON!'
7920        DATA 0
7930 *
7940 *
7950 *
7960 ** SUBROUTINE TO CLEAR VDU SCREEN.
7970 CLER   DATA >25C0        SUBRQUTINE WP.
7980        DATA $+2          PC.
7990        LI   R0,Y1
8000        BLWP @>E01C
8010        LI   R1,20        20MS REFRESH TIME.
8020        CLR  R0
8030 NULL   BLWP @>E018
8040        DEC  R1
8050        JGT  NULL
8060        RTWP
8070 *
8080 *
8090 * SUBPROGRAM TO DRAW LARGE VDU TITLES.
8100 BANN   DATA >2500        SUBROUTINE WP.
8110        DATA $+4
8120        DATA 0            MESSAGE IDENTIFIER STORE.
8130        MOV  @BANN+4,R9
8140        BLWP @CLER        CLEAR SCREEN.
8150        MOV  R9,R0
8160        AI   R0,4
8170        MOV  R0,@CHEK
8180        MOV  @DATK(R9),R8
8190 NXT    MOV  @DATB(R9),R3
8200        INCT R9
8210        MOV  @DATK(R9),R10
8220        DECT R9
8230        MOV  @DATJ(R9),R7
8240 NXT2   A    R7,R3
8250        MOV  @DATC(R8),R2
8260        MOV  @DATN(R8),R4
8270        MOV  @DATL(R8),R5
8280        BL   @CHS
8290        INCT R8
8300        C    R8,R10
8310        JLT  NXT2
8320        INCT R9
8330        C    R9,@CHEK
8340        JLT  NXT
8350        MOV  @DATB+8,R1
8360        LI   R2,>2000
8370        BL   @DRAW
8380        RTWP              RETURN TO MAIN PROG.
8390 CHS    CLR  R6
8400        MOV  R11,R12
8410 NXT1   MOV  R3,R1
8420        A    *R5+,R1
8430        BL   @DRAW
8440        INC  R6
8450        C    R6,R4
8460        JLT  NXT1
8470        B    *R12
8480 *
8490 * SUBROUTINE.
8500 DRAW   MOVB R1,@X
8510        MOV  R1,@Y
8520        MOVB @Z,@Y
8530        MOVB R2,@Y+2
8540        MOVB @Z+1,@Y+3
8550        LI   R0,DAT1
8560        BLWP @>E01C
8570        RT
8580 *
8590 *
8600 CHEK   DATA 0
8610 DAT1   DATA >1B58
8620 X      DATA >1B
8630 Y      BYTE 0,0,0,0,0
8640 Y1     BYTE >1B,>4B,0
8650 Z      BYTE >59,>08
8660 *
```

```
8670 **DAT2
8680         DATA >0705,>0604,>0503,>0402,>0403,>0404,>0305
8690         DATA >0205,>0104,>0103,>0102
8700 *
8710 **DAT3
8720         DATA >0701,>0601,>0501,>0401,>0301,>0201,>0101
8730         DATA >0202,>0303,>0403,>0204,>0105,>0205,>0305
8740         DATA >0405,>0505,>0605,>0705
8750 *
8760 **DAT4
8770         DATA >0605,>0704,>0703,>0702,>0601,>0501,>0401
8780         DATA >0301,>0201,>0102,>0103,>0104,>0205,>0305
8790         DATA >0405,>0505
8800 *
8810 **DAT5
8820         DATA >0101,>0105
8830 *
8840 **DAT6
8850         DATA >0104,>0103,>0102,>0203,>0303,>0403,>0503
8860         DATA >0603,>0703,>0704,>0702
8870 *
8880 **DAT7
8890         DATA >0705,>0604,>0503,>0402,>0303,>0204,>0105
8900 *
8910 **DAT8
8920         DATA >0701,>0601,>0501,>0401,>0301,>0201,>0101
8930         DATA >0102,>0103,>0104,>0105,>0402,>0403,>0702
8940         DATA >0703,>0704,>0705
8950 *
8960 **DAT9
8970         DATA >0703,>0603,>0503,>0403,>0304,>0204,>0105
8980         DATA >0302,>0202,>0101
8990 *
9000 **DAT0
9001         DATA >0402,>0403,>0404
9002 *
9003 **DATD
9004         DATA >0701,>0601,>0501,>0401,>0301,>0201,>0101
9005         DATA >0102,>0103,>0104,>0205,>0305,>0405,>0505
9006         DATA >0605,>0704,>0703,>0702
9007 *
9008 **DAT
9009         DATA >0701,>0601,>0501,>0401,>0301,>0201,>0102
9010         DATA >0103,>0104,>0205,>0305,>0405,>0505,>0605
9011         DATA >0705,>0402,>0403,>0404
9012 *
9013 **DATG
9014         DATA >0205,>0104,>0103,>0102,>0201,>0301,>0401
9015         DATA >0501,>0601,>0702,>0703,>0704,>0605,>0505
9016         DATA >0504
9017 *
9018 **DATS
9019         DATA >0101
9020 *
9021 DATK    DATA 0,10,20,30,42
9022 DATJ    DATA 14,14,14,12
9023 DATN    DATA 18,11,13,18,16
9024         DATA 18,16,11,16,18
9025         DATA 15,16,1,11,16
9026         DATA 14,17,10,15,18,18
9027 DATC    DATA >4D00,>4900,>4300,>5200,>4F00
9028         DATA >4D00,>4F00,>5400,>4F00,>5200
9029         DATA >4700,>4F00,>2000,>5400,>4F00
9030         DATA >4B00,>4500,>5900,>5000,>4100,>4400
9031 *
9032 **(DAT3,DAT6,DAT4,DAT2,DAT4)
9033 DATL    DATA $-374,$-302,$-338,$-396,$-338
9034 *
9035 **(DAT3,DAT4,DAT5,DAT4,DAT2)
9036         DATA $-384,$-348,$-316,$-348,$-406
9037 *
9038 **(DATG,DAT4,DATS,DAT5,DAT4)
9039         DATA $-154,$-358,$-124,$-326,$-358
9040 *
9041 **(DAT7,DAT8,DAT9,DAT0,DAT,DATD)
9042         DATA $-310,$-296,$-262,$-242,$-200,$-236
9043 *
?
9044 DATB    DATA >221A,>2C1A
9045         DATA >221A,>2C1A,>3621
9046         END  SQ7P
```

SOURCE LISTING OF THE MAGSLIP ROTOR-POSITION-CODE-BASED CONTROL PROGRAM.

```
0010 * PROGRAM TO RUN THE MAGSLIP SYSTEM WITH SYNCHRONISM
0020 * MAINTAINED BY SAMPLING A COUNTER AND DRIVING THE
0030 * INVERTER ACCORDINGLY.
0040 * NOVEMBER 19, 1981. N.M. MALLINSON.
0050 *
0060 *
0070 *** THIS PROGRAM SHOULD BE RUN ON A PERKIN ELMER VDU
0080 *** (PREFERABLY WITH 9600 BAUD RATE).
0090 *
0100 ** DATA & COMMANDS ARE ENTERED VIA A KEYPAD
0110 ** DURING MOTOR OPERATION.
0120 *
0130        IDT  'MAGCNT'
0140        AORG >1040
0150 * DEFINE REGISTER LABELS.
0160 R0     EQU  0
0170 R1     EQU  1
0180 R2     EQU  2
0190 R3     EQU  3
0200 R4     EQU  4
0210 R5     EQU  5
0220 R6     EQU  6
0230 R7     EQU  7
0240 R8     EQU  8
0250 R9     EQU  9
0260 R10    EQU  10
0270 R11    EQU  11
0280 R12    EQU  12
0290 R13    EQU  13
0300 R14    EQU  14
0310 R15    EQU  15
0320 MAG    LWPI >1000
0330 *
0340        CLR  @ESC          **CLEAR VDU FLAG.
0350 *
0360 * SET ALL INPORTANT OUTPUT PORTS SAFE.
0370        CLR  @>C00E        VOLTS.
0380        SETO @>C00C        INVERTER CONTROL.
0390        CLR  @>C010        INVERTER SELECT.
0400        CLR  @>C006        DISPLAY PORT.
0410        CLR  @>C012          "     "
0420        CLR  @>C012          "     "
0430 *
0440 *
0450 **VECTOR VARI DATA ADDRESSES FOR DISPLAY ROUTINE.
0460        LI   R1,M0
0470        AI   R1,20
0480        LI   R0,VARI
0490        MOV  R0,*R1+       ****DATA VECTOR
0500        INCT R0
0510        CI   R1,MORE       END OF ADDRESS BLOCK.
0520        JLT  $-8           ****DATA VECTOR
0530 *
0540 *
0550 * VECTOR INPUT, READ,WRITE,
0560 * MODIFY & HEX MOD XOP'S.
0570        DXOP INPU,4
0580        LI   R0,>25A0      XOP4 WP.
0590        MOV  R0,@>0050
0600        LI   R0,IXOP       XOP4 PC.
0610        MOV  R0,@>0052
0620        DXOP READ,5
0630        LI   R0,>2520      XOP5 WP.
0640        MOV  R0,@>0054
0650        LI   R0,RXOP       XOP5 PC.
0660        MOV  R0,@>0056
0670        DXOP WRIT,6
0680        LI   R0,>2540      XOP6 WP.
0690        MOV  R0,@>0058
0700        LI   R0,WXOP       XOP6 PC.
0710        MOV  R0,@>005A
0720        DXOP MOD,0
0730        LI   R0,>2560      XOP0 WP.
0740        MOV  R0,@>0040
0750        LI   R0,MXOP       XOP0 PC.
0760        MOV  R0,@>0042
0770        DXOP HEX,1
0780        LI   R0,>2580      XOP1 WP.
0790        MOV  R0,@>0044
0800        LI   R0,HXOP       XOP1 PC.
0810        MOV  R0,@>0046
0820 *
0830        CLR  @BANN+4       **MESSAGE IDENTIFIER.
0840        BLWP @BANN         WRITE VDU TITLE.
0850        WRIT @G+2          **GO? MESSAGE.
0860        READ @G
0870        BLWP @CLER         CLEAR SCREEN.
0880        WRIT @QUES         PERIPHERAL STATUS QUESTIONS.
0890        WRIT @CHK1
0900        READ @SYMY
0910        WRIT @CHK2
0920        READ @SYMY
0930        WRIT @CHK3
0940        READ @SYMY
0950 OPT    BLWP @CLER         CLEAR SCREEN.
```

```
0960        WRIT @VAR          IS CHANGE IN LIMIT VALUES WANTED?
0970        READ @CHOS         YES/NO CHOICE.
0980        LI   R0,CHOS
0990        C    @RDAT,R0
1000        JNE  $+22          ****CONTINUE
1010        WRIT @VTX1
1020        WRIT @NOTE
1030        WRIT @VTX2
1040        READ @VARD
1050        MOD  @RDAT
1060 * CLEAR SCREEN.
1070        BLWP @CLER         ****CONTINUE
1080        WRIT @HQES         CHANGE RECALL MEMORY LOCATIONS?
1090        READ @CHOS
1100        LI   R0,CHOS
1110        C    @RDAT,R0
1120        JNE  $+10          ****KEYPAD
1130        WRIT @HXCR
1140        HEX  @M0           MODIFY DATA AT LOCATION M0-M9.
1150        WRIT @KEYP         ****KEYPAD
1160        READ @CHOS
1170        C    @RDAT,R0
1180        JNE  OPT
1190        LI   R0,4
1200        MOV  R0,@BANN+4    **MESSAGE IDENTIFIER.
1210        BLWP @BANN
1220 *
1230 *
1240 **********************
1250 * MAIN MOTOR PROGRAM.
1260 * COMPLETE INITIALISATION OF SYSTEM.
1270 *
1280 * VECTOR 9900 INTERRUPTS 8,9,10.
1290 * INTERRUPTS ARE (9900 # IN BRACKETS):
1300 * INT6=STOP (8).
1310 * INT7=DATA (9).
1320 * INT8=DISP (10).
1330 *
1340 * THE WORKSPACES FOR THE MAIN PROGRAM & STOP
1350 * ARE OVERLAPPED TO EASE DATA TRANSFER.
1360        LI   R1,INFO
1370        LI   R2,ADDR
1380        MOV  *R2+,R3       ****VECTOR
1390        MOV  *R1+,*R3
1400        MOV  *R1,*R1       COMPARE TO ZERO.
1410        JNE  $-6           ****VECTOR
1420        JMP  CLRI
1430 * INFORMATION ARRANGED IN FORMAT WP6;PC6;WP7,PC7; ETC..
1440 *
1450 INFO   DATA >1000,STOP,>25E0,DATA,>2600,DISP
1460        DATA 0             ***END OF VECTORING.
1470 *
1480 * ADDRESSES IN FORMAT INT1 WP STORE,
1490 * INT1 PC STORE, ETC..
1500 ADDR   DATA >0020,>0022,>0024,>0026,>0028,>002A
1510 *
1520 CLRI   CKON               CLEAR ALL INTERRUPT LATCHES.
1530        CKOF
1540        LI   R1,>C014
1550        CLR  *R1+          ****WIPE
1560        CI   R1,>C020      STOP VALUE.
1570        JLT  $-6           ****WIPE
1580 *
1590 * SET UP INTERRUPTS.
1600        LI   R12,>100      CRU BASE.
1610        LI   R1,>700       ENABLE 9900 INT'S 8,9,10.
1620        LDCR R1,11         SET INT MODE.
1630 *
1640 * SET DEMANDED VOLTS TO START VALUE & OTHER
1650 * VARIABLE INPUTS TO ZERO FOR SAFETY.
1660        MOV  @VARL+2,@VARI
1670        LI   R1,8
1680        CLR  @VARI+2(R1)   ****CLEAR
1690        DECT R1
1700        JLT  $+4           ****CLRFIN
1710        JMP  $-8           ****CLEAR
1720 *
1730 *
1740 * MAIN PROGRAM.
1750 VRQN   EQU  R1
1760 VRQD   EQU  R2
1770 DANW   EQU  R3
1780 DAOL   EQU  R4
1790 LRQN   EQU  R5
1800 LRQD   EQU  R6
1810 POIN   EQU  R9
1820 LDAN   EQU  R10
1830 CRUB   EQU  R12
1840 * INITIALISE KEYPAD, INVERTER & PSU CRUBIT.
1850        LI   CRUB,>120     ****CLRFIN
1860        SBZ  8
1870 *
1880        CLR  VRQN
1890        CLR  VRQD
1900        CLR  DANW
1910        CLR  DAOL
```

```
1920        CLR  LRQN
1930        CLR  @SLOW
1940        CLR  @>C00E
1950 *
1960 * NOW WAIT FOR STOP, DATA OR DISP INT.
1970 REST   C    @NINE,@ESC
1980        JNE  $+32          ****NOT VDU
1990        LIMI 0
2000 * DISABLE INVERTER.
2010        LI   CRUB,>120
2020        SBO  8
2030 **RING BELL 15 TIMES.
2040        LI   R0,>070F
2050        BLWP @>E018        ****LOOK AT VDU!
2060        DEC  R0
2070        CI   R0,>0700
2080        JGT  $-10          ****LOOK AT VDU!
2090        B    @MAG
2100        LIMI 10            ****NOT VDU
2110 *
2120        JMP  REST          WAIT FOR START INT.
2130 * CHECK REQUIRED DIRECTION OF ROTATION.
2140 AWAY   TB   9             CRUBIT >129.
2150        JNE  $+8           ***FWD
2160 * REVERSE SELECTED.
2170        LI   LDAN,DRVR
2180        JMP  $+6           ****HOP
2190        LI   LDAN,DRVF     ****FWD
2200        MOV  LDAN,@SLOW+8  STORE DIRECTION INFO.
2210        MOV  @VARL+2,VRQO
2220        MOVB @VTAB(VRQO),DAOL
2230        SRL  DAOL,8
2240        AI   DAOL,>6000
2250        MOV  DAOL,@>C00E   SET START VOLTS.
2260        CLR  LRQO
2270 * CHECK MOTOR IS STATIONARY PRIOR TO START.
2280        MOV  @>C000,R8     ****SPEED
2290        CI   R8,16
2300        JGT  $-8           ****SPEED
2310 ********************************************
2320 * SYNCHRONIZATION ROUTINE.
2330 * STEP THE MOTOR & SAMPLE THE COUNT: IF ZERO NOT FOUND
2340 * WITHIN 9 STEPS ASSUME MOTOR IS STALLED & ABORT START.
2350 * WHEN ZERO FOUND STEP MOTOR FOR A REVOLUTION TO CHECK
2360 * ZERO REOCCURS WHEN EXPECTED. ALSO DURING THIS REVOLUTION
2370 * CHECK ROTOR HAS ACTUALLY MOVED & HAS NOT JUST STUCK AT
2380 * ZERO; THIS IS DONE BY SAMPLING THE COUNT AT THE END OF
2390 * THE THIRD PULSE TO CHECK IT IS NON ZERO. THEN CHECK FOR
2400 * ZERO AT THE END OF THE 5TH, 6TH OR 7TH PULSE; (STRICTLY
2410 * IT SHOULD ONLY BE NECESSARY TO CHECK AT THE END OF THE
2420 * SIXTH PULSE BUT THE SLIGHTLY RANDOM NATURE OF THE MOTOR
2430 * MOVEMENT CAN LEAD TO THE ZERO APPEARING AT THE END OF
2440 * THE 5TH OR 7TH PULSE).
2450 * IF ZERO NOT FOUND IN THE CORRECT PLACE RETRY START UP
2460 * TO A MAXIMUM OF 5 TIMES: ABORT START OTHERWISE.
2470        LI   R0,5
2480        MOV  R0,@TRY       **STARTING ATTEMPT STORE.
2490        CLR  @TRY+2        ***STALL DETECT FLAG.
2500 REGO   LI   R0,9          MAX # OF STARTING STEPS.
2510        MOV  @SLOW+8,R7    INVERTER TABLE ADDRESS.
2520 * STEP UNTIL ZERO COUNT.
2530 NXVL   LI   R8,>FFFF
2540        MOV  @SLOW,@SLOW   CHECK FOR STOP.
2550        JNE  BRAK
2560        CI   R0,10
2570        JEQ  SAMP          CHECK IF ZERO INITIALLY.
2580        MOV  *R7+,@>C00C   DRIVE INVERTER.
2590        INCT R7            INCREMENT INVERTER POINTER.
2600 *(NO NEED TO CHECK FOR END OF TABLE SINCE ABORT OCCURS
2610 * BEFORE END OF TABLE REACHED).
2620 *
2630 * MOVE RELEVANT BYTE [>C009] TO DETECT ZERO COUNT.
2640 SAMP   MOVB @>C009,POIN   SAMPLE POSITION.
2650        JEQ  CNTO
2660        DEC  R8
2670        JNE  SAMP
2680        DEC  R0
2690        JNE  NXVL          END OF START ATTEMPT?
2700 *
2710 * ABORT START.
2720 ABOR   LIMI 0
2730 * DISABLE INVERTER.
2740        LI   CRUB,>120
2750        SBO  8
2760        SETO @>C00C
2770        CLR  @>C00E
2780        BLWP @CLER
2790        MOV  @TRY+2,@TRY+2 CHECK STALL FLAG.
2800        JEQ  $+12          ****NO STALL
2810        WRIT @STAL
2820        CLR  @TRY+2        RESET FLAG.
2830        JMP  $+6           ****GO? MESSAGE
2840        WRIT @FAIL         ****NO STALL
2850        WRIT @G+2          ****GO? MESSAGE
2860        READ @G
2870        B    @OPT
```

```
2880 *
2890 DELY   LI   R8,>FFFF
2900 * CHECK IF STOP REQUIRED.
2910        MOV  @SLOW,@SLOW  ****DELAY
2920        JNE  BRAK
2930        DEC  R8
2940        JNE  $-10         ****DELAY
2950        RT
2960 *
2970 **CHECK NOW THAT ZERO REPEATS CORRECTLY.
2980 **ROTATE MOTOR ON FOR FURTHER REVOLUTION.
2990 *** FIRST WAIT FOR ROTOR TO STOP MOVING.
3000 CNT0   BL   @DELY
3010        LI   R0,4         STEP COUNTER
3020        MOV  *R7+,@>C00C  ****STEP
3030        JGT  $+6          ****NO OVERFLOW
3040        AI   R7,-24       RESET TABLE POINTER.
3050 * ENSURE SUCCESSIVE INVERTER OUTPUTS CHANGE.
3060        INCT R7           ****NO OVERFLOW
3070        BL   @DELY
3080 * CHECK IF FIFTH PULSE IS NEXT.
3090        DEC  R0
3100        JEQ  PLS5         PULSE 5 NEXT.
3110        CI   R0,1         3RD PULSE?
3120        JNE  $+8          ****STEP
3130 * CHECK ROTOR HAS MOVED FROM ZERO.
3140        MOVB @>C009,POIN
3150        JEQ  TRIP
3160        B    @CNT0+8      ****STEP
3170 *
3180 PLS5   LI   R8,>FFFF
3190        MOV  *R7+,@>C00C
3200        JGT  $+6          ****NOV
3210        AI   R7,-24
3220 IN     MOVB @>C009,POIN  ****NOV
3230        JEQ  MAIN
3240        DEC  R8
3250        JNE  IN
3260        DEC  R0
3270        CI   R0,-3        END OF REV?
3280        JEQ  TRIP
3290 * ENSURE INVERTER OUTPUT IS INCREMENTED.
3300        INCT R7
3310        JMP  PLS5         OUTPUT 6TH/7TH PULSE.
3320 TRIP   BL   @DELY
3330        DEC  @TRY
3340        JGT  REGO         5 STARTS ATTEMPTED.
3350        B    @ABOR
3360 *
3370 TRY    DATA 0,0          *STARTING & STALL DETECT STORES.
3380 ***SYNCHRONIZED.
3390 *
3400 * DISABLE STALL CHECK WHILST MOTOR SPEEDS UP.
3410 MAIN   LI   R0,>FFF
3420        MOV  R0,@SLOW+10  STALL DISABLE-TIME STORE.
3430 *
3440 * MUST CHECK COUNTER STATE AT LEAST EVERY 400US
3450 * AT 12000 RPM: READ IN ROTOR POSITION.
3460        MOV  @>C008,POIN  ****MAIN TOP
3470 * CHECK INPUT COUNT IS VALID.
3480 * >C009=LS BYTE OF COUNT; >1013=LS BYTE OF POIN.
3490        CB   @>C009,@>1013
3500        JNE  MAIN
3510        ANDI POIN,>1E
3520        A    LDAN,POIN
3530        MOV  *POIN,@>C00C
3540        MOV  @SLOW,@SLOW  CHECK FOR STOP.
3550        JEQ  INPT
3560 * CLEAR VOLTS AND WAIT FOR SYSTEM TO RAMP DOWN.
3570        CLR  VRQN
3580        CI   DAOL,>6000   ZERO VOLTS.
3590        JGT  DIFF
3600 BRAK   LIMI 0
3610        CLR  @SLOW
3620        CLR  @>C00E
3630        SETO @>C00C
3640 **NOTE STOP/START FLIPFLOP IN KEYPAD MAY BE
3650 **SET TO START WHILST MOTOR STOPPING; TO AVOID
3660 **TROUBLE WHEN KEY NEXT PRESSED (GENERATING A
3670 **STOP) CLEAR FLIPFLOP AT END OF A STOP.
3680        LI   CRUB,>120
3690        SBO  8
3700        SBZ  8
3710        CLR  @>C01A       CLEAR STOP INT.
3720        LI   CRUB,>100
3730        LI   POIN,>700
3740        LDCR POIN,11      REENABLE START INT.
3750        LI   CRUB,>120
3760 * SET INITIAL RUNNING VOLTS TO START VALUE.
3770 * (CAN BE CHANGED THOUGH DURING STOPPED TIME).
3780        MOV  @VARL+2,@VARI
3790        B    @REST
3800 *
3810 INPT   MOV  @VARI,VRQN   INPUT VOLTS.
3820 DIFF   C    VRQN,VRQO
3830        JEQ  ANG
```

```
* CHECK IF VRQN IS IN LIMITS.
       C    VRQN,@VARL
       JLT  $+8          ****OKAY
       JEQ  $+6          ***OKAY
       MOV  @VARL,VRQN
       MOVB @VTAB(VRQN),DANW ****OKAY
       SRL  DANW,8
       AI   DANW,>6000
       C    DANW,DAOL
       JEQ  UPDA         UPDATE REGISTERS.
       JGT  INCR
*DECREMENT VOLTS.
       DEC  @SLOW+4      ****DOWN
       JGT  ANG
       DEC  DAOL
       MOV  @VARL+8,@SLOW+4 ****DOWN
       MOV  DAOL,@>C00E  OUTPUT NEW VOLTS.
       JMP  ANG
INCR   DEC  @SLOW+6      ****UP
       JGT  ANG
       INC  DAOL
       MOV  @VARL+8,@SLOW+6 ****UP
       MOV  DAOL,@>C00E  OUTPUT NEW VOLTS.
       JMP  ANG
UPDA   MOV  DANW,DAOL
       MOV  VRQN,VRQO
* LOAD ANGLE CHANGE SECTION.
*(FIRST CHECK ROTOR POSITION).
ANG    MOV  @>C008,POIN
       CB   @>C009,@>1013 CHECK IF DATA VALID.
       JNE  ANG
       ANDI POIN,>1E
       A    LDAN,POIN
       MOV  *POIN,@>C00C
       MOV  @VARI+2,LRQN INPUT LOAD ANGLE.
       C    LRQN,LRQO
       JEQ  BOT          NOW CHECK FOR STALL.
       C    LRQN,@VARL+4 MAX LOAD ANGLE.
       JLT  $+8          ***FINE
       JEQ  $+6          ***FINE
       MOV  @VARL+4,LRQN
       S    LRQO,LRQN    ****FINE
       CI   LRQN,15
       JLT  FALL
       INCT LDAN
       AI   LRQO,30
       JMP  BOT          NOW CHECK FOR STALL.
FALL   CI   LRQN,-15
       JGT  BOT          CHECK FOR STALL.
       DECT LDAN
       AI   LRQO,-30
BOT    MOV  @SLOW+10,@SLOW+10 STALL CHECK DISABLED?
       JEQ  $+10         ****STALL?
       DEC  @SLOW+10
       B    @MAIN+8      ****MAIN TOP
       MOV  @>C000,R0    ****STALL?
       CI   R0,16        MINIMUM SPEED.
       JGT  $+10         ****(MAIN TOP)
       SETO @TRY+2       STALL FLAG.
       B    @ABOR
* RETURN TO TOP.
       B    @MAIN+8      ****MAIN TOP
*
SLOW   DATA 0,-1
       DATA 0,0          **DOWN/UP.
       DATA 0            DIRECTION DATA STORE.
       DATA 0            STALL DELAY-TIME STORE.
*
*
VTAB   BYTE 0,5,10,15,20,26
       BYTE 31,36,41,46,51,56,61,66
       BYTE 71,77,82,87,92,97,102
       BYTE 107,112,117,122,128,133
       BYTE 138,143,148,153,158,163
       BYTE 168,173,179,184,189,194
       BYTE 199,204,209,214,219,224
?
       BYTE 230,235,240,245,250,255,0
*
*
* INVERTER DRIVE SIGNALS TABLE.
*
*
DRVF   DATA >001D,>001D,>001E,>001E,>002E,>002E
       DATA >002B,>002B,>0033,>0033,>0035,>0035
       DATA >801D,>801D,>801E,>801E,>802E,>802E
       DATA >802B,>802B,>8033,>8033,>8035,>8035
*
* REVERSE TABLE SHIFTED BY 180 DEG.
*
DRVR   DATA >002E,>002E,>001E,>001E,>001D,>001D
       DATA >0035,>0035,>0033,>0033,>002B,>002B
       DATA >802E,>802E,>801E,>801E,>801D,>801D
       DATA >8035,>8035,>8033,>8033,>802B,>802B
*
*********************************
```

```
4790 *INTERRUPT SERVICE ROUTINES.
4800 *
4810 * STOP *
4820 * IN SAME WP AS MAIN PROG: IN STOPPED MODE WAITING
4830 * FOR START SIGNAL MAKE SURE VOLTS & LOAD ANGLE ARE
4840 * SET TO VSTART & ZERO RESPECTIVELY.
4850 * WHEN EXECUTING A "STOP" REQUEST ENSURE THAT A START
4860 * IS INHIBITED UNTIL "STOP" COMPLETE.
4870 STOP   TB   10          CHECK IF STOP/START.
4880        JNE  STRT
4890 * SHUT DOWN MOTOR.
4900        MOV  @SLOW+2,@SLOW SET SLOW=-1.
4910        CLR  @VARI+2      RESET INPUT LOAD ANGLE.
4920 ** DISABLE STOP INT, BUT DO NOT CLEAR.
4930        LI   CRUB,>100
4940        SBZ  8            MASK STOP INT.
4950        LI   CRUB,>120
4960 ** CLEAR STOP INT WHEN STOP COMPLETED.
4970        RTWP
4980 STRT   LIMI 0
4990        LI   R14,AWAY     RETURN ADDRESS.
5000 * ENSURE INT MASK IS SET SO THAT IF DATA OR DISPLAY
5010 * WERE INTERRUPTED BY START, THEY CAN RECALL MICRO.
5020        LI   R15,10
5030        CLR  @>C01A       CLEAR INT.
5040        RTWP              RETURN TO AWAY.
5050 *
5060 * DATA *
5070 DATA   LIMI 0
5080        MOV  @>C002,R0    DATA WORD.
5090        MOV  @>C004,R1    KEY WORD.
5100        LIMI 8            ENABLE INT'S.
5110        ANDI R1,>7        CLEAR SPURIOUS BITS.
5120        DECT R1           CORRECT KEY #.
5130        SLA  R1,1         *2.
5140        MOV  R0,@VARI(R1) PUT DATA IN STORE.
5150        CLR  @>C01C       CLEAR DATA INT.
5160        RTWP
5170 *
5180 *
5190 * DATA INTERRUPT STORAGE AREA.
5200 ** DATA ORDER: VIN; LDIN; SPEED; ETC..
5210 VARI   DATA 0,0,0,0,0,0
5220 *
5230 *
5240 * DISP *
5250 DISP   MOV  @>C004,R1    KEY WORD.
5260 * (TRANSFER DATA IN FIRST PROTECTED INSTRUCTION).
5270        CLR  @>C006
5280        ANDI R1,>F
5290        CI   R1,9         RETURN TO VDU?
5300        JNE  $+8          **DISPLAY WANTED.
5310        MOV  R1,@ESC      VDU WANTED FLAG.
5320 * (ESCAPE AFTER NEXT STOP INT).
5330        JMP  $+16         ***QUIT
5340        MOV  R1,@>C012    TRANSFER KEY.
5350        SLA  R1,1         *2
5360        MOV  @M0(R1),R2
5370        MOV  *R2,@>C012
5380 * CLEAR DISP INT.
5390        CLR  @>C01E       ****QUIT
5400        RTWP
5410 ESC    DATA 0            **VDU FLAG.
5420 NINE   DATA 9            **VDU CHECK DATA.
5430 *
5440 *
5450 RXOP   MOV  R11,R1       READ XOP ROUTINE.
5460        INPU R0           READ CHARACTER.
5470        MOV  *R1+,R2      ****NXCH
5480        MOV  R2,R3
5490        ANDI R2,>FF00
5500        ANDI R3,>FF
5510        C    R0,R2
5520        JEQ  $+14         ****ECHO
5530        CI   R3,>002E     FULL STOP.
5540        JNE  $-20         ****NXCH
5550 * NO VALID CHARACTER: DO NOT ECHO; RING BELL.
5560        WRIT @BELL
5570        JMP  RXOP
5580        DECT R1           ***ECHO
5590        MOV  R1,@RDAT     CHARACTER ADDR. PUT AT RDAT.
5600        BLWP @>E018
5610        RTWP
5620 *
5630 *
5640 IXOP   LI   R12,>80      INPUT CHARACTER XOP ROUTINE.
5650        TB   21
5660        JNE  IXOP
5670        CLR  *R11
5680        STCR *R11,8
5690        SBZ  18
5700        RTWP
5710 *
5720 *
```

```
5730 WXOP   MOV  R11,R0       WRITE XOP ROUTINE.
5740        BLWP @>E01C
5750        RTWP
5760 *
5770 *
5780 MXOP   MOV  *R11,R10     MODIFY XOP ROUTINE.
5790 * R10 CONTAINS ADDRESS OF SYMBOL.
5800        MOV  *R10,R1      R1 CONTAINS SYMBOL.
5810        CI   R1,>452E     "E.".
5820        JEQ  EXIT
5830        WRIT @METH
5840 TOP    ANDI R1,>FF00
5850        AI   R1,>20       R1=SYM+SPACE.
5860        MOV  R1,@LINS+2
5870        LI   R3,VARD
5880        S    R3,R10       MOD #.
5890 SHOW   MOV  @VARL(R10),R0
5900        LI   R5,6
5910        LI   R6,>2020
5920        MOV  R6,@LIND(R5) ****LOOP
5930 * (DATA BUFFER + R5)
5940        DECT R5
5950        JLT  $+4          ****FIN
5960        JMP  $-8          ****LOOP
5970        LI   R1,LIND      ****FIN
5980 * (LIND IS BUFFER FOR BLWP @>E00C.)
5990        BLWP @>E00C       CONVERT TO DEC. ASCII.
6000 ERRT   WRIT @LINS
6010        BLWP @>E020       READ IN STRING.
6020 * TERMINATE WITH CR; THIS IS NOT ECHOED.
6030        WRIT @CRLF
6040 * CHECK IF VARD SYMBOL ENTERED.
6050        CLR  R1
6060        MOVB @>0110,R1
6070        CI   R1,>4500     "E".
6080        JEQ  EXIT
6090        LI   R5,8
6100        CB   R1,@VARD(R5) ****COMPARE
6110        JEQ  RSET
6120        DECT R5
6130        JLT  $+4          ****NOMATCH
6140        JMP  $-10         ****COMPARE
6150 * CHECK FOR VALID NUMBER INPUT.
6160        CLR  R1           ****NOMATCH
6170        MOVB @>0110,R1
6180        CI   R1,>2B00     "+".
6190        JEQ  NUM
6200        CI   R1,>2D00     "-".
6210        JEQ  NUM
6220        CI   R1,>3000     ASCII 0.
6230        JLT  ERR
6240        CI   R1,>3900     ASCII 9.
6250        JGT  ERR
6260 * CONVERT DATA AT >0110 TO BINARY.
6270 NUM    LI   R1,>0110
6280        BLWP @>E004
6290 * CHECK IF NUMBER WAS IN CONVERSION RANGE.
6300        LI   R1,NMCK
6310        BLWP @>E00C       CONVERT BACK TO ASCII.
6320        LI   R7,>0110
6330        CLR  R1
6340        MOVB *R7,R1
6350        CI   R1,>2B00     "+".
6360        JNE  $+4          ****NPOS
6370        INC  R7
6380        LI   R8,NMCK      ****NPOS
6390        CB   *R7+,*R8+    ****VALID
6400        JEQ  $-2          ****VALID
6410        DEC  R7
6420        DEC  R8
6430        CB   *R7,@CRLF
6440        JNE  ERR
6450        CB   *R8,@LINS+4  (SPACE).
6460        JNE  ERR
6470 * NUMBERS AGREE.
6480        MOV  R0,@VARL(R10)
6490        JMP  SHOW
6500 RSET   LI   R10,VARD
6510        A    R5,R10
6520        JMP  TOP
6530 ERR    WRIT @ER
6540        JMP  ERRT
6550 EXIT   RTWP
6560 *
6570 *
6580 * HEX XOP ROUTINE.
6590 * ROUTINE ACCEPTS HEX ASCII & KEEPS LAST
6600 * FOUR DIGITS; TERMINATE WITH CR.
6610 HXOP   CLR  R0           HEX BUFFER.
6620        LI   R2,RDAT      RXOP CHARACTER ADDRESS STORE.
6630        LI   R4,>40       HEX A-F SYMBOL CHECK WORD.
6640        WRIT @KNUM
6650 NKY    READ @NKEY        CHECK INPUT AGAINST 0-9 & Q.
6660        MOV  *R2,R9       CHARACTER ADDRESS.
6670        MOV  *R9,R6       CHARACTER.
6680        CI   R6,>512E     "Q."
6690        JEQ  LEAV
```

```
6700         MOV  R11,R10      ADDRESS OF MEMORY RECALL DATA.
6710         SLA  R6,4
6720         SRL  R6,11        EXTRACT DATA FROM ASCII & *2.
6730         A    R6,R10
6740         MOV  *R10,R0
6750         LI   R1,LIND
6760         BLWP @>E010
6770         SLA  R6,7
6780         AI   R6,>3020
6790         MOV  R6,@LIMS+2
6800 FALS    WRIT @LIMS
6810         CLR  R8           FIRST DIGIT INDICATOR.
6820 HXIN    READ @KHEX
6830         MOV  *R2,R9
6840         MOV  *R9,R3
6850         CB   R3,@CRLF
6860         JEQ  CR
6870         CB   R3,@MKEY+20  "Q".
6880         JNE  $+6          ****NOTQ
6890         MOV  *R10,R0      REFILL R0.
6900         JMP  FALS
6910         SWPB R3           ****NOTQ
6920         COC  R4,R3
6930         JNE  NHEX
6940         AI   R3,9
6950 NHEX    ANDI R3,>F
6960         MOV  R8,R8        IF ZERO, CLEAR R0.
6970         JNE  $+6          ****NOT FIRST DIGIT
6980         CLR  R0
6990         SETO R8
7000         SLA  R0,4         ****NOT FIRST DIGIT
?
7010         A    R3,R0
7020         JMP  HXIN
7030 CR      LI   R5,6
7040         LI   R6,>2020
7050 CIRC    MOV  R6,@LIND(R5)
7060         DECT R5
7070         JLT  $+4          ****OUT
7080         JMP  CIRC
7090 * PUT HEX BACK IN STORE.
7100         MOV  R0,*R10      ****OUT
7110         LI   R1,LIND      BUFFER FOR BLWP @>E010.
7120         BLWP @>E010       CONVERT TO HEX ASCII.
7130         WRIT @LIMS
7140         WRIT @MORE        ANY MORE CHANGES.
7150         JMP  MKY
```

```
7160 LEAV    WRIT @CRLF
7170         WRIT @CRLF
7180         RTWP
7190 *
7200 *
7210 G       TEXT 'G.'
7220         DATA >0D0A,>0D0A
7230         TEXT 'TYPE G TO PROCEED...'
7240         DATA 0
7250 BELL    DATA >0700        BELL
7260 QUES    TEXT 'PERIPHERAL STATUS QUESTIONS:'
7270         TEXT ' TYPE Y TO PROCEED. '
7280         DATA 0
7290 CHK1    DATA >0D0A,>0D0A
7300         TEXT 'IS KEYPAD UNIT POWERED UP? '
7310         BYTE 0
7320 CHK2    DATA >0D0A,>0D0A
7330         TEXT 'ARE 60V PSU''S POWERED UP? '
7340         DATA 0
7350 CHK3    DATA >0D0A,>0D0A
7360         TEXT 'IS INVERTER POWERED UP & 100KHZ ON? '
7370         DATA 0
7380 SYNY    TEXT 'Y.'
7390 VAR     DATA >0D0A,>0D0A
7400         TEXT 'DO YOU WISH TO MODIFY ANY LIMITS?'
7410         TEXT ' (Y/N): '
7420         BYTE 0
7430 CHOS    TEXT 'Y,N.'
7440 VTX1    DATA >0D0A
7450         TEXT 'MAX VOLTS V, START VOLTS B, MAX DELTA D,'
7460         TEXT ' MAX SPEED S, OR RAMP DELAY T?'
7470         DATA 0
7480 NOTE    DATA >0D0A,>0D0A
7490         TEXT '(MAX NUMERICAL INPUT IS /32767/).'
7500         BYTE 0
7510 VTX2    DATA >0D0A,>0D0A
7520         TEXT 'TYPE SYMBOL OR E TO EXIT: '
7530         DATA 0
7540 METH    DATA >0D0A
7550         TEXT '** TYPE "CR" TO ENTER SYMBOL OR DATA **'
7560         BYTE 0
7570 RDAT    DATA 0
7580 LIMS    DATA >0D0A,>0
7590         TEXT ' = '
7600 LIND    DATA 0,0,0,0,0
7610 VARD    TEXT 'V,B,D,S,T,E.'
7620         DATA 0
```

```
7630 VARL   DATA 30,7,0,30000,200 LIMIT VALUES.
7640 CRLF   DATA >0D0A,0
7650 ER     TEXT 'WHAT?'
7660        BYTE 0
7670 NMCK   DATA 0,0,0,0,0
7680 HQES   DATA >0D0A,>0D0A
7690        TEXT 'DO YOU WISH TO CHANGE ANY RECALL'
7700        TEXT ' MEMORY LOCATIONS? (Y/N): '
7710        DATA 0
7720 HXCR   DATA >0D0A
7730        TEXT '***IF INSPECTED DATA IS CORRECT, TYPE '
7740        TEXT '"CR"***'
7750        BYTE 0
7760 KNUM   DATA >0D0A,>0D0A
7770        TEXT 'TYPE KEY # 0-9 OR Q TO QUIT: '
7780        BYTE 0
7790 KHEX   DATA >0D2C        "CR,".
7800        TEXT 'A,B,C,D,E,F,'
7810 KKEY   TEXT '0,1,2,3,4,5,6,7,8,9,Q.'
7820 * RECALL ADDRESSES M0-M9.
7830 **WORKING VOLTS: WORKING LOADANG: SPEED: ETC..
7840 M0     DATA >1004,>100C,>C000,0,0,0,0,0,0,0
7850 * DATA INTERRUPT STORAGE AREA ADDRESSES.
7860 ** ADDRESS ORDER: VIN; LDIN; SPEED; ETC..
7870        DATA 0,0,0,0,0,0  SET BY INITIALIZATION ROUTINE.
7880 MORE   DATA >0D0A
7890        TEXT 'ANY MORE? TYPE # OR Q: '
7900        BYTE 0
7910 KEYP   DATA >0D0A,>0D0A
7920        TEXT 'READY FOR KEYPAD? (Y/N): '
7930        BYTE 0
7940 FAIL   DATA >0707,>0707
7950        TEXT 'START ABORTED.....TRY INCREMENTING'
7960        TEXT ' STARTING VOLTS?'
7970        DATA 0
7980 STAL   DATA >0707,>0707
7990        TEXT 'MOTOR STALLED...INVESTIGATE REASON!!'
8000        DATA 0
8010 *
8020 *
8030 *
8040 ** SUBROUTINE TO CLEAR VDU SCREEN.
8050 CLER   DATA >25C0        SUBROUTINE WP.
8060        DATA $+2          PC.
8070        LI   R0,Y1
8080        BLWP @>E01C
8090        LI   R1,20        20MS REFRESH TIME.
8100        CLR  R0
8110 NULL   BLWP @>E018
8120        DEC  R1
8130        JGT  NULL
8140        RTWP
8150 *
8160 *
8170 * SUBPROGRAM TO DRAW LARGE VDU TITLES.
8180 BANN   DATA >2500        SUBROUTINE WP.
8190        DATA $+4
8200        DATA 0            MESSAGE IDENTIFIER STORE.
8210        MOV  @BANN+4,R9
8220        BLWP @CLER        CLEAR SCREEN.
8230        MOV  R9,R0
8240        AI   R0,4
8250        MOV  R0,@CHEK
8260        MOV  @DATK(R9),R8
8270 NXT    MOV  @DATB(R9),R3
8280        INCT R9
8290        MOV  @DATK(R9),R10
8300        DECT R9
8310        MOV  @DATJ(R9),R7
8320 NXT2   A    R7,R3
8330        MOV  @DATC(R8),R2
8340        MOV  @DATN(R8),R4
8350        MOV  @DATL(R8),R5
8360        BL   @CHS
8370        INCT R8
8380        C    R8,R10
8390        JLT  NXT2
8400        INCT R9
8410        C    R9,@CHEK
8420        JLT  NXT
8430        MOV  @DATB+8,R1
8440        LI   R2,>2000
8450        BL   @DRAW
8460        RTWP              RETURN TO MAIN PROG.
8470 CHS    CLR  R6
8480        MOV  R11,R12
8490 NXT1   MOV  R3,R1
8500        A    *R5+,R1
8510        BL   @DRAW
8520        INC  R6
8530        C    R6,R4
8540        JLT  NXT1
8550        B    *R12
8560 *
```

```
8570 * SUBROUTINE.
8580 DRAW   MOVB R1,@X
8590        MOV  R1,@Y
8600        MOVB @Z,@Y
8610        MOVB R2,@Y+2
8620        MOVB @Z+1,@Y+3
8630        LI   R0,DAT1
8640        BLWP @>E01C
8650        RT
8660 *
8670 *
8680 CHEK   DATA 0
8690 DAT1   DATA >1B58
8700 X      DATA >1B
8710 Y      BYTE 0,0,0,0,0
8720 Y1     BYTE >1B,>4B,0
8730 Z      BYTE >59,>08
8740 *
8750 **DAT2
8760        DATA >0705,>0604,>0503,>0402,>0403,>0404,>0305
8770        DATA >0205,>0104,>0103,>0102
8780 *
8790 **DAT3
8800        DATA >0701,>0601,>0501,>0401,>0301,>0201,>0101
8810        DATA >0202,>0303,>0403,>0204,>0105,>0205,>0305
8820        DATA >0405,>0505,>0605,>0705
8830 *
8840 **DAT4
8850        DATA >0605,>0704,>0703,>0702,>0601,>0501,>0401
8860        DATA >0301,>0201,>0102,>0103,>0104,>0205,>0305
8870        DATA >0405,>0505
8880 *
8890 **DAT5
8900        DATA >0101,>0105
8910 *
8920 **DAT6
8930        DATA >0104,>0103,>0102,>0203,>0303,>0403,>0503
8940        DATA >0603,>0703,>0704,>0702
8950 *
8960 **DAT7
8970        DATA >0705,>0604,>0503,>0402,>0303,>0204,>0105
8980 *
8990 **DAT8
9000        DATA >0701,>0601,>0501,>0401,>0301,>0201,>0101
9001        DATA >0102,>0103,>0104,>0105,>0402,>0403,>0702
9002        DATA >0703,>0704,>0705
9003 *
9004 **DAT9
9005        DATA >0703,>0603,>0503,>0403,>0304,>0204,>0105
9006        DATA >0302,>0202,>0101
9007 *
9008 **DAT0
9009        DATA >0402,>0403,>0404
9010 *
9011 **DATD
9012        DATA >0701,>0601,>0501,>0401,>0301,>0201,>0101
9013        DATA >0102,>0103,>0104,>0205,>0305,>0405,>0505
9014        DATA >0605,>0704,>0703,>0702
9015 *
9016 **DAT
9017        DATA >0701,>0601,>0501,>0401,>0301,>0201,>0102
9018        DATA >0103,>0104,>0205,>0305,>0405,>0505,>0605
9019        DATA >0705,>0402,>0403,>0404
9020 *
9021 **DATG
9022        DATA >0205,>0104,>0103,>0102,>0201,>0301,>0401
9023        DATA >0501,>0601,>0702,>0703,>0704,>0605,>0505
9024        DATA >0504
9025 *
9026 **DATS
9027        DATA >0101
9028 *
9029 DATK   DATA 0,10,20,30,42
9030 DATJ   DATA 14,14,14,12
9031 DATN   DATA 18,11,13,18,16
9032        DATA 18,16,11,16,18
9033        DATA 15,16,1,11,16
9034        DATA 14,17,10,15,18,18
9035 DATC   DATA >4D00,>4900,>4300,>5200,>4F00
9036        DATA >4D00,>4F00,>5400,>4F00,>5200
9037        DATA >4700,>4F00,>2000,>5400,>4F00
9038        DATA >4B00,>4500,>5900,>5000,>4100,>4400
9039 *
9040 **(DAT3,DAT6,DAT4,DAT2,DAT4)
9041 DATL   DATA $-374,$-302,$-338,$-396,$-338
9042 *
9043 **(DAT3,DAT4,DAT5,DAT4,DAT2)
?
9044        DATA $-384,$-348,$-316,$-348,$-406
9045 *
9046 **(DATG,DAT4,DATS,DAT5,DAT4)
9047        DATA $-154,$-358,$-124,$-326,$-358
9048 *
9049 **(DAT7,DAT8,DAT9,DAT0,DAT,DATD)
9050        DATA $-310,$-296,$-262,$-242,$-200,$-236
9051 *
9052 DATB   DATA >221A,>2C1A
9053        DATA >221A,>2C1A,>3621
9054        END  MAG
```

APPENDIX 4C CONTINUED.

REFERENCES IN VOLUME ONE

CHAPTER 1

1.1 Williamson, A.C.; Issa, N.A.H.; Makky, A.R.A.M.: 'Variable-Speed Inverter-Fed Synchronous Motor Employing Natural Commutation', Proc. IEE, Vol 125, No 2, Feb 1978, pp.113-120.

1.2 Russell, S.W.; Fyfe, R.L.: 'Brushless D.C. Motor With Variable Speed And Synchronous Control', in "Electrical Variable-Speed Drives", IEE Conference Publication 179, 25-27 Sept 1979, pp.78-83.

1.3 Persson, E.K.: 'Brushless D.C. Motors In High Performance Servo Systems', Proc. 4th Annual Symposium On Incremental Motion Control Systems And Devices, 1-3 April 1975, pp.T1-T15.

1.4 Persson, E.K.: 'Brushless Low-Inertia Motors', in "Small Electrical Machines", IEE Conference Publication 136, 1976, pp.19-22.

1.5 Ashen, R.A.; Bolton, H.R.: 'Performance And Design Of A Brushless D.C. Torque Motor', in "Small Electrical Machines", IEE Conference Publication 136, 1976, pp.27-30.

1.6 Feltbower, B.; Waldinger, H.: 'Machine-Commutated Inverter Drives Of Large Power', in "Electrical Variable-Speed Drives", IEE Conference Publication 179, 25-27 Sept 1979, pp.61-65.

1.7 Schauder, C.D.; Caddy, R.: 'Current Control Of Voltage Source Inverters For Fast Four-Quadrant Drive Performance', Conference Record, Industry Applications Society IEEE-IAS-1981 Annual Meeting, 5-9 Oct 1981, pp.592-599.

1.8 Conference Record, Industry Applications Society, IEEE-IAS-1981 Annual Meeting, 5-9 Oct 1981.

1.9 Electrical Variable-Speed Drives, IEE Conference Publication 179, 25-27 Sept 1979.

1.10 Proceedings Of International Conference On Electrical Machines, Athens, Sept 1980.

1.11 International Conference On Numerical Control Of Electrical Machines (CONUMEL), 1980, Lyon, France.

1.12 McLean, G.W.; Nix, G.F.; Alwash, S.R.: 'Performance And Design Of Induction Motors With Square-Wave Excitation', Proc. IEE, Vol 116, No 8, Aug 1969, pp.1405-1411.

1.13 Enslin, N.C.; Malengret, M. (Cape Town University, South Africa): 'Flux And Current Distributions In The 9 Phase Inverter Driven Induction Motor', Private communication to Imperial College, Dept of Electrical Engineering, July 1978.

CHAPTER 2

2.1 Say, M.G.: 'Alternating Current Machines', (Pitman Publishing, 1978), Chapter 10.

2.2 Matsch, L.W.: 'Electromagnetic And Electromechanical Machines', (I.E.P., 1977).

2.3 Ellison, A.J.: 'Electromechanical Energy Conversion', (Harrap, 1965).

2.4 Adkins, B.: 'The General Theory Of Electrical Machines', (Chapman and Hall, 1962).

2.5 Ashen, R.A.: 'Some Aspects Of High-Torque, Low Speed, Brushless Electric Motors', Ph.D Thesis, Imperial College, London University, 1977.

2.6 Woodbury, J.R.: 'Efficient Control Of A-C Machines In The Brushless D-C Configuration', IEEE Transactions On Industrial Electronics And Control Instrumentation, Vol IECI-24, No 1, Feb 1977, pp.13-23.

2.7 Seshanna, P.; Ilango, B.: 'Characteristics Of A Variable Speed Synchronous Motor', Electric Machines

And Electromechanics, Vol 3, No 2, 1979, pp.143-155.

2.8 Slemon, G.R.; Forsythe, J.B.; Dewan, S.B.: 'Controlled-Power-Angle Synchronous Motor Inverter Drive System', IEEE Transactions on Industry Applications, Vol IA-9, No 2, Mar/Apr 1973, pp.216-219.

2.9 Chalmers, B.J.; Pacey, K.; Gibson, J.P.: 'Brushless D.C. Traction Drive', Proc. IEE, Vol 122, No 7, July 1975, pp.733-738.

2.10 Shepherd, J.: 'Developments Of Inverter-Synchronous Motor Systems', Proc. International Conference On Electrical Machines, Athens, 1980, pp.1036-1042.

2.11 Woodbury, J.R.: 'The Design Of Brushless DC Motor Systems', IEEE Transactions on Industrial Electronics and Control Instrumentation, Vol IECI-21, No 2, May 1974, pp52-60.

2.12 Gumaste, A.V.; Slemon, G.R.: 'Steady-State Analysis Of A Permanent Magnet Synchronous Motor Drive With Voltage-Source Inverter', IEEE Transactions on Industry Applications, Vol IA-17, No 2, Mar/Apr 1981, pp.143-151.

2.13 Say, M.G.: 'Alternating Current Machines', (Pitman Publishing, 1978), pp.372 & 387.

CHAPTER 3

3.1 Fink, R.A.: 'The Brushless DC Motor: Link To System Versatility', Control Engineering, June 1970, pp.75-78.

3.2 Fink, R.A.: 'The Brushless Motor: Types And Sources', Control Engineering, August 1970, pp.42-45.

3.3 Yates, W.W.: 'Brushless D-C Motors', Machine Design, March 3, 1966, pp.136-142.

3.4 Woodbury, J.R.: 'The Design Of Brushless DC Motor Systems', IEEE Transactions on Industrial Electronics and Control Instrumentation, Vol IECI-21, No 2, May 1974, pp.52-60.

3.5 Woodbury, J.R.: 'Efficient Control Of A-C Machines In The Brushless D-C Configuration', IEEE Transactions on Industrial Electronics and Control Instrumentation, Vol IECI-24, No 1, Feb 1977, pp.13-23.

3.6 Binns, K.J.; Sneyers, B.; Maggetto, G.; Lataire, Ph.: 'Rotor-Position-Controlled Permanent Magnet Synchronous Machines For Electrical Vehicles', Proc. International Conference on Electrical Machines, Athens, 1980, pp.346-357.

3.7 Rieke, B.: 'The Microprocessor Control Of A Four Phase Star-Connected Multi-Pole Reluctance Motor', Proc. of International Conference on Electrical Machines, Athens, 1980, pp.394-401.

3.8 Le-Huy, H.; Jakubowicz, A.; Pacaut, R.: 'Starting Scheme For A Microprocessor-Based Converter-Fed Self-Control Synchronous Motor Drive', Proc. of International Conference on Electrical Machines, Athens, 1980, pp.946-953.

3.9 Taub, H.; Schilling, D.: 'Digital Integrated Electronics', (McGraw-Hill Kogakusha, 1977), pp.106-107.

3.10 Peter, J.M.: 'Limits Of Safe Operation Of Power Transistors In The Switching Mode', (Sescosem Publication, 1975), pp.4-31.

3.11 McMurray, W.: 'Optimum Snubbers For Power Semiconductors', in "Power Semiconductor Applications, Vol 1 1972", (IEEE Press, Editors: Harnden, J.D., & Golden, F.C.), pp.33-41.

3.12 McMurray, W.: 'Selection Of Snubbers And Clamps To Optimise The Design Of Transistor Switching Converters', IEEE Transactions on Industry Applications, Vol IA-16, No 4, July/Aug 1980, pp.513-523.

3.13 Mallinson, N.M.: 'Power Switch Protection Using Snubbers And Freewheel Diodes', Internal Report, Department of Electrical Engineering, Imperial College, London, May 1979.

3.14 Slemon, G.R.; Forsythe, J.B.; Dewan, S.B.: 'Controlled-Power-Angle Synchronous Motor Inverter Drive System', IEEE Transactions on Industry Applications, Vol IA-9, No 2, Mar/Apr 1973, pp.216-219.

CHAPTER 4

4.1 Schnieder, E.: 'Control Of DC-Drives By Microprocessors', IFAC Symposium on Control in Power Electronics and Electrical Drives, 2nd Dusseldorf, 1977, (Editor: Leonhard, W.), pp.603-608.

4.2 Sen, P.C.; Trezise, J.C.; Sack, M.: 'Microprocessor Control Of An Induction Motor With Flux Regulation', IEEE Transactions on Industrial Electronics and Control Instrumentation, Vol IECI-28, No 1, Feb 1981, pp.17-21.

4.3 Harashima, F.; Naitoh, H.; Koyama, M.; Kondo, S.: 'Performance Improvement In Microprocessor-Based Digital PLL Speed Control System', IEEE Transactions on Industrial Electronics and Control Instrumentation, Vol IECI-28, No 1, Feb 1981, pp.56-61.

4.4 Rajashekara, K.S.; Vithayathil, J.: 'Microprocessor Based Sinusoidal PWM Inverter By DMA Transfer', IEEE Transactions on Industrial Electronics, Vol IE-29, No 1, Feb 1982, pp.46-51.

4.5 Le-Huy, H.; Jakubowicz, A.; Pacaut, R.: 'Starting Scheme For A Microprocessor-Based Converter-Fed Self-Control Synchronous Motor Drive', Proc. of International Conference on Electrical Machines, Athens, 1980, pp.946-953.

4.6 Rieke, B.: 'The Microprocessor Control Of A Four Phase Star-Connected Multi-Pole Reluctance Motor', Proc. of International Conference on Electrical Machines, Athens, 1980, pp.394-401.

4.7 Fornel, B. De; Hapiot, J.C.; Reboulet, C.; Boidin, M.: 'Numerical Speed Control Of A Current-Fed Asynchronous Machine', Proc. of International

Conference on Electrical Machines, Athens, 1980, pp.793-800.

4.8 TMS9900 Microprocessor Data Manual (Dec 1976), The Engineering Staff of Texas Instruments Incorporated Semiconductor Group.

4.9 TMS9901 Programmable Systems Interface (July 1978), The Engineering Staff of Texas Instruments Incorporated Semiconductor Group.

4.10 TMS9902 Asynchronous Communications Controller Data Manual (July 1978), The Engineering Staff of Texas Instruments Incorporated Semiconductor Group.

4.11 9900 Family Systems Design And Data Book, 1st Ed., Texas Instruments Learning Center.

4.12 TMS9900 Family System Development Manual, a Texas Instruments Application Report, 1977, Bulletin MP702.

4.13 TM990/100M Microcomputer User's Guide (October 1979), The Engineering Staff of Texas Instruments Incorporated Semiconductor Group.

4.14 TM990/401-1 TIBUG Monitor Listing (June 1977), The Engineering Staff of Texas Instruments Incorporated Semiconductor Group.

4.15 TM990/302 Software Development Board User's Guide (May 1979), The Engineering Staff of Texas Instruments Incorporated Semiconductor Group.

4.16 TM990/302 Hardware User's Guide (January 1979), The Engineering Staff of Texas Instruments Incorporated Semiconductor Group.

4.17 TM990/201 And TM990/206 Expansion Memory Boards (Jan 1979), The Engineering Staff of Texas Instruments Incorporated Semiconductor Group.

4.18 TM990/512 TM990/513 Universal Prototyping Boards (February 1979), The Engineering Staff of Texas Instruments Incorporated Semiconductor Group.

4.19 TM990/510 TM990/520 Card Cages (February 1979), The Engineering Staff of Texas Instruments Incorporated Semiconductor Group.

4.20 Model-550B Users Manual (Sept 16, 1980), Perkin Elmer Terminals Division.

4.21 The TTL Data Book For Design Engineers (2nd Ed.), The Engineering Staff of Texas Instruments Components Group.

MICROPROCESSOR-CONTROLLED INVERTER-FED SYNCHRONOUS MOTOR

by
Nicholas Martin Mallinson, B.Sc.(Eng), A.C.G.I.

Volume Two of Two Volumes

October 1983

A thesis submitted for the degree of
Doctor of Philosophy of the University of London
and for the Diploma of Membership of the
Imperial College of Science and Technology

Department of Electrical Engineering
Imperial College of Science and Technology
London SW7 2AZ

ABRIDGED LIST OF CONTENTS IN VOLUME TWO

(For full details see Volume One)

CHAPTER 5

THE SEVEN PHASE MOSFET INVERTER

5.1 Introduction

The basic philosophy of the work covered by this thesis is explained in Chapter 1, where it is stated that a square wave voltage source inverter was chosen for the drive system, because such an inverter is simple when compared to some other forms of inverter. This is rather a bold statement to make. Therefore, in an attempt to justify the choice, the first part of this chapter reviews the commonest forms of inverter and the modes in which they can operate. In addition, the choice of MOSFET power transistors as the switching elements is justified by briefly comparing their main characteristics with those of the more commonly used thyristors and bipolar transistors.

The design, construction, and initial testing of a 7 phase square wave voltage source inverter, with a rating of 3 kW is then described. Seven phases are required to match the 7 phase motor designed in Chapter 6. The inverter is configured as a peripheral of the microprocessor system described in Chapter 4.

5.2 Review of Possible Inverter Types

Variable speed operation with an a.c. motor can be achieved by supplying the motor from a variable frequency power source. The oldest and arguably simplest variable

frequency source is an alternator driven by a variable speed prime mover, but this solution can be expensive, bulky and requires regular maintenance. Since the invention of semiconductor devices, much work has been done to produce electronic variable frequency supply units, commonly known as inverters.

Inverters can be classified in three ways:

(a) by the shape of waveform produced;

(b) by the manner in which the waveform is generated;

(c) by the electrical quantity that forms the controlled waveform.

These classifications are discussed briefly below.

5.2.1 Waveform Shape

There are essentially three forms of waveform commonly produced by inverters.

5.2.1.1 Sinusoidal Waveform Inverter

Electronic feedback is required within an inverter to ensure that the output is sinusoidal. Sinusoidal inverters are well suited to driving standard sinusoidal motors, as the low harmonic content of the waveform reduces the losses within the motor, but the production of such a waveform can involve increased losses within the inverter.

5.2.1.2 Quasi-Square Waveform Inverter

This type of inverter produces an output that has a square or rectangular form. The output is either on or off, and so in general, no feedback is needed within the inverter to achieve the desired waveform. The circuitry

required to generate a quasi-square output is usually simpler than that required for sinusoidal outputs. However, a square waveform causes the losses in a sinusoidal motor to increase. Consequently the motor rating can be reduced.

5.2.1.3 Complex Waveform Inverter

This type of inverter produces an output that is a replica of a reference signal. The waveform can be of any form within the frequency response of the inverter and so sinusoidal or square wave outputs are possible.

5.2.2 Waveform Generation

Irrespective of the desired output waveform, there are two popular techniques that can be employed to implement the inverter action. An inverter can be designed to operate either in a linear mode or in a switching mode. In exceptional cases a combination of modes is used. The relative merits of the two modes are summarised below.

5.2.2.1 Linear Mode Inverter

This form of inverter is essentially a large power amplifier, the output being an amplified version of a control signal. It can therefore produce any waveform within its bandwidth limitations. The output stage can operate in the Class A or Class B mode, the peak efficiencies of which are 50% and 78.5% respectively. Therefore a significant portion of the power supplied to a linear inverter is dissipated within the inverter in the form of heat. A linear mode inverter provides an output whose amplitude and frequency is easily controlled. A sinewave output with low harmonic content can be easily obtained if a suitable reference signal is available. Such an inverter is therefore suitable for driving sinusoidal motors. Power amplifiers are available commercially with outputs up to several kilowatts but they are expensive, (several would be needed for a polyphase motor), and relatively inefficient.

Hence they cannot be considered for the majority of drive systems.

5.2.2.2 Switching Mode Inverter

In this form of inverter the semiconductor components are used as switches. This is the most efficient way of using semiconductor components, as the losses in a perfect switch are zero. Therefore, an inverter efficiency of 100% is theoretically possible. In practice, semiconductors do dissipate some power when conducting, due to a finite voltage drop across them. In addition the finite time taken for a semiconductor to switch, means that there are times when the device is carrying current whilst there is a significant voltage across it. The dissipation during switching can be minimised by ensuring that the device switches as fast as possible, but the resulting rates of change of voltage and current can damage the device. Protection circuits, known as snubbers, can be added to prevent this but some power is dissipated by the snubbers and the power loss is proportional to switching frequency. Hence the efficiency of a switching mode inverter is, in practice, less than 100%.

A switching inverter is ideally suited to produce quasi-square waveforms at the fundamental frequency required by the load motor. When used in this manner, the square wave amplitude can be controlled by adjusting the d.c. input to the inverter. For example, this can be achieved by means of a phase controlled rectifier or by a d.c. chopper. Alternatively, the amplitude of the square wave output can be controlled directly by the inverter, by using modulation techniques where the inverter switches at a frequency much higher than the desired output frequency. Two applicable techniques are pulse frequency modulation (PFM) and pulse width modulation (PWM). In PFM the inverter produces pulses of fixed width and the frequency of the pulses is controlled to give the desired average output amplitude. In PWM the inverter produces pulses at a

fixed frequency and the width of the pulses is controlled. Both PFM and PWM rely on the motor winding inductance to filter out the high frequency switching, so that the winding effectively only "sees" the average amplitude. PWM is a popular technique because the control signals necessary for the inverter can be produced simply by comparing the required reference signal with a high frequency triangular wave, as described by Issawi (5.1). A PWM switching inverter can produce sinusoidal outputs and the harmonic content of the waveform can be minimised by the use of a high switching frequency. However, the maximum switching frequency is limited by the semiconductor device used and the magnitude of the switching losses (including snubber losses) that can be tolerated. When allowance is made for switching losses, switching mode inverters typically have efficiencies in excess of 90%.

5.2.3 Inverter Output Controlled Quantity

An inverter can be arranged to control any electrical variable such as voltage, current, or power, and the controlled variable directly affects the motor performance characteristics. For example, when an induction motor is fed from a constant power source it has a characteristic similar to a d.c. series motor (5.2). However, voltage or current are the variables normally controlled by inverters and they are briefly discussed below.

5.2.3.1 Voltage Source Inverter

In a voltage source inverter the control signals to the inverter and/or its d.c. supply are arranged so that the voltage output waveform is controlled in both frequency and amplitude. The resulting load current is determined by the load impedance. A synchronous motor driven by such an inverter can exhibit series or shunt d.c. motor characteristics as shown by Chalmers et al (5.3).

A major point in favour of voltage source inverters

is the ease with which the output voltage can be sensed and compared with the reference signal. The main disadvantage of a voltage forced inverter is that it is not inherently short circuit proof and so some current overload protection is necessary, the simplest form being fast acting semiconductor protection fuses.

5.2.3.2 Current Source Inverter

A current source inverter controls the frequency and amplitude of the current output from the inverter. The voltage across the load therefore depends on the load impedance. Current source inverters are widely used in large industrial motor drives. A typical current source inverter system is described by Slemon et al (5.4), in which a controlled rectifier and inductor provide a controlled constant current to a quasi-square wave thyristor inverter. An alternative method of achieving a current source output is described by Hurley (5.5). The method does not require an inductor in the d.c. supply. The inverter switches either the full positive or negative supply voltage across the load. Whenever the load current goes above or below desired thresholds, the inverter changes the polarity of the output volts to force the current back within the limits. The action is analagous to a thermostat in a water heater.

If a current source inverter is used to drive a synchronous motor with a position sensor system, the resulting torque is proportional to supply current and remains constant as the speed varies. Such a system is very useful in servo type applications.

A useful performance feature of a current source inverter is the fact that it is short circuit proof. However, this is offset to some extent by the need to incorporate current sensors to enable the control circuit to hold the output current at the desired value. A current sensor is more difficult to implement than a voltage sensor, partly

because the current sensor has to be located in series with the load.

5.2.4 Published Literature on Various Aspects of Inverters

The literature dealing with inverters and with inverter/motor drive systems is now very extensive. In addition to a number of significant papers in the journals of the learned societies, there have been a number of highly relevant conference proceedings and books published in recent years. A reasonably comprehensive view of the state of the art can be gained from references (5.6), (5.7), and (5.8), which each contain a number of papers on inverter-fed synchronous motor drives. Where special features of the presented work in this thesis overlap with those in existing published drive schemes, (at least in those of which the author is aware), specific references are cited.

5.3 The Chosen Inverter Type

Having studied the various inverter options described in section 5.2, it was decided that the simplest type was a switching inverter producing a voltage-controlled, quasi-square waveform at the frequency required by the load motor. The chain of decisions involved in the choice can be summarised as follows.

A linear inverter was rejected immediately on the grounds of high device cost and low efficiency. Therefore a switching inverter was chosen. A switching inverter is most suited to producing square waves since that is the basic action of a switch. Furthermore, if the switching frequency is kept to the minimum possible, the losses in the inverter are minimised and the driver circuits for the semiconductor switches can be simpler because of the low frequency operation. Therefore, a sinusoidal output was rejected on the grounds that it is not a "natural" output from a switching inverter. The techniques (such as PWM)

needed to synthesize sinewaves, require extra complexity in the signal portions of the inverter and higher output stage switching frequencies. Finally, it was decided that a voltage source inverter was preferable to a current source inverter in order to avoid the need for current sensors. These decisions, which of course led to a convenient inverter design, were made in the knowledge that some of the disadvantages of the square wave voltage source inverter could very likely be attenuated by the incorporation of certain features in the motor design.

The choice of a quasi-square wave voltage source switching inverter as the supply for an autopiloted synchronous motor, has two important consequences for the system as a whole. One of the primary aims of the project was to match the inverter and motor waveforms as closely as possible. (The reasons and benefits of this are discussed in Chapter 1). The matching of voltage waveforms in autopiloted synchronous motor systems can lead to a reduction in current peaks with a corresponding drop in heating losses and a maximisation of inverter output per device rated voltage x rated current.

Therefore, the first consequence arising from the choice of a quasi-square wave voltage source inverter was the need for a motor capable of generating a quasi-square back-emf phase waveform, in order to achieve the aim of a motor/feed match. No standard brushless motors with quasi-square back-emf voltages are commercially available. Hence in order to achieve the desired match between the inverter and motor, a special square wave synchronous motor was designed and is described in Chapter 6. The pros and cons of a square wave synchronous motor configuration are also discussed in Chapter 6. The only other known work on a "square wave" synchronous motor was recently reported by Frankl (5.9), (5.10) and (5.11).

The second consequence arising from the choice of a quasi-square wave voltage source switching inverter was

the need for a means to control the magnitude of the voltage waveforms. Since the inverter cannot control the magnitude of the voltage waveforms, it is necessary to supply the inverter from a variable voltage source which must be able to handle the total inverter power requirements. (Note that the losses in such a supply may be comparable with, or even greater than, the losses incurred in a PWM inverter, but since no motor can produce a PWM back-emf, it is not possible to use a PWM inverter and still achieve a motor/inverter waveform match.)

The initial calculations on the square wave motor led to a requirement for a 7 phase square wave inverter with a total rating of about 3 kW, and a capability of switching at frequencies up to 500 Hz. The final design of the inverter described in this chapter, is in fact capable of higher frequencies and so it could be used for PWM work if desired. A block diagram of the inverter system is shown in fig. 5.1.

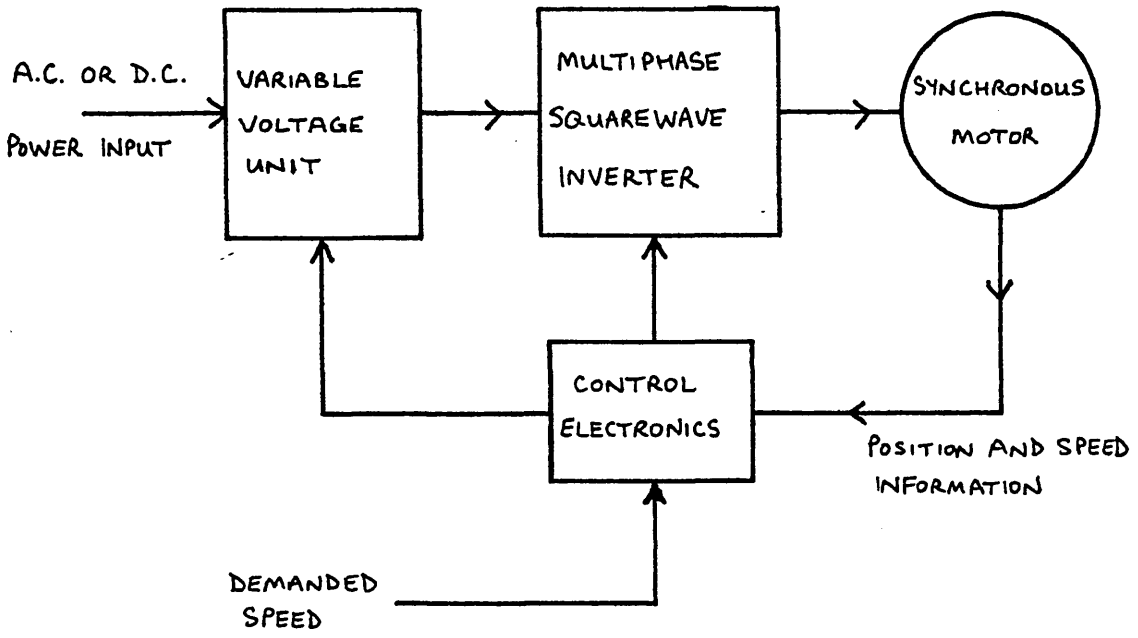


FIG. 5.1. BLOCK DIAGRAM OF THE SQUARE WAVE INVERTER SYSTEM

5.4 The Choice of Semiconductor Switch for Use in the Inverter

Three semiconductor switches were considered for use in the inverter. They were:

(a) the thyristor;

(b) the bipolar power transistor;

and (c) the power MOSFET transistor.

All the devices are suitable for use in inverter applications, with the required power level and switching frequency determining which is the most appropriate device to use. The devices obviously must be rated for the voltage and current levels that they have to switch. They may also require what is known as snubber protection, to prevent secondary breakdown effects during switching.

Secondary breakdown is a condition where imperfections in the device structure cause uneven current distribution during switching, resulting in hot spots due to increased current density in those regions. The devices can fail due to secondary breakdown. Snubber networks limit the rate of change of current and voltage in the device and so prevent secondary breakdown. An excessive rate of change of voltage is especially troublesome, since it can cause the devices to avalanche into conduction at a time when they should be off, and the resulting uneven current distribution leads to secondary breakdown.

It is not felt necessary to examine snubbers in any detail in this chapter, as design techniques are discussed in several papers including those by McMurray (5.12) (5.13), Rice (5.14) and Balthasar (5.15). Appendix 3A contains the design details for the snubbers used in the Magslip system transistor inverter described in Chapter 3.

The manner in which thyristors and bipolar transistors work can be found in any good electronics textbook. However, MOSFETs are relatively new devices and their ratings are constantly being improved. The acronym MOSFET stands for Metal-Oxide Semiconductor Field-Effect-Transistor. Some information is available in textbooks, but technical magazine articles (references 5.16 to 5.25 inclusive) and manufacturers application notes (5.26),(5.27),(5.28),(5.29) are more helpful at present.

The main characteristics of each device have been summarised in the following subsections, so that the type of device chosen for the inverter can be justified.

5.4.1 Thyristor Characteristics

Salient features of a thyristor include:

(a) The turn on time of a thyristor is relatively slow, of the order of 2μs to 10μs, and the turn off time is very slow, being about 10μs to 20μs for a high speed thyristor.

(b) A certain minimum time is required when commutating a thyristor, during which time the blocking action re-establishes itself.

(c) Conditions (a) and (b) place a maximum practical switching frequency limit of between 1 kHz to 10 kHz on thyristor circuits.

(d) The gate current required to switch large anode currents is relatively small, typically 100 mA for a 30 A thyristor.

(e) The device may require an inductive snubber in series with it to limit the rate of rise of anode current (dI_a/dt) at switch on. "Hot spots" within the thyristor structure can occur if the current is allowed to rise too

fast before the thyristor is fully conducting.

(f) The device can be triggered on by large rates of change of voltage across the anode and cathode terminals, (dV_{ac}/dt). Therefore, snubber circuits are needed to limit the maximum dV_{ac}/dt.

(g) The device has a negative temperature coefficient of resistance which can lead to thermal runaway if it gets hot.

(h) Thyristors can be connected in series or parallel to increase the total rating of the switch, but precautions must be taken to ensure that the total voltage or current is shared equally (as far as possible) amongst the devices.

(i) Gate turn off (GTO) thyristors have recently been developed. They do not need commutation circuitry, but their use is not yet widespread.

5.4.2 Bipolar Transistor Characteristics

The important features of power bipolar transistors can be summarised as follows.

(a) The structure of pnp transistors makes them less suitable for high power applications than npn transistors. Therefore npn transistors often have to be used in parts of circuits where ideally pnp transistors would be used. This can result in the need for complicated driver circuits for the npn transistors.

(b) The collector current is a function of the base current. The ratio of these two currents (the current gain β) is in the order of 5 to 100 for power transistors. Current gains of up to 250 can be achieved using Darlington power transistors but at the expense of slower switching speeds.

(c) When the base current is sufficiently large to take it into saturation, the transistor is effectively a closed switch with a collector to emitter voltage drop of between 0.1 volts to 2.5 volts, depending on the transistor design.

(d) An excess stored charge builds up inside the transistor if the base current is larger than the value required to just hold the transistor in saturation. The stored charge has to be removed before the collector current can fall to zero, and the design of the base driver circuits must allow for this.

(e) Current conduction in a bipolar transistor is essentially by means of the minority carriers. The recombination times associated with the minority carriers limit the switching times of bipolar transistors. The turn-on and turn-off times of a power bipolar transistor are typically 1μs when the base is correctly driven, although much faster devices with switching times of the order of 300 ns are now becoming available.

(f) The transistor can be turned off at will by removing the base current.

(g) The need for a continuous base current to maintain the transistor in an on state, results in a significant base drive power requirement.

(h) The rate of change of collector current (dI_c/dt) may need to be limited at switch on, (as for thyristors), in order to prevent hot spots and consequent damage.

(i) To prevent avalanche break down of the collector-base junction, and also limit power dissipation during switch-off, it can be necessary to use snubbers to limit the rate of change of voltage (dV_{ce}/dt) across the collector emitter terminals.

(j) A bipolar transistor has a negative temperature coefficient of resistance which results in thermal runaway when the device gets hot, and this makes the use of transistors in parallel or series combinations difficult.

(k) Bipolar transistors can handle voltages of up to 1000 volts and currents of up to 100 amps, but in general their power handling capability is less than thyristors.

5.4.3 The MOSFET Transistor

The circuit symbols for p type and n type MOSFETS are shown in fig. 5.2.

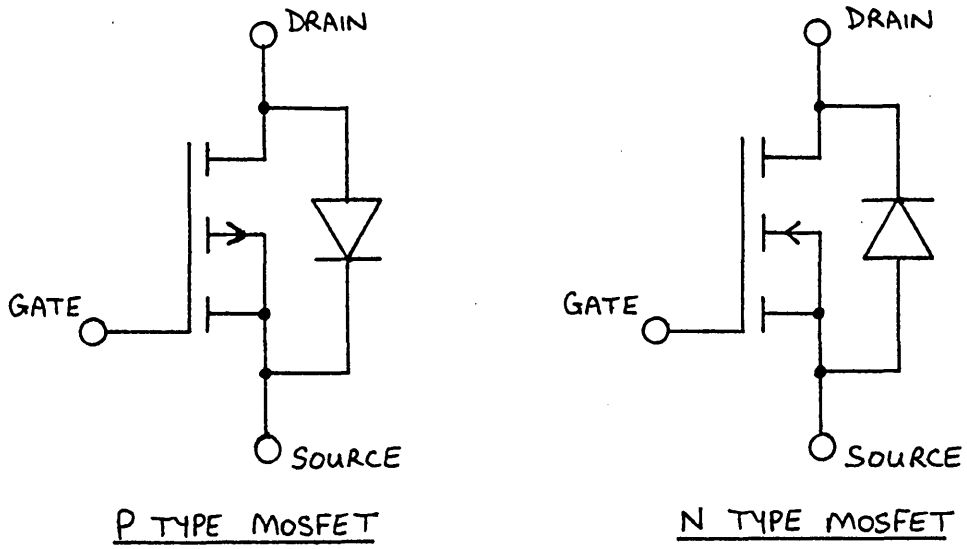


FIG. 5.2. CIRCUIT SYMBOLS FOR P TYPE AND N TYPE MOSFETS

The power MOSFET can be summarised as follows.

(a) The control terminal is called the gate.

(b) When zero voltage is applied to the gate with respect to the source, the MOSFET presents a very high resistance between the drain and the source.

(c) If the gate to source voltage is raised above a fixed threshold voltage $V_{gs(th)}$, the MOSFET turns on and the

drain and source are connected by a low value resistance $R_{ds(on)}$, typically 0.01Ω to 2.5Ω, depending on the voltage rating of the MOSFET.

(d) The gate current I_g drawn when the device is on is typically nanoamperes due to its high input impedance. During turn-on or turn-off, the gate current is larger as it is essentially the charging current for the gate capacitance.

(e) The ratio of drain current to gate current is extremely large, (theoretically infinite), and the device can be considered as a voltage controlled switch.

(f) The current conduction is by means of majority carriers. The switching times are therefore very fast, (10 to 100 ns), and are essentially only limited by the time taken to charge and discharge the gate capacitance. The times are essentially independent of temperature.

(g) The device has a positive temperature coefficient of resistance and so it protects itself against thermal runaway. This also allows the devices to be parallelled much more easily than bipolar transistors.

(h) The structure of a MOSFET is such that a parasitic transistor exists which is effectively a diode connected in reverse across the drain and source. This diode can have a switching speed approaching that of the MOSFET and it has an equal current rating. Therefore it can be used as a freewheel diode in some circuit configurations.

(i) Commercially available MOSFETs can switch voltages of up to 1000 volts and carry currents up to about 30 amps at present. (The higher current ratings are available at the lower voltage ratings; e.g. 60 volt devices are rated up to 30 amps whereas 1000 volt devices are only rated up to about 4.7 amps. The higher voltage rating devices are very expensive and have a high on resistance, (e.g. 2Ω for

an 800v device). In high voltage applications (e.g. 800 volts), where the power dissipation due to the high on resistance cannot be tolerated, a cascode connection of a MOSFET and bipolar transistor can be employed (5.23). The cascode combination couples the desirable performance features of MOSFETs to the high voltage capability of bipolar technology. A very cost effective high voltage switch is obtained which has a lower power dissipation than a MOSFET of equal voltage rating.

(j) In certain applications some snubber protection may be necessary to limit the rate of change of voltage across the drain and source terminals (dV_{ds}/dt), and so prevent turn on of the parasitic transistor. However, manufacturers have recently stated (5.30) that MOSFETs can easily withstand a dV_{ds}/dt of 6000v/μs and sample MOSFETs have withstood 50000v/μs. Therefore, in most practical circuits no snubbering is needed because rates of change of voltage are rarely of these magnitudes and suitable gate control can reduce the rates of change if necessary. If the parasitic transistor is used as a freewheel diode it should be allowed 5 to 10 ns of reverse recovery time before a large dV_{ds}/dt is placed across it, but this delay can easily be achieved by suitable gate control or by a small amount of snubbering across the device. Manufacturers of MOSFETs are still trying to fully understand the mechanisms of the parasitic transistor, but they do state quite firmly that MOSFETs are robust devices which are conservatively rated in the data sheets. They say that no trouble should occur, providing the rated drain to source voltage is not exceeded.

(k) The faster switching times and the ability to operate without snubbers, result in a reduction in switching losses compared to bipolar transistors.

(l) Until recently, n type MOSFETs have been much more readily available at high power ratings than p types. This is because the majority carriers in p type MOSFETs

are holes and these have a lower mobility than electrons, resulting in an on resistance roughly twice that of n type MOSFETs. Therefore, gate drive circuits have had to be developed to enable the use of n type MOSFETs in positions where a p type would be more appropriate.

(m) It has been found that during switching, some MOSFETs have a tendency to oscillate at frequencies so high that they are not visible on 100 MHz oscilloscopes. The dissipation resulting from the oscillations can destroy the device. The oscillations can be prevented by placing a resistor of about 100Ω in the gate connection or by placing a ferrite bead on the gate lead.

5.4.4 The Chosen Inverter Switching Device

Having compared the characteristics of the thyristor, bipolar transistor and MOSFET, it was apparent that any of them could be used to build a 7 phase 3 kW inverter capable of producing a 500 Hz quasi-square wave output. However, the thyristor was rejected because it requires commutation circuits to turn it off. The bipolar transistor was then rejected because it requires a significant base current to maintain it in conduction and it needs significant snubber protection. The MOSFET appeared to be the ideal choice as it only requires very low power control signals, it switches very fast, needs little (if any) protection and has a self contained freewheel diode (5.21). In addition the seven phase design of the inverter meant that the power per phase was at a level that could be dealt with by MOSFETs. Output device cost was higher but power MOSFET costs are dropping fast, and the drive circuit costs are of course considerably lower. A MOSFET was therefore chosen as the switching device.

At the time of the inverter design, the only MOSFETs readily available in Britain with reasonable power handling capabilities, were the International Rectifier n type "Hexfet" devices. Today MOSFETs are available from many

other manufacturers including Intersil, Siliconix, Motorola, Siemens, Hitachi and Ferranti.

Given that a 3 kW power level was required from the inverter, the selection of the particular type of hexfet to be used was made on the basis that it was important to minimise the current requirements of both the inverter and motor. This is because a variable voltage source supplies the inverter, and the cost and availability of such supplies depends heavily on the required output current capability. The power input of the inverter is the product of the d.c. rail voltage and current, and so by using a high d.c. rail voltage the current is minimised. Therefore, it was desirable to choose the highest voltage hexfet that could be afforded.

The required switch voltage rating is also dependent on the configuration of the switches in an inverter. There are basically five ways in which a phase winding of a motor can be driven and they are shown in figures: 5.3(a),(b),(c),(d) and (e). Freewheel diodes are shown where they can be employed. The configurations are compared below on the basis that the same motor phase winding is used in each circuit. (i.e. the winding impedance and phase current magnitude are the same in all cases.) Hence the voltage rail magnitudes are defined and so a comparison between the required switch voltage ratings in the various configurations can be made.

The single "unipolar" drive shown in fig. 5.3(a) only requires one switch per phase and the switch is conveniently connected to the ground rail, making the interfacing to the driver circuit easy. However, the unipolar connection can only drive current through the phase winding in one direction and so the machine power output is significantly reduced. The switch must be rated for the supply rail voltage V.

The double "unipolar" drive shown in fig. 5.3(b) uses

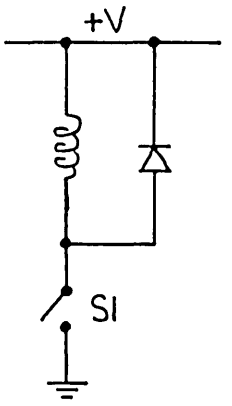


FIG. 5.3a SINGLE "UNIPOLAR" DRIVE PER PHASE

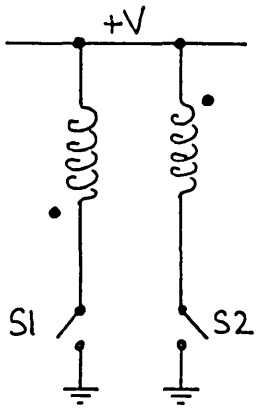


FIG. 5.3b DOUBLE "UNIPOLAR" DRIVE PER PHASE

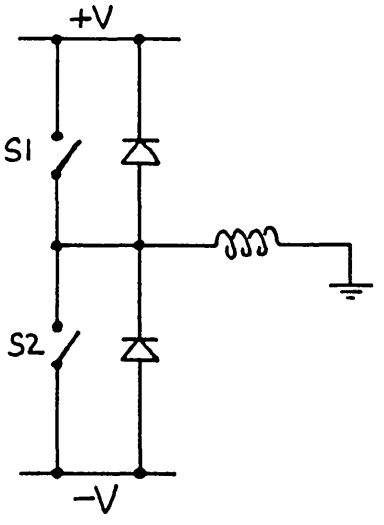


FIG. 5.3c HALF BRIDGE PER PHASE

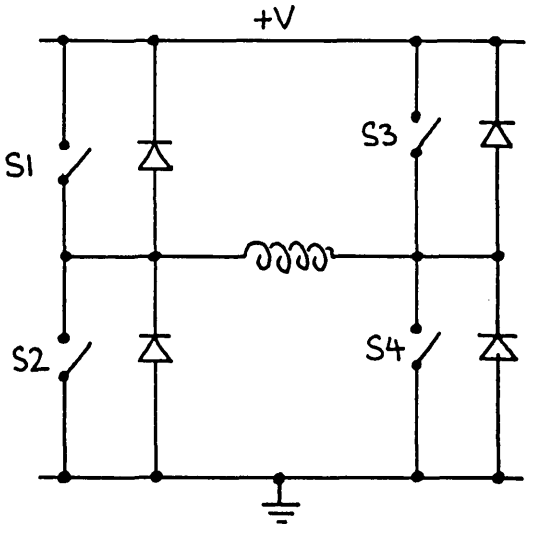


FIG. 5.3d FULL BRIDGE PER PHASE

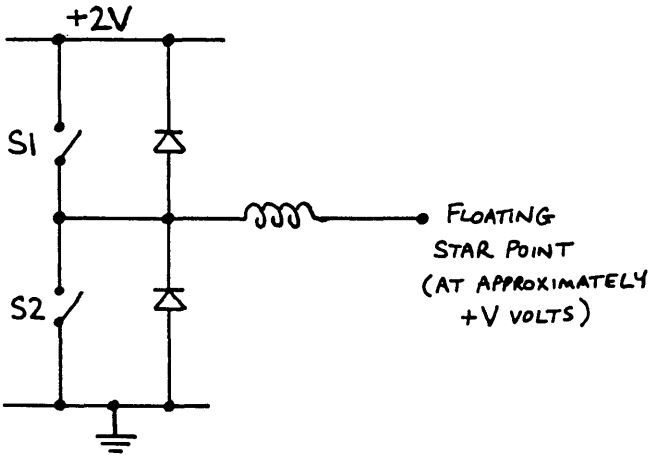


FIG. 5.3e SINGLE-RAIL HALF BRIDGE PER PHASE

FIG. 5.3. THE ALTERNATIVE WAYS OF DRIVING A PHASE WINDING

two switches. Each switch drives half of a bifilar wound phase winding, and the polarity of the connections means that the phase current can be reversed. However, the transformer action between the two coils is a problem, since when one switch is closed, the other switch experiences a voltage twice the size of the supply voltage V. The switches must therefore be rated appropriately. In addition, the transformer action prevents the use of freewheel diodes across the coils.

The dual voltage rail half-bridge circuit shown in fig. 5.3(c) requires only two switches per phase winding and bidirectional phase current is possible. The main disadvantage of this circuit is the need for a positive and negative voltage supply, and the fact that the transistors must be rated for the total voltage between the two rails. (i.e. 2V.)

The Full Bridge circuit shown in fig. 5.3(d) uses four switches, but can provide a bidirectional phase current from a single rail supply. Therefore, the switches only have to be rated for the rail voltage V. The main disadvantage of this circuit is its need for four switches and four driver circuits.

Finally, the half-bridge single rail circuit is shown in fig. 5.3(e). This circuit can provide a bidirectional phase current from a single rail supply if the phase windings of the motor are interconnected by a floating star point. (This type of motor is very common in inverter fed systems.) The circuit relies on the star point being at half the supply rail potential. The circuit only requires two switches per phase but they need a voltage rating of 2V. (Alternatively, the circuit can use a V volt supply rail and the motor winding can be modified to draw double the phase current taken in the other circuits. In this way the volt-ampere input is maintained but the voltage rating of the switches is only V volts.)

On balance it was decided that in spite of their simpler circuitry, the unipolar drive options were less attractive than the other options, due mainly to the accompanying reduced motor performance. The choice hence lay between the half bridge and full bridge circuits. The half bridge single voltage rail circuit was rejected because the floating star point connection between the phase windings, prevents the applied phase voltages from being accurately defined. (i.e. The matching of the applied phase voltage and the back-emf voltage cannot be achieved.) Finally, on the grounds of cost and circuit complexity, it was felt that a dual voltage rail half bridge solution was better than a full bridge solution. (The proposed 7 phase inverter requires 14 hexfets and 14 drivers if dual voltage rail half bridges are used, or 28 hexfets and 28 drivers for a full bridge solution.) The half bridge solution does require hexfets with a higher voltage rating, but for the particular inverter design considered, the extra cost per hexfet was such that the half bridge solution was still cheaper than a full bridge solution.

The highest voltage hexfet that could reasonably be afforded was the IRF230. A data sheet for this device is included in Appendix 5.A. The ratings for the IRF230 are as follows.

Maximum drain-source voltage (V_{ds}):	200 volts
Maximum continuous drain current (I_d):	7 amps
Maximum pulsed drain current (I_{dm}):	15 amps
Maximum on resistance ($R_{ds(on)}$):	0.4Ω
Maximum rise time (t_r):	140 ns
Maximum fall time (t_f):	140 ns

The phase voltage waveform required by the motor designed in Chapter 6 is shown in fig. 5.4. The phase is energised for (6/7) of the time. Hence the power input per phase

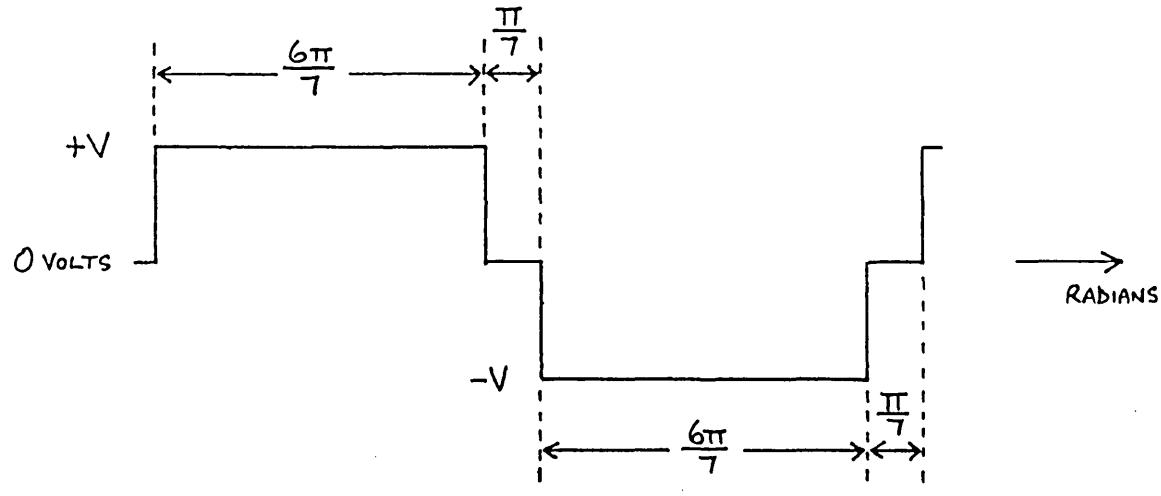


FIG.5.4 ONE CYCLE OF THE REQUIRED PHASE VOLTAGE WAVEFORM

P_{ph} is given by:

$$P_{ph} = (6/7).V_{ph}.I_{ph} \quad \text{watts} \qquad 5.1$$

where V_{ph} and I_{ph} are the values of phase voltage and current respectively. The current is assumed to be constant during the on time, since the inverter and motor voltage waveforms are matched to try to ensure this. For a total inverter power rating of 3 kW, the power per phase is 429 watts. Using the IRF230 MOSFET permits a maximum phase current of 7 amps and equation 5.1 then gives the required phase voltage as 71.5 volts for full power output. If voltage drops in the inverter switches are neglected, the input voltage to the inverter is 71.5 volts and the potential difference between the positive and negative supply rails is 143 volts. Therefore, the hexfet is not working near to its maximum rated voltage of 200 volts, and there is an adequate safety margin should small spikes be present on the supply rails. Hence the IRF230 MOSFET was a suitable transistor to use.

The motor in Chapter 6 was actually designed to run at 80 volts and 6.25 amps per phase because at the time it was envisaged that the IRF232 hexfet would be used. (This hexfet has a rating of 200 volts and 6 amps, and it was planned to slightly overrun the hexfet at full load.) However, when the inverter was finally built, IRF230 hex-

fets were used and the inverter is thus rated for 3 kW total output at 71.5 volts and 7 amps per phase, allowing a slightly larger margin of safety for the hexfet voltage rating, V_{ds}.

5.5 The Inverter Design

As stated earlier in this chapter, the inverter produces a 7 phase square wave output. The inverter is made up of 14 switch units, with pairs of switches connected as half bridges to each phase winding. The inverter is microprocessor controlled. It has been designed so that it is in a safe state if it is not initialised by a key binary word from the microprocessor. The inverter is arranged as shown in fig. 5.5 and forms a self contained unit.

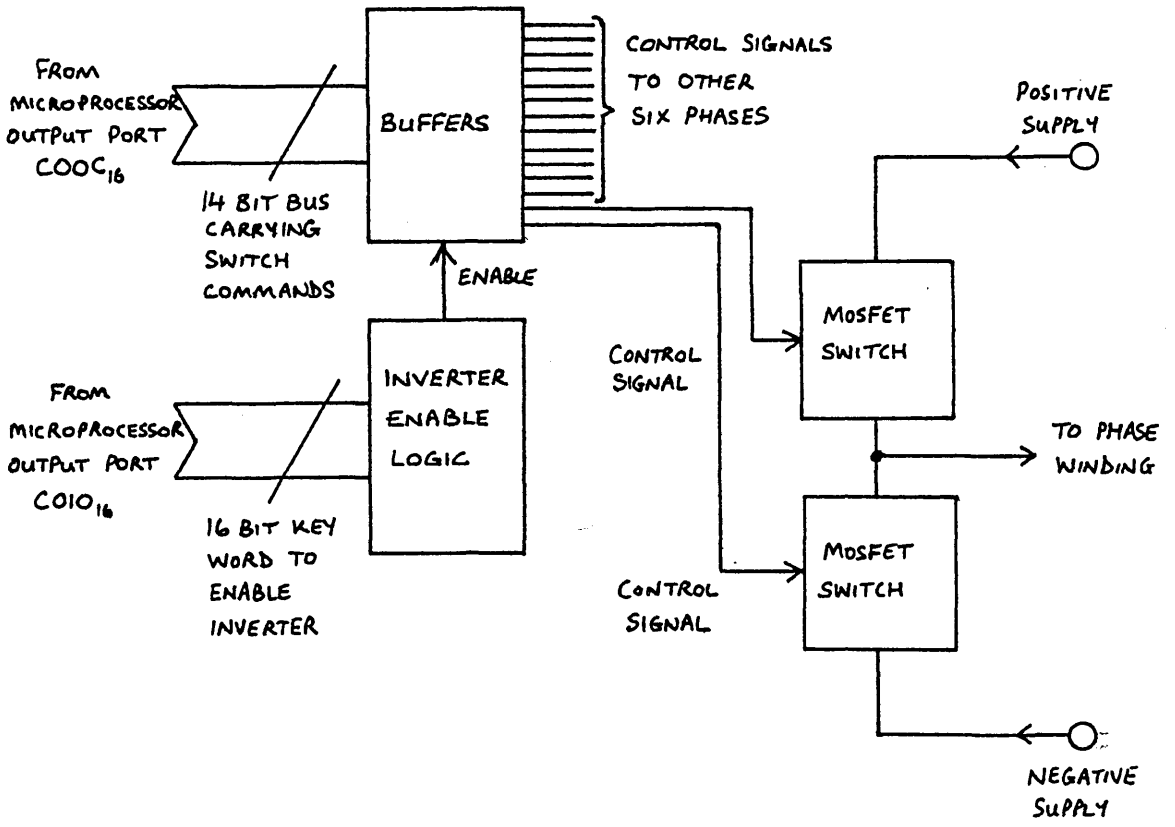


FIG. 5.5. SCHEMATIC SHOWING THE ARRANGEMENT OF THE INVERTER CIRCUITS

It was decided at the outset that all the circuits in the inverter would be assembled on printed circuit boards and mounted in a eurocard rack. The connections between the various boards would then be made between the various edge connectors at the back of the unit. It was planned to mount each half bridge unit on a double-size eurocard, (233.4 x 160 mm). As these boards can be mounted vertically in the rack there is good convection cooling of the hexfets. The various parts of the inverter are described in the following sub-sections.

5.5.1 The Inverter Control Logic

The inverter control logic receives the inverter enable and inverter control signals from the microprocessor output ports $C010_{16}$ and $C00C_{16}$ respectively. Each hexfet in the inverter requires an individual control signal and so there are 14 control signal lines which carry the switching information from the microprocessor to the inverter. (i.e. The control input receives a 14 bit word.) As TTL logic is used in the inverter control circuit, a control input will float to logic 1 if its connecting wire from the microprocessor is broken for any reason. Therefore, "active-low" signals are used to indicate when a hexfet should be on. As a further safeguard it was decided that the control signals should only be gated to the hexfet drivers if the correct inverter enable word was present on the enable inputs. The enable word chosen was 1234_{16}. The circuit diagram of the control logic is shown in fig. 5.6(a). The enable and control words were routed into the inverter unit via two D type connectors and the connections are summarised in fig. 5.6(b). The inverter enable word is decoded by logic gates IC1, IC2 and IC3. When the correct enable word is present the light emitting diode (LED) is lit, and the tristate buffers (IC4, IC5, IC6, IC7) pass the inverter control signals on to the hexfet driver circuits. At all other times the tristate buffer outputs are in a high impedance state and the hexfet drivers are inactive. The buffers can easily sink the necessary current needed

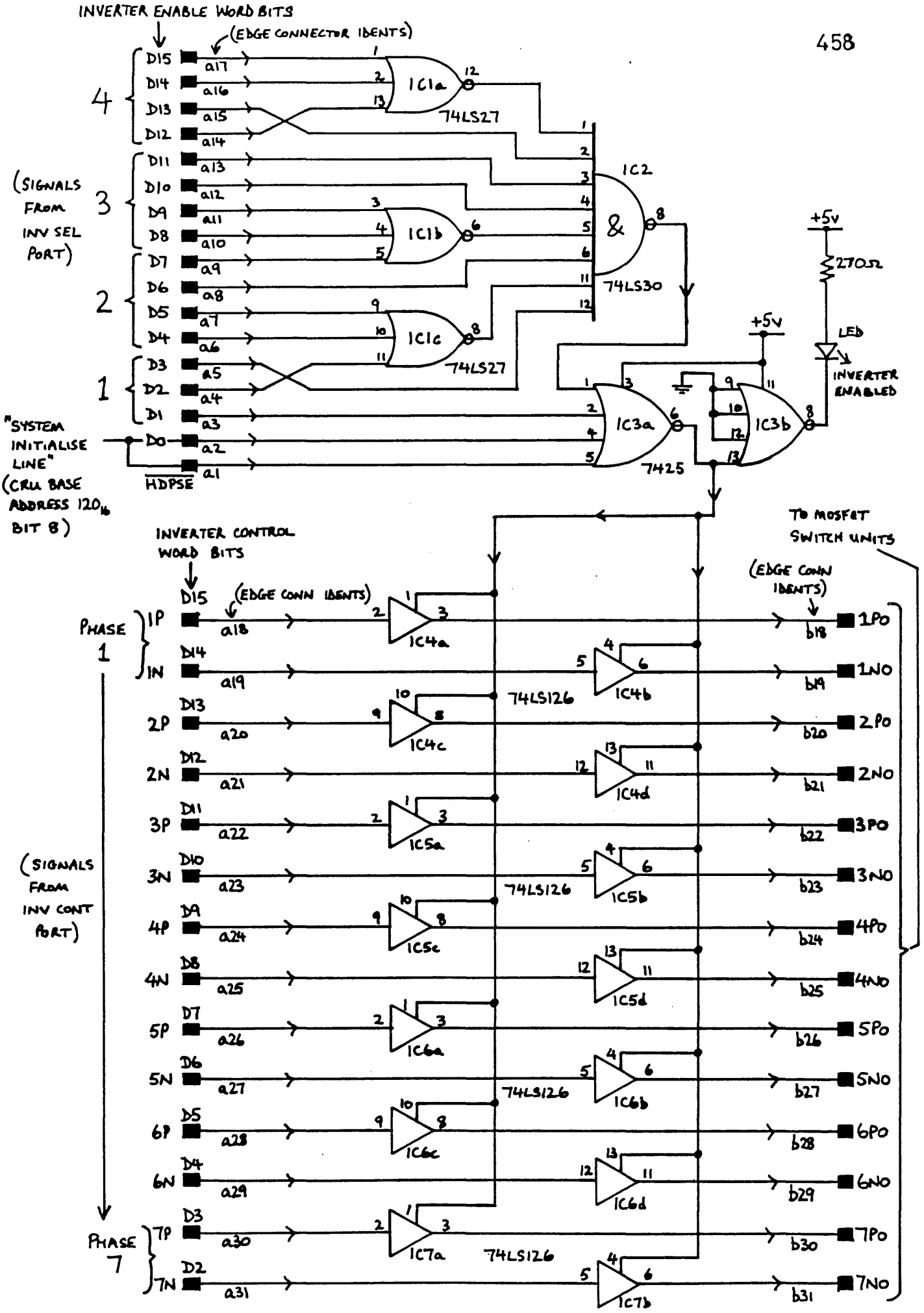


FIG.5.6a. THE INVERTER ENABLE CIRCUIT

	INV SEL PORT	INV CONT PORT		
PIN NUMBER ON D CONNECTOR	INVERTER SELECT (ENABLE) D PLUG: PIN FUNCTIONS ↓	INVERTER CONTROL D SOCKET		
		INVERTER CONTROL WORD BIT	SOCKET PIN FUNCTIONS	
1	DØ (MSB)			
2	D1			
3	D2	D2	PHASE 7 NEG	7N
4	D3	D3	PHASE 7 POS	7P
5	D4	D4	PHASE 6 NEG	6N
6	D5	D5	PHASE 6 POS	6P
7	D6	D6	PHASE 5 NEG	5N
8	D7	D7	PHASE 5 POS	5P
9	D8	D8	PHASE 4 NEG	4N
10	D9	D9	PHASE 4 POS	4P
11	D10	D10	PHASE 3 NEG	3N
12	D11	D11	PHASE 3 POS	3P
13	D12	D12	PHASE 2 NEG	2N
14	D13	D13	PHASE 2 POS	2P
15	D14	D14	PHASE 1 NEG	1N
16	D15 (LSB)	D15	PHASE 1 POS	1P
17	GROUND			
18				
19				
20				
21				
22				
23				
24				
25				

(SIGNAL NAMES ON FIG 5.6a)

FIG. 5.6b. THE SIGNALS CARRIED BY THE VARIOUS PINS OF THE INVERTER SELECT (INV SEL) AND INVERTER CONTROL (INV CONT) D CONNECTORS MOUNTED AT THE REAR OF THE MOSFET INVERTER

to enable the hexfet drivers. The most significant bit (D0) of the inverter enable word is arranged so that it goes to logic 1 and so disables the inverter if the microprocessor is reset. This precaution is necessary because the inverter control and enable words are held on the microprocessor interface board on latches. If the microprocessor is reset for any reason, the latch contents are unchanged and the inverter therefore remains in the state it was in just prior to the microprocessor being reset. Therefore, to prevent this, the D0 bit is fed from a microprocessor signal that does become reset (CRU bit address 130_{16})*. Further details of this signal are given in Chapter 4. The inverter control circuit was constructed on a 100 mm x 160 mm eurocard double sided printed circuit board (pcb). Standard precautions were taken with layout. Adequate decoupling was provided to prevent spurious noise appearing on the tristate buffer outputs. No problems were experienced with the circuit. (* CRU BASE ADDRESS 120_{16}, BIT 8)

5.5.2 The Hexfet Switch Unit

The n type hexfet switches are basically connected in each half bridge circuit as shown in fig.5.7, and the diodes within the hexfets provide the freewheel paths. The form of voltage waveform required by a typical phase of the 7 phase motor is as shown in fig. 5.4. The dead time required between the output switching from positive to negative or vice versa, significantly reduces the risk that the two hexfets may conduct simultaneously, such as can occur when a 180^{o} waveform is generated.

It can be seen in fig. 5.7 that hexfet HF1 has its drain connected to the positive supply and so when it switches, it is the voltage at its source which changes. The hexfet is turned on by a voltage applied between the gate and source, and the fact that the source is not at a fixed potential means that the driver circuit must be free to follow the source potential. However, the signal from the inverter control circuit is referenced to ground, and so

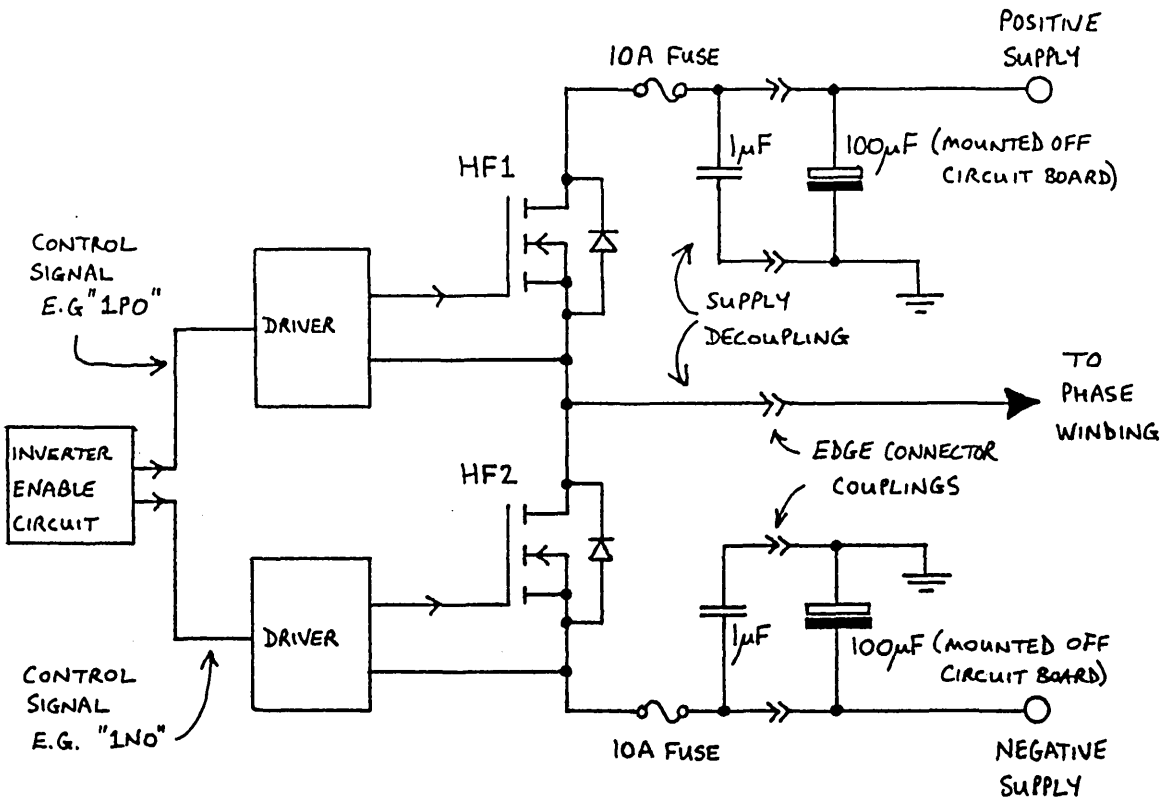


FIG. 5.7. TYPICAL HALF-BRIDGE ARRANGEMENT IN THE INVERTER

some form of level shifting is required for the signal.

The situation with hexfet HF2 is not quite so difficult. The source of this hexfet is connected to the negative supply rail and so the source potential only changes if the supply voltage is changed, and not each time the hexfet switches as in the case of HF1. However, the driver for HF2 is situated at the negative supply potential and so the control signal must also be level shifted.

In view of the fact that both control signals need level shifting it was thought sensible to use a method that electrically isolates the power hexfets from the low power microprocessor system. This can be achieved effectively and cheaply by optical isolators. A pulse transformer solution was rejected because the amplitude of the output voltage from a pulse transformer can suffer significant decreases for certain control signal mark-space ratio and

repetition rate conditions (5.20). The arrangement for a single hexfet is shown in block diagram form in fig. 5.8.

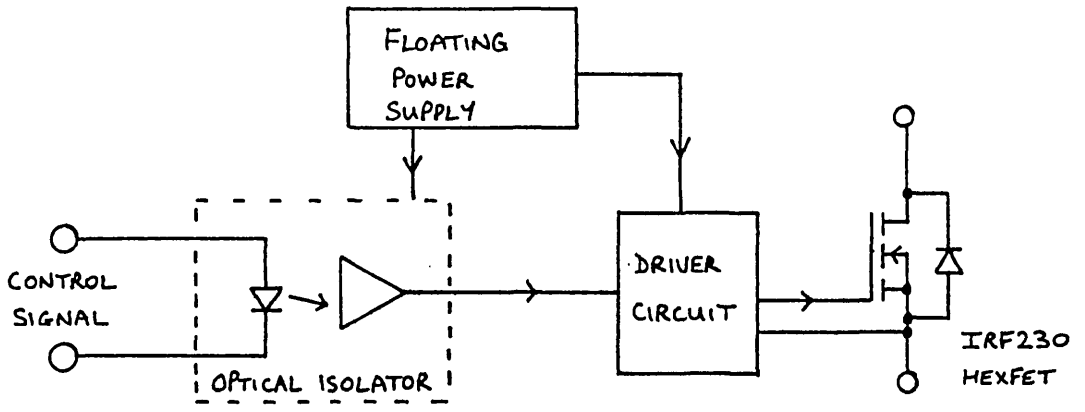


FIG. 5.8. BLOCK DIAGRAM OF THE OPTICALLY ISOLATED HEXFET SWITCH

The output of the optical isolator controls the driver circuit. Both the optical isolator and the driver need a power supply which is able to float up and down with them as the hexfet switches.

In order to achieve fast switching times for the hexfet it is essential that the driver circuit charges and discharges the gate capacitance as fast as possible (5.18, 5.20). A driver circuit with a low output impedance is therefore required. It is not possible to drive MOSFETs directly by single TTL or CMOS gates (as suggested by some manufacturers) if fast switching times are required because they simply cannot source or sink sufficient current. It is necessary to use either a commercial low impedance clock driver such as the DS0026 (made by Intersil and National Semiconductor) or a discrete component driver designed specially for the task. The DS0026 can sink or source 1.5A and switch in 20 ns. However, it was thought that a discrete component driver could be built at a lower cost and provide adequate performance.

5.5.2.1 Hexfet Driver Circuit Design

The basic tasks of the driver circuit are:

(a) To rapidly turn on the hexfet and keep it on. To achieve this the driver has to raise the hexfet gate to source voltage (V_{gs}) as quickly as possible to a level sufficient to support the maximum rated current of the device. The maximum pulsed current that the IRF230 hexfet can handle is 15 amps. From the data in Appendix 5A, it is found that the necessary V_{gs} is typically just less than 7.0 volts.

(b) To rapidly turn off the hexfet and hold it off. To achieve this the driver must reduce V_{gs} below the hexfet threshold voltage as quickly as possible. The minimum threshold voltage for the IRF230 hexfet is 1.5 volts.

Therefore, the driver should ensure that the gate is held at a voltage either greater than 7.0 volts or less than 1.5 volts, and the transition between the two states should be as fast as possible to minimise switching times. It was decided that a 10 volt floating power supply would be sufficient to ensure that V_{gs} could exceed 7.0 volts.

The driver circuit shown in fig. 5.9 was designed to achieve objectives (a) and (b). The diagram includes the optical isolator and the hexfet. The circuit is essentially very simple. The control signal is coupled in by the opto-isolator OP1. The output of the opto-isolator is level shifted to control the low impedance gate driver, made up by a pair of complementary emitter followers. The circuit is powered by a floating +10 volt power supply which is described in sub-section 5.5.2.2.

The optical isolator is a Hewlett PacKard HCPL-2601 device. This is a high speed opto-isolator and its output can change states in typically 25 ns. This speed is ach-

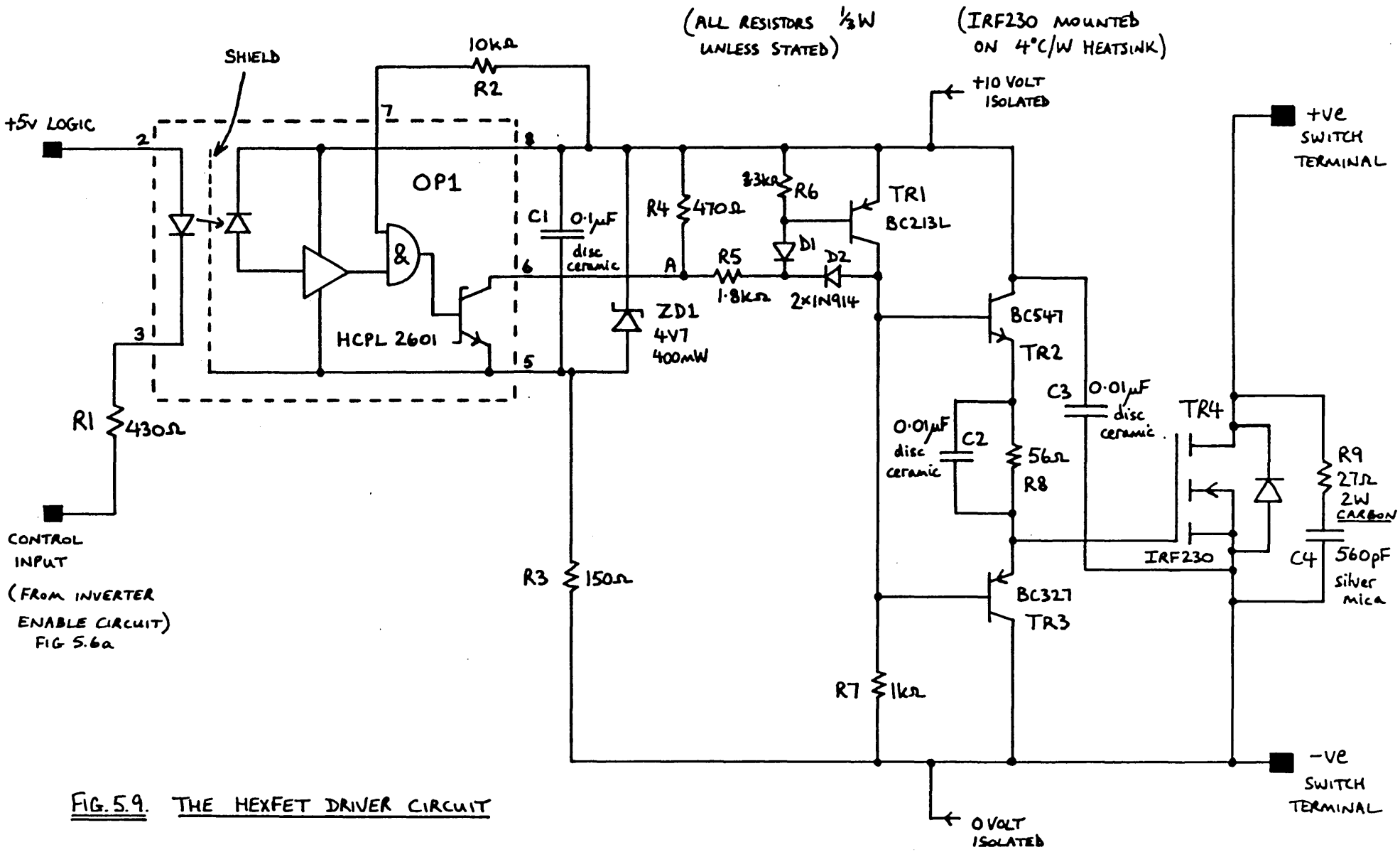


FIG. 5.9. THE HEXFET DRIVER CIRCUIT

ieved by the use of schottky circuitry. The switching speed of the driver circuit is therefore not unduly limited by the rise and fall times of the opto-isolator. In fact the rise and fall times of the input signal to the opto-isolator are about 35 ns, as the signal is supplied by a low power schottky TTL buffer. Therefore, the chosen opto-isolator provides isolation with no degradation to the signal rise and fall times. The opto-isolator requires a +5 volt power supply, which is derived in the driver circuit from the +10 volt supply by using a 4.7 volt zener diode, ZD1, and a resistor, R3. The +5 volt supply is decoupled by a 0.1 μF disc ceramic capacitor C1 which has to be placed directly adjacent to the opto-isolator power supply pins, in order that the specified rise and fall times can be achieved. The HCPL-2601 has an internal shield which results in a common mode transient immunity of 1000 volts/μs, and this essentially means that the output of the device is truly isolated from the input for most applications. A load resistor of 350Ω is recommended to achieve the specified rise and fall times. However, to reduce the supply current requirements it was thought that a small sacrifice in switching times could be tolerated. Therefore, the load resistance has a value of 430Ω made up by the resistor network of R4, R5 and R6.

When the control input to the opto-isolator is at +5 volts (or disconnected due to a broken link) the opto-isolator output is inactive and the load resistors hold point A at +10 volts. Transistor TR1 is held in an off state by resistor R6, and its collector is pulled to 0 volts by R7. The emitter follower, TR3, pulls the hexfet gate down to about +0.6 volts (assuming a 0.6 volt base-emitter voltage drop for a silicon transistor). The gate potential is then well below the hexfet switch on threshold voltage. Therefore, the hexfet is in an off state. When the control input to the opto-isolator is taken to 0 volts, the opto-isolator output transistor turns on and pulls point A down to +5 volts. Transistor TR1 is then forward

biassed and it attempts to pull its collector up to the +10 volt rail. (The switch on time is minimised by deliberately making the base current of TR1 large.) However, the Baker Clamp, formed by D1 and D2, holds the collector voltage at about +9.4 volts. By diverting excess base current, the Baker Clamp prevents excess base charge building up in TR1 and so minimises the switch off time of the transistor. With the collector of TR1 at +9.4 volts, the emitter of transistor TR2 is held at a potential of about +8.8 volts (again assuming a 0.6 volt base-emitter voltage drop). A charging current is supplied by TR2 to the hexfet gate capacitance and it charges up to +8.8 volts. The maximum threshold voltage of an IRF230 hexfet is +3.5 volts and so the device has definitely started to turn on as soon as the gate voltage exceeds +3.5 volts. With a gate voltage of about +8.8 volts the hexfet can support a drain current in excess of 20 amps according to the data sheet in Appendix 5A. The hexfet will therefore be switched on very firmly for drain currents up to the 7 amp continuous rating.

Resistor R8 is included in the circuit to limit the possible short circuit current that can flow if transistors TR2 and TR3 are on together for any reason. Normally only one of the transistors is on, but during the time that the driver output changes state it is possible that both may conduct for a very short period. The 0.01 μF disc ceramic capacitor C2 is included to bypass resistor R8, so that the pulse of current that flows to charge the hexfet gate capacitance is not limited by R8.

Capacitor C3 is included to decouple the +10 volt power supply directly next to the hexfet. C3 is a 0.01 μF disc ceramic capacitor and its main purpose is to supply the pulse of gate drive current; (i.e. it ensures that the inductive impedance of the +10 volt power supply does not limit the gate charging current). Capacitor C3 is chosen by assuming that it supplies the gate current entirely. If it is desired to charge the gate capacitance C_g from

zero volts up to the on value voltage $V_{gs(on)}$ in a time t_{on} seconds, the required charging current I_g (assumed constant over the period) is given by:

$$I_g \cdot t_{on} = C_g \cdot V_{gs(on)} \qquad 5.2$$

i.e. $$I_g = \frac{C_g \cdot V_{gs(on)}}{t_{on}} \quad \text{Amps} \qquad 5.3$$

If the current I_g is supplied by capacitor C3, the capacitor voltage will fall by an amount ΔV_{c3} given by:

$$I_g \cdot t_{on} = C3 \cdot \Delta V_{c3} \qquad 5.4$$

i.e. $$\Delta V_{c3} = \frac{I_g \cdot t_{on}}{C3} \quad \text{volts} \qquad 5.5$$

For the IRF230, the data sheet in Appendix 5A gives the maximum gate capacitance as 1000 pF. It was decided that the voltage across capacitor C3 should not fall by more than 1 volt during the turn on time. Combining equations 5.2 and 5.4 gives:

$$C3 = \frac{C_g \cdot V_{gs(on)}}{\Delta V_{c3}} \quad \text{Farads} \qquad 5.6$$

Substituting a value of $V_{gs(on)}$ = 8.8 volts and ΔV_{c3} = 1.0 volt gives:

$$C3 = \frac{1000 \times 8.8}{1.0} \quad \text{pF} \qquad 5.7$$

$$C3 = 8800 \text{ pF} \qquad 5.8$$

A value of 0.01 μF was actually chosen. C3 recharges to 10 volts during the time between switchings.

The time taken to charge the gate voltage up to 8.8 volts was chosen as 100 ns to ensure that the hexfet turn on time was reasonably fast. Using equation 5.3 then yields:

$$I_g = \left(\frac{1000 \times 10^{-12} \times 8.8}{100 \times 10^{-9}} \right) \text{ Amps} \qquad 5.9$$

$$I_g = 88 \text{ mA} \qquad 5.10$$

Therefore transistors TR2 and TR3 can be low power signal transistors. The actual transistors used were chosen because they had good current gains, high cut off frequencies, and they were readily available. Resistors R1 to R8 inclusive were 0.3 watt components. It is not thought worthwhile discussing the design of the driver in any further detail since it is essentially a very simple circuit. However, some explanation of the design and operation of the driver power supply is necessary, and this is covered in the following sub-section.

5.5.2.2 The Floating Power Supplies for the Hexfet Driver Circuit

Each hexfet driver requires an independent power supply that can float freely above ground potential with minimal leakage capacitance to ground. The transfer of power from a ground referenced power source to the floating hexfet driver is most easily achieved by use of a d.c. to d.c. converter, with electrical isolation achieved with a high frequency transformer. The power supply block diagram is shown in fig. 5.10. To minimise the size of transformer it is necessary to use a high switching frequency and 100 kHz was chosen. A pair of transistors chop the +15 volt d.c. supply rail to generate the 100 kHz square wave input to the transformer primary. The secondary output is bridge rectified and smoothed to give a floating +10 volt d.c. supply. The magnitude of supply current required by a hexfet driver circuit is a maximum of about 45 mA and the

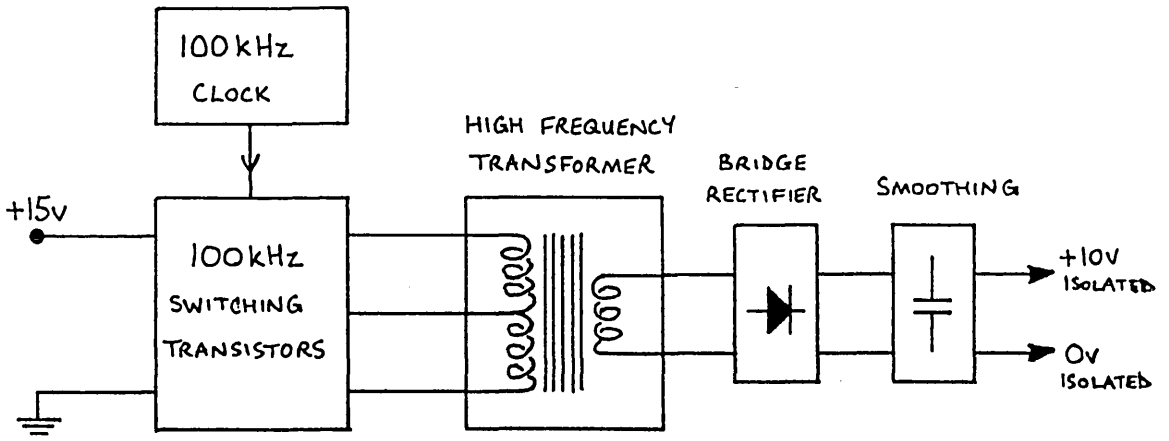


FIG. 5.10. BLOCK DIAGRAM OF THE FLOATING 10 VOLT POWER SUPPLY

power supply was designed to be able to supply this continuously. The circuit diagram of a typical power supply is shown in fig. 5.11, and the idealised waveforms with the circuit supplying zero load current are shown in fig. 5.12. The waveforms assume that the circuit has been operating for some time so that the flux waveform is symmetrical about zero. The polarity of the flux and secondary voltage waveforms are arbitrary.

The circuit consists of a transformer with a centre tapped primary winding. The transformer is driven in the well known push-pull mode and its operation need not be discussed in great detail. However, several features deserve some mention.

(a) The ends of the primary winding, A and B, are connected to zero volts alternately by transistors TR3 and TR4 respectively. Due to the action of the transformer primary, these transistors must be able to withstand a voltage at least twice that of the d.c. supply rail. In practice voltage spikes caused by the transformer leakage inductance when the transistors switch, result in a need for an even higher voltage rating. In operation it was found that spikes of about 50 volts were present on the trans-

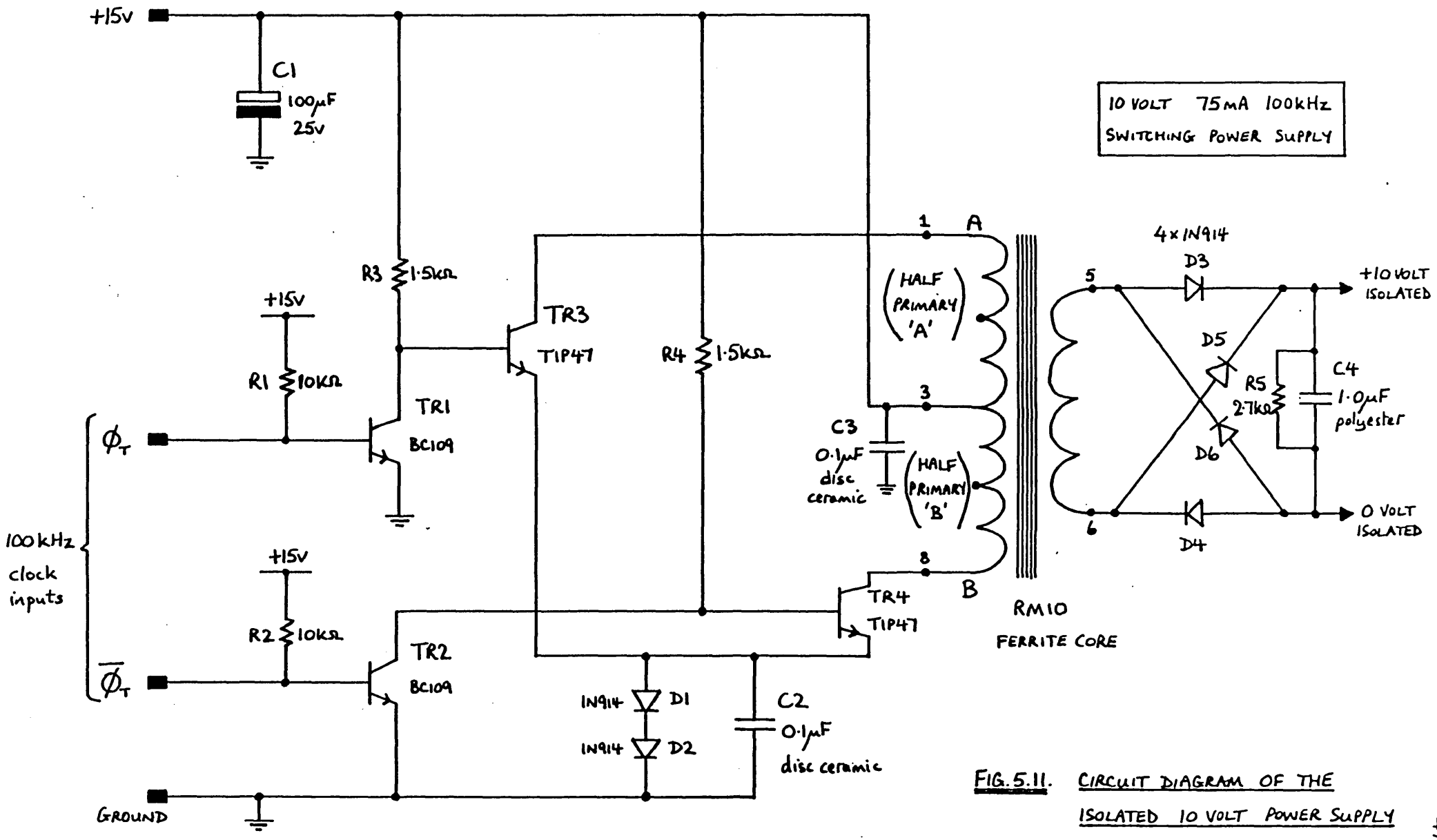


FIG. 5.11. CIRCUIT DIAGRAM OF THE ISOLATED 10 VOLT POWER SUPPLY

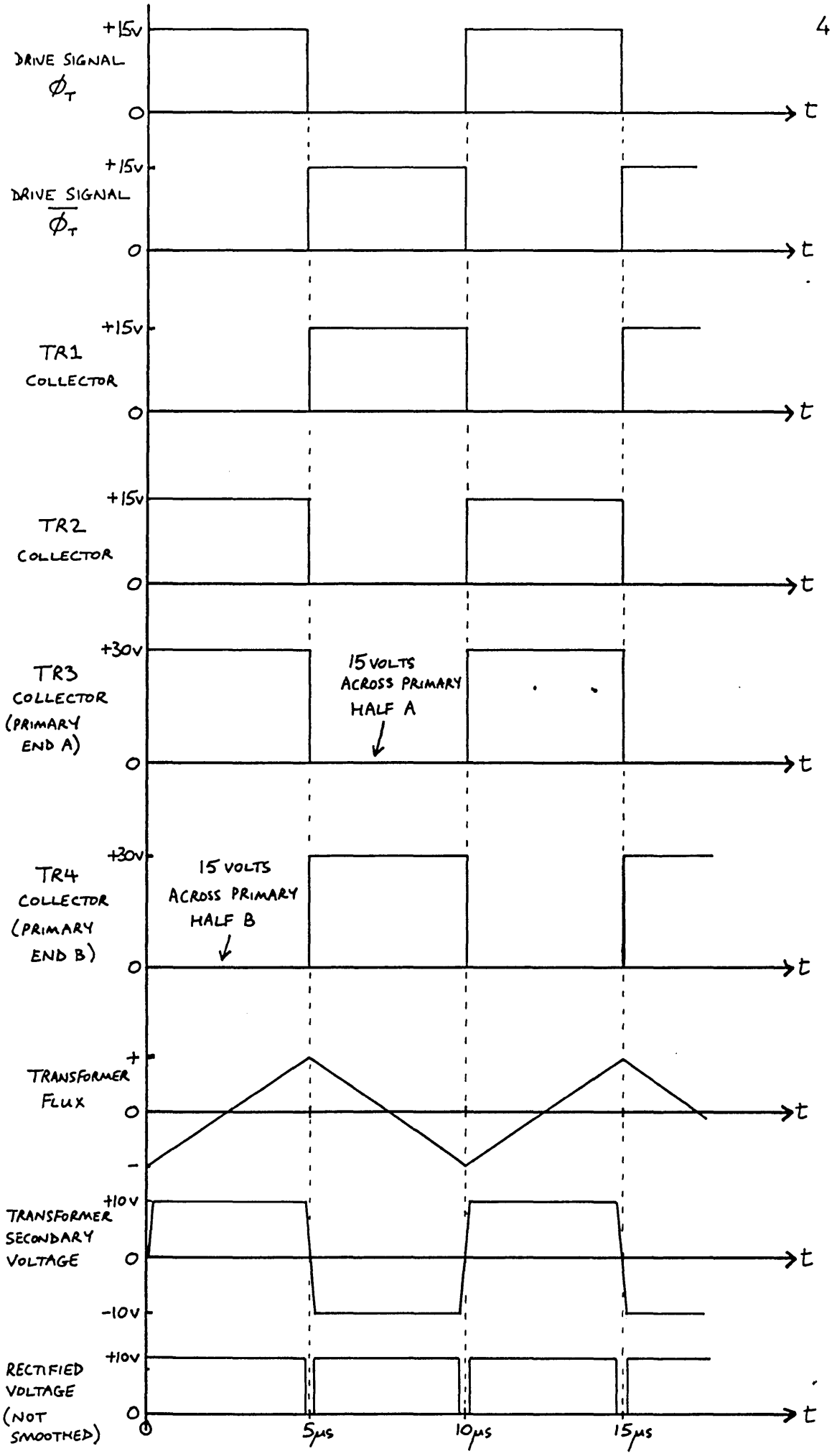


FIG. 5.12 IDEALISED WAVEFORMS IN THE ISOLATED 10 VOLT POWER SUPPLY

former primary. To ensure reliable operation, small power transistors (TIP47) with a collector-emitter voltage rating of 250 volts were used for TR3 and TR4.

(b) To minimise switching losses in TR3 and TR4 it is desirable to make them switch as fast as possible. An improved turn-off time was achieved by arranging their emitters to sit at a potential of about +1.2 volts. Thus, when the base of TR3 or TR4 is pulled to zero volts, the base emitter junction is reverse biassed and the excess base charge is removed quickly. The 1.2 volt reverse bias is generated by two silicon diodes D1 and D2, and smoothed by a disc ceramic capacitor C2. As either TR3 or TR4 is on at any time, there is always some current flowing thr-ough the diodes to generate the required voltage drop.

(c) Transistors TR1 and TR2 were included to buffer the 100 kHz drive signals, and also to ensure that if a drive signal became disconnected, the relevant transformer switching transistor (TR3 or TR4) would switch off rather than stay on permanently. This prevents a large primary current flowing under such conditions.

(d) To prevent the transformer core drifting towards sat-uration, it is necessary that the antiphase drive signals have an exact 50:50 mark-space ratio. This ensures that the volt-second product on the transformer primary is zero over a whole cycle. The 100 kHz transformer drive signals are derived from the clock circuit shown in fig. 5.13. The circuit provides buffered drive signals for the 14 power supplies used in the 7 phase inverter. A schmitt trigger oscillator (IC1a) generates a 200 kHz waveform which is divided by a flip-flop (IC3) to give a pair of antiphase 100 kHz signals with 50:50 mark-space ratios. The signals are buffered by 7406 open collector TTL inver-ters (IC4-IC8) so that they can drive directly into the +15 volt power supply circuits. The clock circuit outputs can be forced into a high state by a toggle switch SW1. When this occurs all the power supplies are effectively

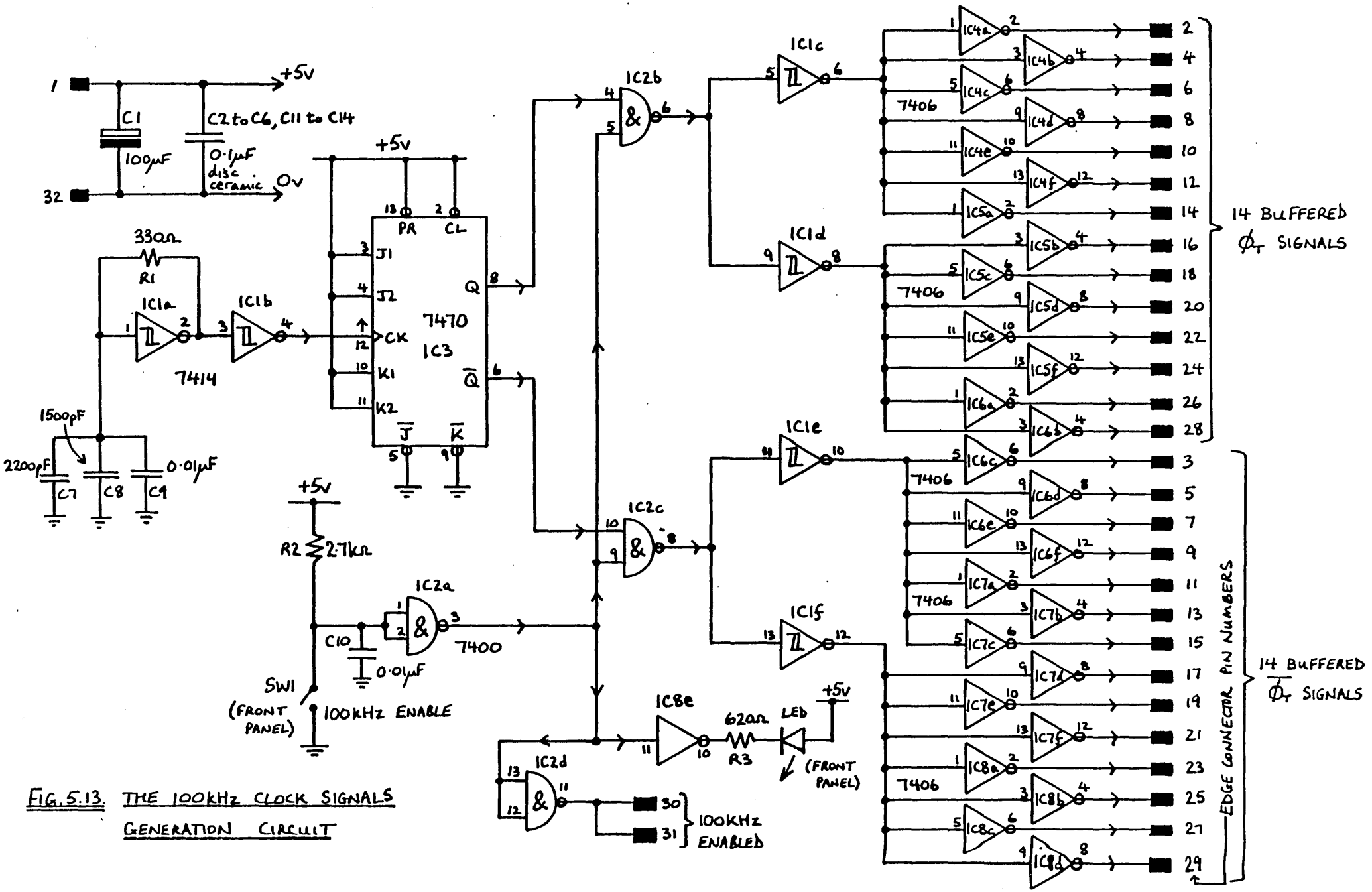


FIG. 5.13. THE 100kHz CLOCK SIGNALS GENERATION CIRCUIT

switched off because the transformer switching transistors are all held in an off state.

(e) The number of turns on each half of the transformer primary should be large enough to ensure that the transformer core does not work close to its saturation point. This reduces the chance of the flux density creeping towards saturation.

The secondary winding must be able to supply 45 mA continuously, and 50 mA was actually used in the calculations. The transformer ratio required must take into account any voltage drops in the system. The applied primary voltage is not 15 volts due to the 1.2 volt emitter potential of TR3 and TR4, plus their respective collector-emitter voltage drops when on (say 0.3 volts). (i.e. The primary voltage is about 13.5 volts.) The secondary voltage must be 10 volts plus an extra 1.2 volts to take account of the bridge rectifier voltage drop; i.e. a secondary voltage of 11.2 volts. Therefore, the transformer voltage ratio is 13.5:11.2 and so the primary current required to support a secondary current of 50 mA is 41.5 mA. In addition the primary winding must carry the magnetising current. A maximum value of magnetising current was set at 20 mA, giving a total peak primary current of about 60 mA. A suitable gauge of enamelled copper wire for both primary and secondary windings is S.W.G. 34, which is rated at 80 mA. This is more than adequate for the primary as each half of the primary only conducts current for half a cycle. The number of primary turns needed to limit the magnetising current I_m to a peak of 20 mA is calculated using the simple equation:

$$V_p = L_p \frac{dI_m}{dt} \qquad 5.11$$

where V_p is the applied primary voltage and L_p the primary inductance.

i.e. $$\frac{V_p}{L_p} = \frac{dI_m}{dt} \qquad 5.12$$

or $$\frac{V_p}{L_p} \simeq \frac{\Delta I_m}{\Delta t} \qquad 5.13$$

As the applied voltage is constant, the rate of change of current is fixed as shown by equation 5.12. At a frequency of 100 kHz, the primary winding is energised for a period of 5 μs. Therefore, if the magnetising current is not to exceed 20 mA in a time of 5 μs, the rate of change of current, $\Delta I_m/\Delta t$, has a value not exceeding 4000 amps per second. Substituting this value into equation 5.13 with a primary voltage value of 13.5 volts gives:

$$L_p = 3.375 \text{ mH} \qquad 5.14$$

For a given transformer core, the inductance L_p is related to the reluctance R by the relationship:

$$L_p = \frac{N_p^2}{R} \qquad 5.15$$

where N_p is the number of primary turns. An RM10 ferrite core was selected as a possible transformer core and its data sheet is included in Appendix 5B. The RM10 core has a reluctance of 2.5×10^6 H^{-1} and so the number of primary turns N_p calculated using equation 5.15 is 92. A figure of 100 turns was chosen and this results in the primary having a total of 200 turns, with the secondary having 83 turns. An RM10 core can easily accomodate 283 turns of S.W.G. 34 wire. The peak magnetising flux density is only 9.6 mT and so the transformer core is well away from the ferrite saturation limit of 300 mT.

The leakage inductance and eddy current losses in a transformer winding depend on the way in which the primary and secondary are interleaved. The subject has been dealt with by Snelling (5.31). Leakage inductance causes voltage

spikes as the transistors switch off, and also droop in the secondary voltage as the load current is increased. It was suggested that a suitable arrangement of primary and secondary windings was as shown in fig. 5.14, in which each half of the primary winding is split into two fifty turn sections.

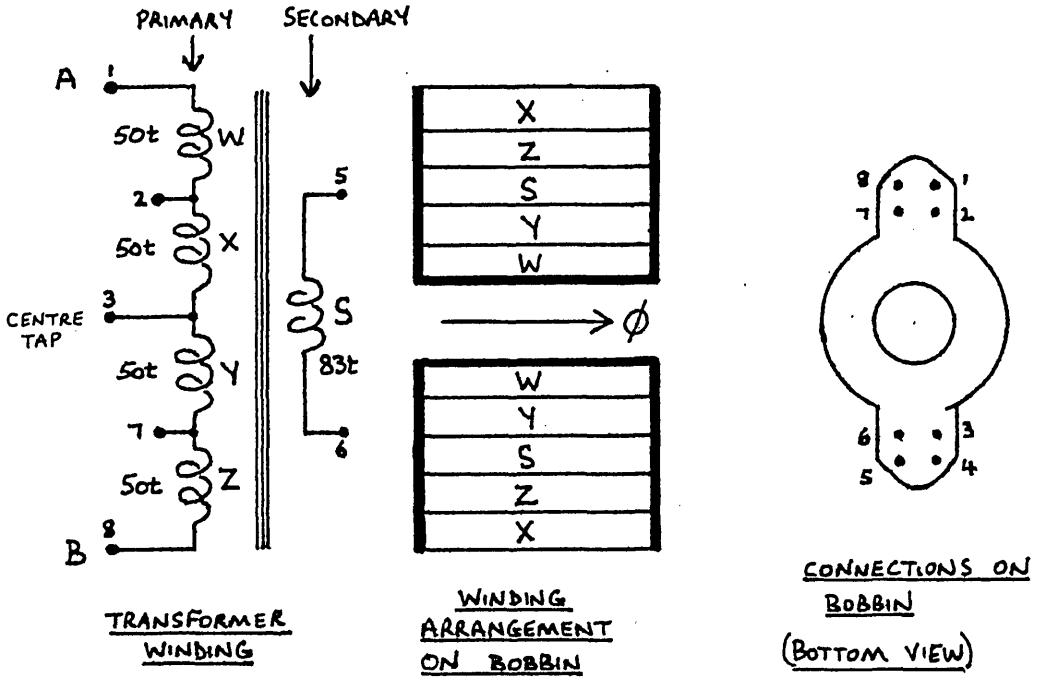


FIG. 5.14 WINDING DETAILS FOR THE 100kHz FERRITE TRANSFORMER

The final check required in the transformer design was to assess the increase in winding resistance caused by skin effect when high frequencies are used. Snelling (5.32) states that the a.c. resistance of a wire is given by:

$$R_{ac} = R_{dc} + R_{se} \qquad 5.16$$

i.e. $$R_{ac} = R_{dc}\,(1+F)\ \Omega \qquad 5.17$$

where R_{se} is the increase in resistance due to skin effect and F is the skin effect factor. The skin effect factor is a function of d/Δ where d is the wire diameter and Δ is the penetration depth of the current. The penetration depth Δ is defined as:

$$\Delta = \sqrt{(\rho_c / \pi \mu_o \mu_c f)} \quad \text{metres} \qquad 5.18$$

where ρ_c is the conductor resistivity, μ_c is the relative permeability of the conductor and f is the frequency. For round copper wire it is found that the skin effect is negligible for ratios of d/Δ less than 2. Therefore, to ensure that skin effect is not a problem, the conductor diameter d must be chosen so that:

$$d/\Delta \leqslant 2 \qquad 5.19$$

The diameter of S.W.G. 34 wire is 2.34×10^{-4} m and so the actual value of d/Δ for copper wire at 100 kHz is:

$$d/\Delta = \frac{2.34 \times 10^{-4}}{\left[1.79 \times 10^{-8}/(\pi \times 4\pi \times 10^{-7} \times 1 \times 10^{5})\right]^{\frac{1}{2}}} \qquad 5.20$$

$$d/\Delta = 1.1 \qquad 5.21$$

Therefore, skin effect in the transformer winding is negligible.

(f) The final power supply component whose selection requires some explanation is the smoothing capacitor, C4. This capacitor would not be required if the secondary waveform was a perfect square wave, and the diodes D3 to D6 had zero switching times. However, the switching times of transistors TR3 and TR4 plus those of diodes D3 to D6 result in small gaps in the rectified output voltage as indicated on the idealised rectified output shown in fig. 5.12. For the duration of the gap the load current must be supplied by a smoothing capacitor. Allowing for switching times, transistor TR3 or TR4 conducts for a period just less than 5 μs at 100 kHz, and the turn off time is certainly much less than 1 μs. If it is assumed that the turn off time t_{off} is 1 μs and the load current I_1 is 50 mA, then a value of smoothing capacitor C_{sm} can be calculated

such that the output voltage does not fall by more than ΔV (= 1 volt say).

i.e. $$C_{sm} = \frac{I_1 \cdot t_{off}}{\Delta V} \quad \text{Farad} \qquad 5.22$$

$$C_{sm} = \frac{50 \times 10^{-3} \times 1 \times 10^{-6}}{1.0} \quad F \qquad 5.23$$

$$C_{sm} = 0.05\ \mu F \qquad 5.24$$

A 1.0 μF polyester low inductance capacitor was chosen and this was more than adequate to smooth the supply. A 2.7 kΩ resistor, R5, was connected across the smoothing capacitor to prevent the output voltage rising to a large value with no load connected. (This is caused by the voltage spikes generated by the leakage inductance of the transformer winding.)

5.5.2.3 The Testing of the Prototype Switch Unit

A single +10 volt power supply and driver circuit were built in prototype form to check the circuit operation. A photograph of the circuit is shown in fig. 5.15. The resistances of the prototype transformer were 2.0Ω for each half primary and 1.6Ω for the secondary.

The power supply was found to give an output of 11.20 volts on no load (i.e. just with the 2.7 kΩ bleed resistor connected) and it drew 40 mA from the +15 volt rail. The output voltage fell to 8.24 volts when a 100Ω resistor was used to load the power supply, and a current of 102 mA was drawn from the +15 volt rail. Therefore when supplying 82.4 mA, the power supply has an efficiency of 44%. This is not a good efficiency but the design procedure was not aiming for good efficiency. The power supply was left running with the 100Ω load for one hour with no problems, and transistors TR3 and TR4 were only slightly warm by the end of the test. The driver circuit was then connected to the

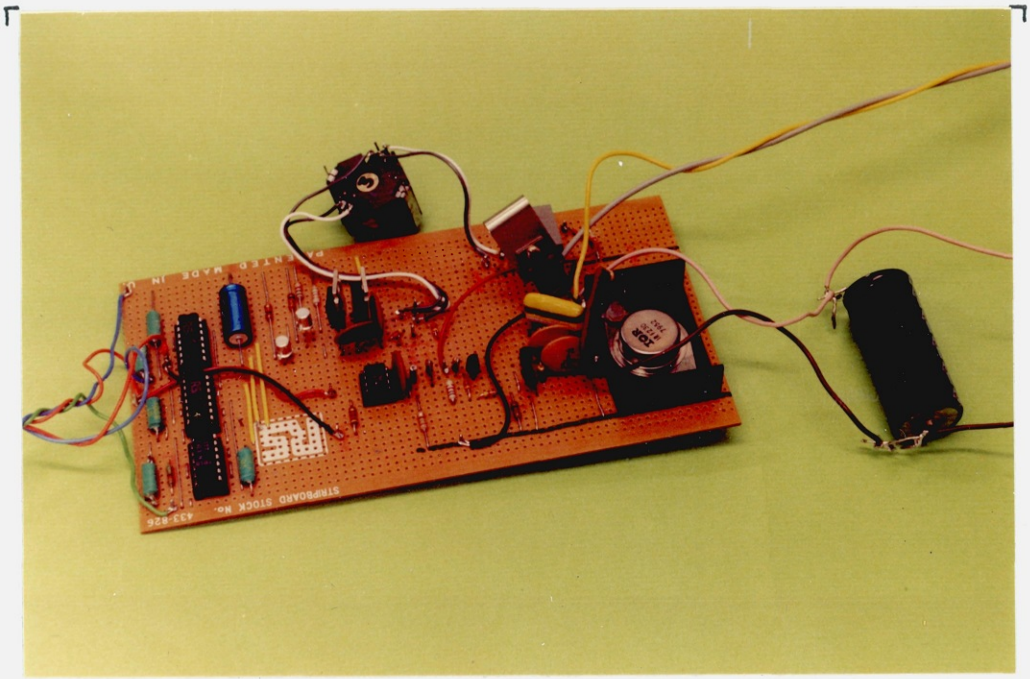


FIG. 5.15. THE PROTOTYPE HEXFET SWITCH UNIT

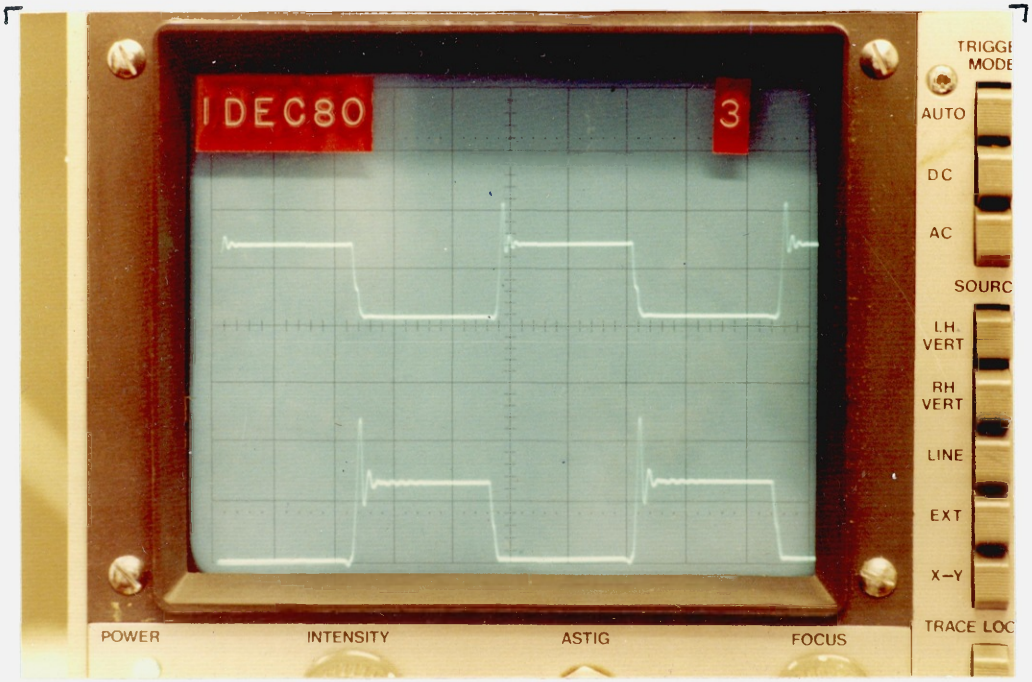


FIG. 5.16. TYPICAL COLLECTOR WAVEFORMS OF TRANSISTORS TR3 AND TR4 IN THE ISOLATED 10 VOLT POWER SUPPLY
Y AXIS : 20v/cm X AXIS: 2μs/cm

power supply. The power supply output voltage was 9.27 volts and the current drawn from the +15 volt rail was 62 mA. A hexfet was then connected to the driver and the hexfet gate was driven with a range of frequencies from d.c. to 25 kHz. It was found that the power supply output voltage stayed constant at 9.27 volts and the +15 volt supply rail current also remained constant at 62 mA. The typical collector waveforms of transistors TR3 and TR4 in the power supply are shown in fig. 5.16, and their emitter waveform is shown with one of the collector waveforms in fig. 5.17. It can be seen that the transistors are switching on and off in less than 1 μs.

To check the hexfet switching waveform, a 19Ω rheostat was connected to the drain as shown in fig. 5.18. As the rheostat load was slightly inductive, a fast diode BYW29 (35 ns reverse recovery time) was connected across the rheostat to provide a freewheel path. The circuit was supplied from a 0 to 100 volt variable voltage supply. During initial switching tests it was found that large voltage spikes (+280 volts) were present on the drain at switch off. It was realised that these spikes were due to the inductance of the power supply leads combined with the rapid switching speed of the hexfet. The supply was decoupled with a 33 μF electrolytic capacitor plus a 0.47 μF polyester capacitor. The capacitors were placed as close as possible to the hexfet and rheostat load. The turn off spike was virtually removed by the decoupling. In addition a series RC network of 27Ω and 560 pF was placed across the hexfet drain and source terminals. This reduced the overshoot spike slightly and it was thought worthwhile to leave it connected. Fig. 5.19 shows the typical gate and drain voltage waveforms for the hexfet switching at 600 Hz with a 100 volt supply. The average load current measured was 2.5 amps. It can be seen that the gate-source "on" voltage is about 8.0 volts. No oscillations are evident and the switch off time appears to be very fast. Increasing the decoupling capacitance to 132 μF reduced the voltage overshoot to an even greater extent so that it was only

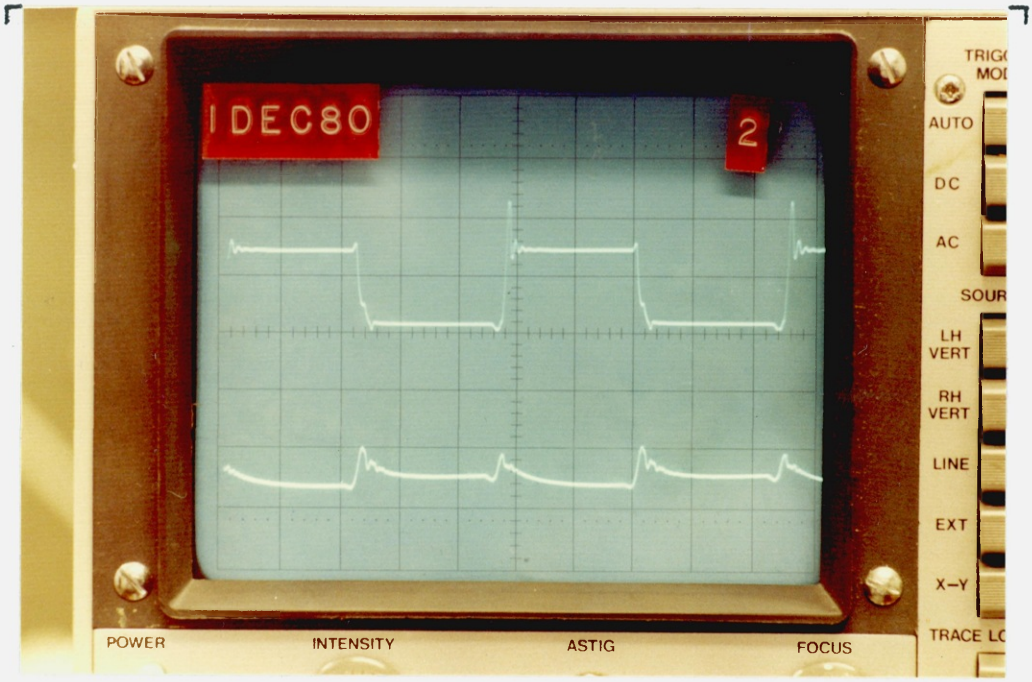


FIG. 5.17. 10 VOLT POWER SUPPLY WAVEFORMS — TOP TRACE : TRANSISTOR TR3 COLLECTOR WAVEFORM (20V/cm) — BOTTOM TRACE : EMITTER WAVEFORM OF TRANSISTORS TR3 AND TR4 (1V/cm). TIMEBASE 2μs/cm

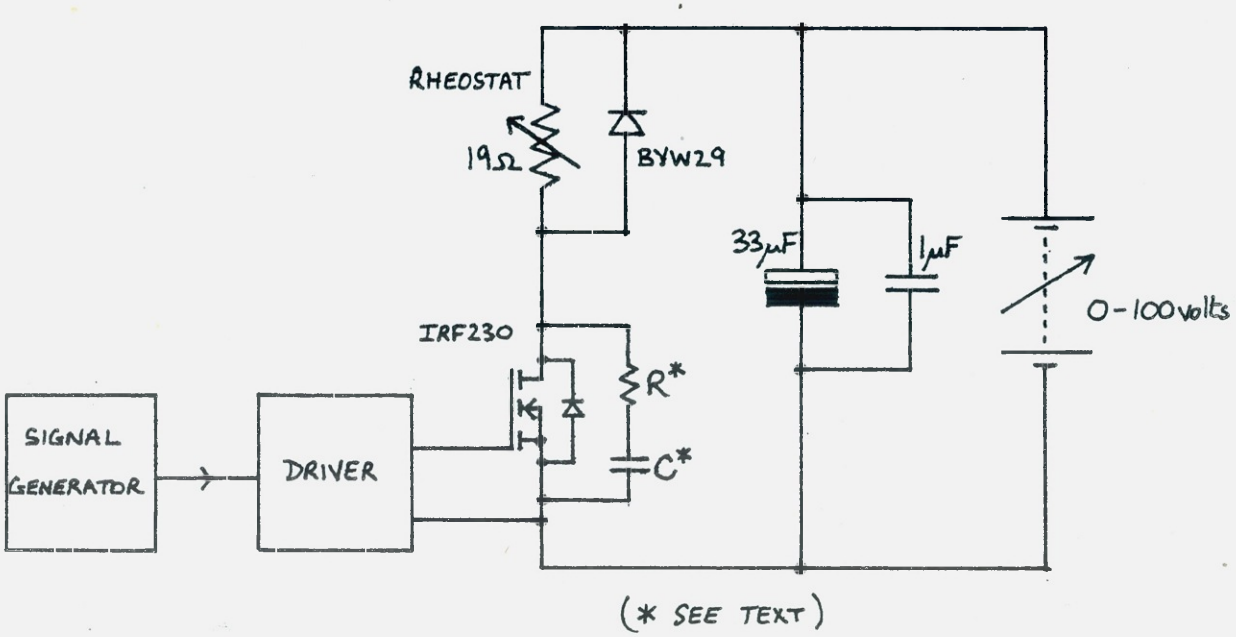


FIG. 5.18. THE CIRCUIT USED TO CHECK THE HEXFET SWITCHING WAVEFORMS

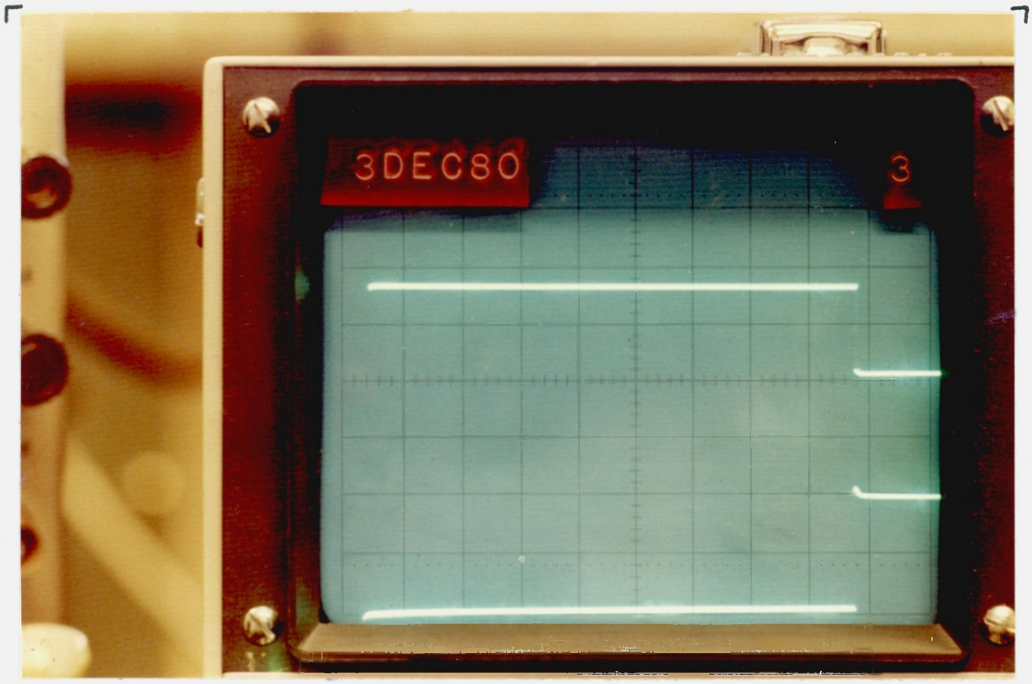


FIG. 5.19 TYPICAL GATE (TOP TRACE) AND DRAIN (BOTTOM TRACE) VOLTAGE WAVEFORMS FOR THE IRF230 HEXFET SWITCHING 2·5 AMPS (AVERAGE) AT 600 Hz ON A 100 VOLT SUPPLY RAIL.
TOP: 5v/cm — BOTTOM: 50v/cm — TIMEBASE: 0·1ms/cm

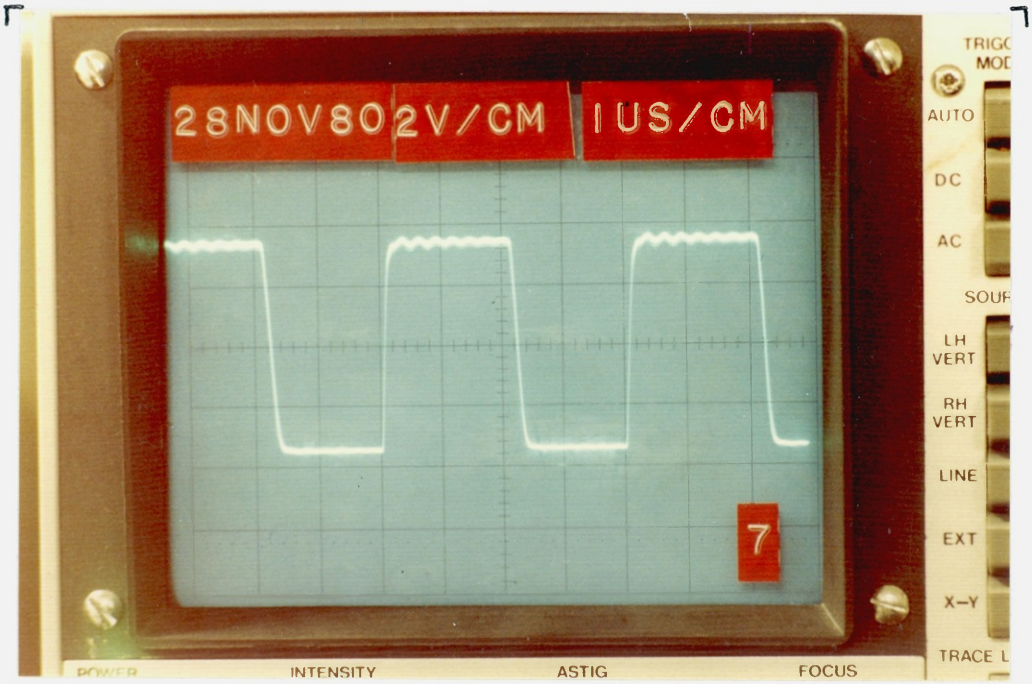


FIG. 5.20. HEXFET GATE-SOURCE VOLTAGE WAVEFORM WITH THE GATE DRIVEN AT 250 kHz (HEXFET DRAIN UNENERGISED)
Y AXIS: 2V/cm — X AXIS: 1μs/cm

about 4 volts. The circuit was operated at 100 volts over a wide frequency range up to 25 kHz and no problems were encountered. It ran for a total of about 5 hours without failure.

To get some idea of the gate-source voltage rise and fall times, the gate was driven at 250 kHz with the hexfet drain unenergised. The gate-source voltage waveform is shown in fig. 5.20. The rise and fall times are about 100 ns and 200 ns respectively, and so the driver is certainly working satisfactorily. The actual gate voltage in fig. 5.20 is lower than usual because at the time that the waveform was taken a transformer with a 75 turn secondary was being tried. It is interesting to see that the driver circuit could be used to drive a hexfet at frequencies up to 250 kHz.

The rise and fall times of the hexfet when it was switching at frequencies between d.c. and 25 kHz were difficult to measure on the available oscilloscopes but they were better than 300 ns. The rheostat was connected onto the hexfet source and the tests were repeated. No problems were encountered with this configuration and so it was decided to go ahead and transfer the power supply and driver circuits onto a printed circuit board.

5.5.2.4 The Design of the Printed Circuit Boards for the Hexfet Switch Circuits

The 100 kHz clock circuit (shown in fig. 5.13) that drives all the inverter floating 10 volt gate driver power supplies was built on a 100 mm x 160 mm printed circuit eurocard. Standard layout precautions were taken and the board was well decoupled.

As mentioned earlier, it was planned from the beginning that each pair of hexfet switches (forming a half bridge) should be mounted on a 233.4 mm x 160 mm eurocard printed circuit board. During the design of the prototype

printed circuit board layout, several factors were carefully considered. They were:

(a) the thickness of copper conductor needed on the printed circuit boards;

(b) the need (if any) for snubber components;

(c) the size of hexfet heatsink;

(d) the layout of the board;

(e) the size of connector needed to couple the power lines onto the board;

(f) the connection (or not) of the logic zero volts rail to the power zero volts rail;

(g) the need for supply decoupling close to the hexfets;

(h) the need for some form of current limiting protection, to prevent damage to the printed circuit board in the event of the hexfets being damaged and shorting the power supplies.

The various factors are discussed individually below.

(a) The thickness of copper track required was very quickly settled. The largest continuous current that the printed circuit board must handle is 7 amps. Printed circuit board is generally available with a copper foil thickness of 0.00135 inches (known as 1 oz/ft^2 board) or 0.0027 inches (2 oz/ft^2 board). The current capacity versus track width for a standard 40°C temperature rise is dealt with by Eisler (5.33) and fig. 5.21 shows the design curves that can be used. The curves show that a 7 amp current can be carried by a 1/8 inch wide track on 1 oz or 2 oz copper board. It was decided to make the main current

carrying tracks much wider than this to minimise the temperature rise and keep the inductance of the tracks as small as possible.

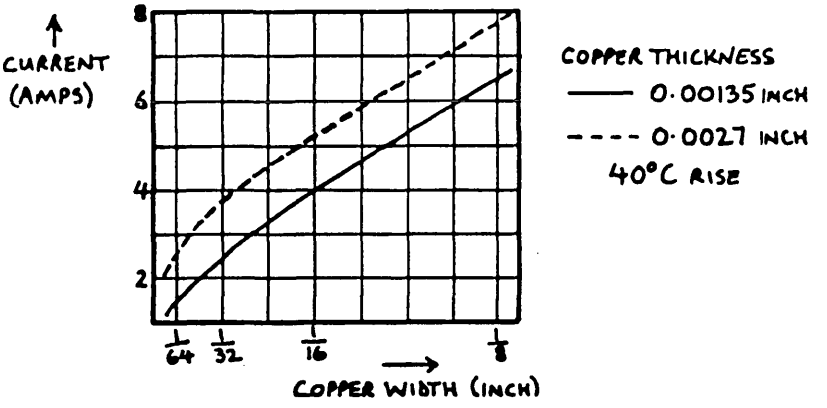


FIG. 5.21. CURRENT CAPACITY VERSUS TRACK WIDTH FOR THE STANDARD COPPER FOIL THICKNESSES AVAILABLE ON PRINTED CIRCUIT BOARDS

The main connections to the hexfet drain and source were therefore made at least 3/16 inch wide and the track going to the phase winding was 0.7 inch in width. The prototype board was etched on 2 oz copper laminate.

(b) At the time that the inverter was designed, it was not very clear from the manufacturers data whether any snubber protection is needed for MOSFET transistors. The data sheets all stated that MOSFETs do not suffer from second breakdown and none of the suggested applications showed any snubbers. Indeed, the data did not even mention that the parasitic transistor forming a freewheel diode was present within the MOSFET, and hence it was originally planned to use discrete freewheel diodes. It was only as the design was being finalised that information was published by Severns (5.21) of Intersil, revealing that a transistor exists within the MOSFET and this transistor can breakdown if the rate of change of voltage across the drain-source terminals is too great. The situation was still not very clear though because very little information was available about either the turn on/turn off times of the parasitic transistor or its limiting value of dV/dt.

It was therefore impossible to decide if any snubbering was required. At this stage it was decided to make use of the parasitic transistor by allowing it to perform as the freewheel diode, and it was simply assumed that it would be fast enough for the job. Reports from other users of MOSFETs indicated that reliable performance with inductive loads had been achieved with simple resistor/capacitor snubbers, the resistor having a typical value of 10Ω to 300Ω and the capacitor values of 100 pF to 1000 pF. These values were essentially found experimentally by trial and error.

It was found during the initial testing of the "bread-board" prototype that a snubber made up of a 27Ω carbon resistor in series with a 560 pF silver mica capacitor did reduce the overshoot spike on the drain by a noticeable amount, and so it was decided to use these values on the printed circuit board prototype. The 27Ω resistor value was chosen so that the peak current rating of the hexfet is not exceeded at any time. The highest voltage that the snubber capacitor can be charged up to is 160 volts if the inverter is running on ±80 volt rails. The IRF230 hexfet can carry a peak pulsed current of 15 amps and so at turn on it could support a 7 amp load current plus an 8 amp snubber capacitor discharge current.

i.e. $$R_{snubber} = \frac{160 \text{ volts}}{8 \text{ amps}} \qquad 5.25$$

$$R_{snubber} = 20 \text{ ohm} \qquad 5.26$$

A value of 27Ω was chosen, thus limiting the peak capacitor discharge current to 5.9 amps. The power rating of the resistor was chosen on the basis that the inverter might be used for PWM at 25 kHz at some time in the future. Assuming the capacitor discharges totally during the hexfet on period, the energy stored in the capacitor must be dissipated in the snubber resistor. If the hexfet switching

frequency is f_s, the power dissipated during discharging is given by:

$$P_{snub} = \text{energy stored x switching frequency}$$

$$P_{snub} = 1/2\ C_{snub} \cdot V_{ds}^{2} \cdot f_s \qquad 5.27$$

i.e.

$$P_{snub} = 1/2(560\text{x}10^{-12})160^{2}\text{x}25\text{x}10^{3} \text{ watts}$$

$$P_{snub} = 0.18 \text{ watts} \qquad 5.28$$

As there is an equal power loss in the resistor when the snubber capacitor charges up, the total rating of the snubber capacitor must be at least 0.36 watts and a 2 watt resistor was actually used. Carbon resistors and silver mica capacitors were used for the snubber components to minimise inductance. The time constant of the snubber is 15 ns and so it has ample time to charge and discharge with the hexfet switching at frequencies up to 25 kHz.

No pretence can be made that the snubbers were carefully designed. The chosen values have been remarkably successful as only two hexfets have failed in use and in both cases it was for reasons other than inadequate snubbering. As mentioned in section 5.4.3 it is now known that MOSFETs really do not need snubber protection and the only time when care should be exercised is when the parasitic transistor is used as a freewheel diode. In such cases the rate of voltage change should be limited during the first 15 to 20 ns of its reverse recovery time. Considering the inverter described in this chapter, the conduction pattern of the two hexfets per phase is such that if the parasitic transistor of one hexfet is conducting or is just recovering from conducting a freewheel current, it is that hexfet which next turns on anyway, and so its parasitic transistor is at no risk during recovery from a large dV/dt caused by the opposite hexfet turning on. The small load inductance of the 7 phase motor is also a reason for the

reliable operation of the hexfets, since there is far less stored energy available to dissipate in the parasitic transistors.

(c) The choice of a suitable heatsink for the hexfets was determined both by the thermal limits of the hexfets and the space restrictions on the printed circuit board. To ensure that the two hexfets operate with similar drain to source on resistances it is best to mount them on the same heatsink so that they are essentially at the same temperature. The power P_{fet} dissipated in a MOSFET when carrying a drain current I_d is simply given by:

$$P_{fet} = 0.5\, I_d^2 . R_{ds(on)} \qquad 5.29$$

where $R_{ds(on)}$ is the drain to source on resistance at the operating temperature of the MOSFET. $R_{ds(on)}$ for the IRF-230 hexfet varies with temperature as shown graphically in Appendix 5A, and it can be approximated by the equation:

$$R_{ds(on)} = R_{ds(on)25}\left[(0.5T_j/75) + 5/6\right] \Omega \qquad 5.30$$

where $R_{ds(on)25}$ is the drain to source resistance at 25°C and T_j is the hexfet junction temperature in degrees centigrade. Substitution of equation 5.30 into 5.29 gives:

$$P_{fet} = 0.5 I_d^2 . R_{ds(on)25}\left[(0.5T_j/75) + 5/6\right] \text{ watts} \qquad 5.31$$

The phase voltage and current waveforms required by the square wave motor have the form shown in fig. 5.4. Each hexfet conducts for 6/14 of each cycle and the average power dissipated by each device is therefore given by:

$$P_{fet(av)} = (6/14) P_{fet} \qquad 5.32$$

The heatsink must dissipate the power developed by the two hexfets and its temperature T_h is given by:

$$T_h = 2(P_{fet(av)})R_h + T_{amb} \qquad 5.33$$

where T_{amb} is the ambient air temperature, and R_h is the thermal resistance of the heatsink. (The power from each hexfet enters the heatsink at different points but this is neglected here.) The hexfet junction temperature T_j is then given by:

$$T_j = T_h + P_{fet(av)}(R_{jc} + R_{ch}) \qquad 5.34$$

where R_{jc} is the thermal resistance from the hexfet junction to the case, and R_{ch} is the thermal resistance from the hexfet case to the heatsink. (i.e. The thermal resistance of the mica washer required to electrically isolate the cases of the two hexfets.) Combining equations 5.33 and 5.34 yields:

$$T_j = T_{amb} + P_{fet(av)}\left[2R_h + R_{jc} + R_{ch}\right] {}^{\circ}C \qquad 5.35$$

Substitution of equations 5.32 and 5.31 into 5.35 then gives:

$$T_j = \frac{\left[T_{amb} + \frac{5}{28}\left\{I_d^2 . R_{fet(on)25}(2R_h + R_{jc} + R_{ch})\right\}\right]}{\left[1 - \frac{1}{700}\left\{I_d^2 . R_{fet(on)25}(2R_h + R_{jc} + R_{ch})\right\}\right]} \; {}^{\circ}C \qquad 5.36$$

The largest heatsink that could easily be accomodated on the printed circuit board was one having a thermal resistance of 4°C/W. Choosing the maximum $R_{ds(on)25}$ of 0.4Ω and substituting the values:

$T_{amb} = 25^{\circ}C$

$R_h = 4^{\circ}C/W$

$R_{jc} = 1.67^{\circ}C/W$

$R_{ch} = 1.0^{\circ}C/W$

and $I_d = 7.0$ A

into equation 5.36 yields:

$$T_j = 88.9^{\circ}C = 89^{\circ}C \qquad 5.37$$

This is well below the maximum junction temperature of 150°C. For $T_j = 89^{\circ}C$, the value of $R_{ds(on)}$ given by equation 5.30 is 0.57Ω as compared with 0.595Ω obtained from the manufacturers characteristic in Appendix 5A. Equation 5.30 is therefore quite accurate enough for use in the heatsink calculation. The heatsink temperature at the rated current of 7 amps is obtained by use of equations 5.31, 5.32 and 5.33, and is found to be 73.0°C. It should be noted that the calculated junction and heatsink temperatures are the worst case values because the maximum value of $R_{ds(on)25}$ has been used. At the calculated case temperature of 79°C, the hexfet can safely dissipate 42w according to Appendix 5A and this is well above the actual dissipation at full drain current.

(d) To minimise any possible interference between the logic signal inputs and the hexfet outputs, it was decided that the control signals and low voltage power supplies should be carried onto the inverter pcb via one connector (PLA), whilst the high voltage hexfets supplies and the phase output should be carried on a second connector, (PLB). In addition, it was thought that the hexfet heatsink should be separated from the pcb by a gap of 5 mm to prevent excessive heat transfer to other parts of the circuit. The layout of components on the pcb is shown schematically in fig. 5.22.

(e) The most convenient way of connecting onto a eurocard board is via standard eurocard plugs and sockets. Each pin of the connector is rated at 1 amp and so a suitable 7 amp connection can be made up by paralleling 7 or more pins.

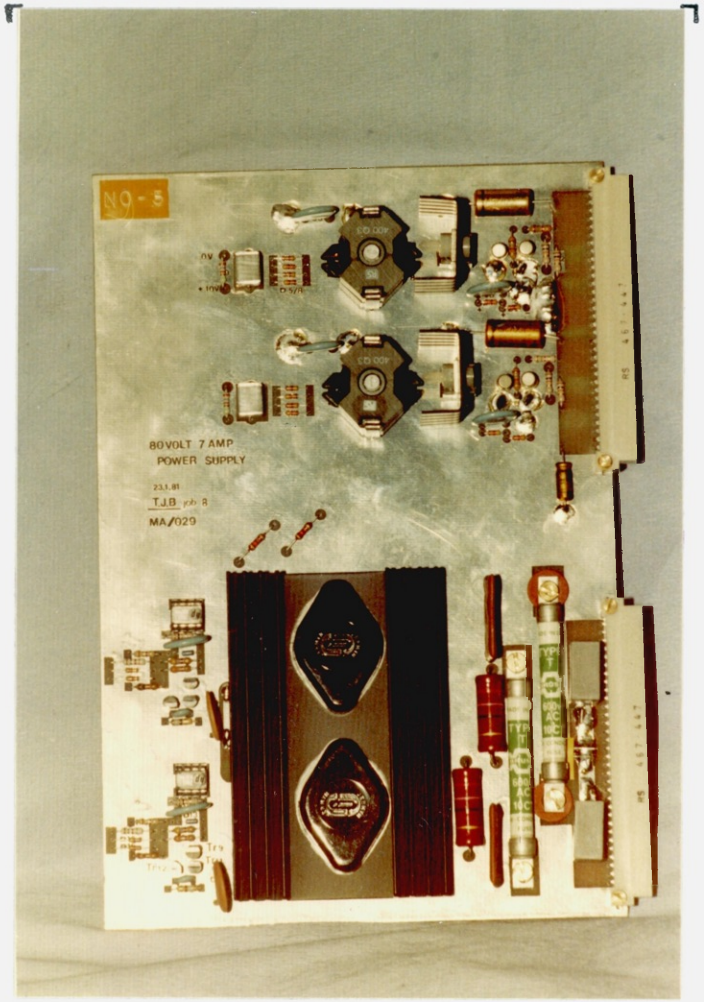


FIG. 5.23. PHOTOGRAPH OF A TYPICAL INVERTER PRINTED CIRCUIT BOARD

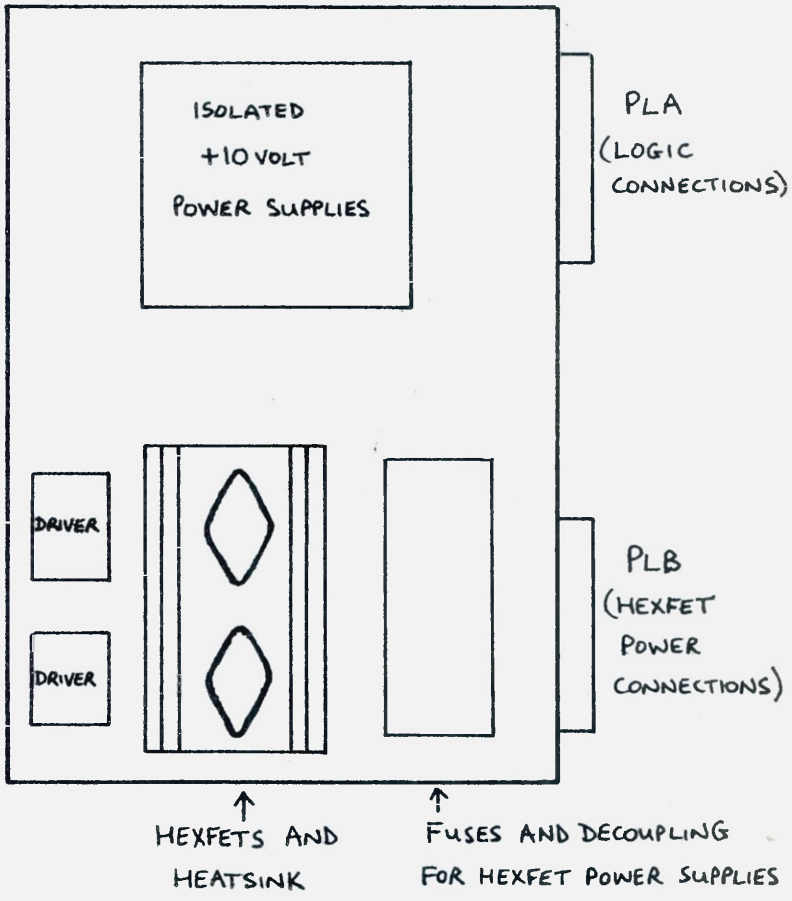


FIG. 5.22 LAYOUT OF THE COMPONENTS ON THE HEXFET SWITCH PRINTED CIRCUIT BOARD

(f) Double sided printed circuit laminate was used for the inverter board. The copper on the component side of the board was arranged as a ground plane connected to the zero volts of the logic power supplies. It was not known whether any problems would arise if the zero volt connection of the large hexfet power supplies was also connected to the ground plane. Therefore, it was left unconnected but with suitable tracks available to enable it to be connected if necessary.

(g) It was known that in order to prevent large voltage spikes on the drain of each hexfet, it would be necessary to decouple the inverter variable voltage power supplies as close to the hexfets as possible. It was decided to use a 1 μF polyester capacitor to decouple each supply on the printed circuit board, and an extra 100 μF electrolytic capacitor per supply mounted on the sockets into which the pcb plugs.

(h) To prevent huge fault currents in the event of both hexfets going short circuit, it was decided to mount 10 amp HRC semiconductor protection fuses in series with the supply to each hexfet. Whilst these fuses cannot rupture fast enough to protect a hexfet, they can prevent serious damage to the pcb conductors.

The decoupling capacitors and fuses are indicated on fig.5.7.

The printed circuit board was designed, etched, assembled and successfully tested. The control signals needed for the two hexfets were conveniently generated by the TMS 9900 microprocessor system. This allowed test frequencies up to 3400 Hz to be used. It was found that at certain load currents the hexfets produced a clicking noise which was found to be a high frequency oscillation. Such oscillations can be removed by placing a ferrite bead or resistor (50Ω - 100Ω) in series with the hexfet gate lead (5.18). However, the oscillation was removed by connecting together

CONNECTOR PLA	
PIN NUMBER	FUNCTION
1 to 7	+15 VOLT LOGIC SUPPLY
8 & 9	NO CONNECTION
10	100 kHz DRIVE SIGNAL ϕ_{T1} (Q_A)
11	100 kHz DRIVE SIGNAL $\overline{\phi_{T1}}$ $(\overline{Q_A})$
12	NO CONNECTION
13 to 17	LOGIC GROUND
18	NO CONNECTION
19 to 23	+15 VOLT LOGIC SUPPLY
24 & 25	NO CONNECTION
26	100 kHz DRIVE SIGNAL ϕ_{T2} (Q_B)
27	100 kHz DRIVE SIGNAL $\overline{\phi_{T2}}$ $(\overline{Q_B})$
28 & 29	NO CONNECTION
30	"POSITIVE" HEXFET OPTO ISOLATOR INPUT A
31	"NEGATIVE HEXFET" OPTO ISOLATOR INPUT B
32	+5 VOLT OPTICAL ISOLATOR SUPPLY

CONNECTOR PLB	
PIN NUMBER	FUNCTION
1 to 8	POSITIVE HEXFET SUPPLY
9	NO CONNECTION
10 & 11	POWER GROUND
12	NO CONNECTION
13 to 20	PHASE OUTPUT
21	NO CONNECTION
22 & 23	POWER GROUND
24	NO CONNECTION
25 to 32	NEGATIVE HEXFET SUPPLY

FIG. 5.24. EDGE CONNECTOR DETAILS FOR THE HEXFET INVERTER PRINTED CIRCUIT BOARD

the zero volt rails of the logic and variable voltage power supplies. A photograph of the finished inverter board is shown in fig. 5.23. The connections onto the board are summarised in fig. 5.24.

5.5.3 The Inverter Assembly and Initial Testing

The printed circuit board layouts for the inverter control logic, the 100 kHz clock logic, and the hexfet switch units are given in Appendix 5C. A photograph of the assembled inverter control logic and 100 kHz clock logic boards is shown in fig. 5.32.

Seven hexfet printed circuit boards were constructed. (Three of them on 1 oz copper board due to a shortage of 2 oz board.) A eurocard rack was constructed to house all the inverter boards. The +15 volt and +5 volt power supplies necessary for the logic circuitry were also housed in the eurocard rack. The circuit diagram of the logic power supplies is shown in fig. 5.26.

The circuit boards were arranged in the eurocard rack as shown in fig. 5.25.

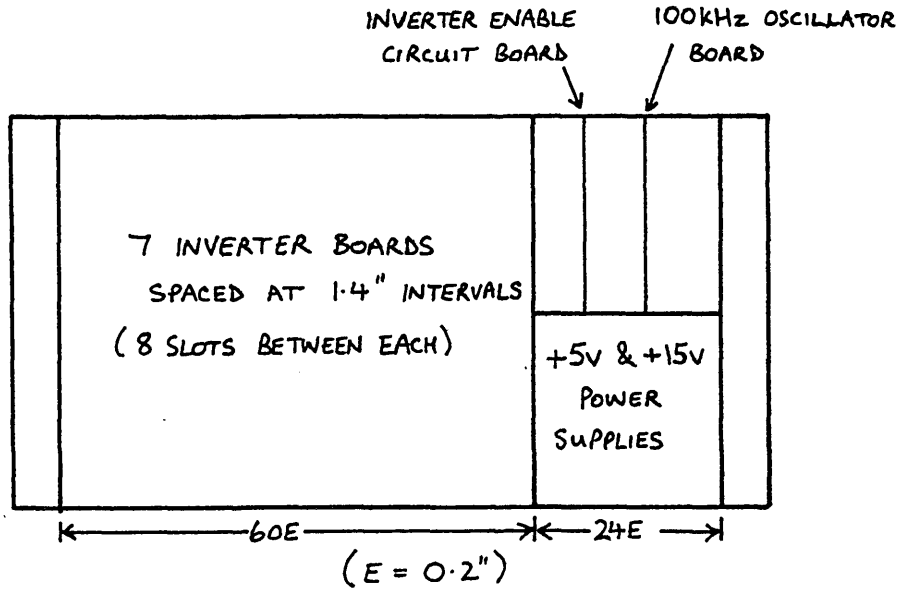


FIG 5.25. SCHEMATIC FRONT VIEW OF INVERTER EUROCARD RACK

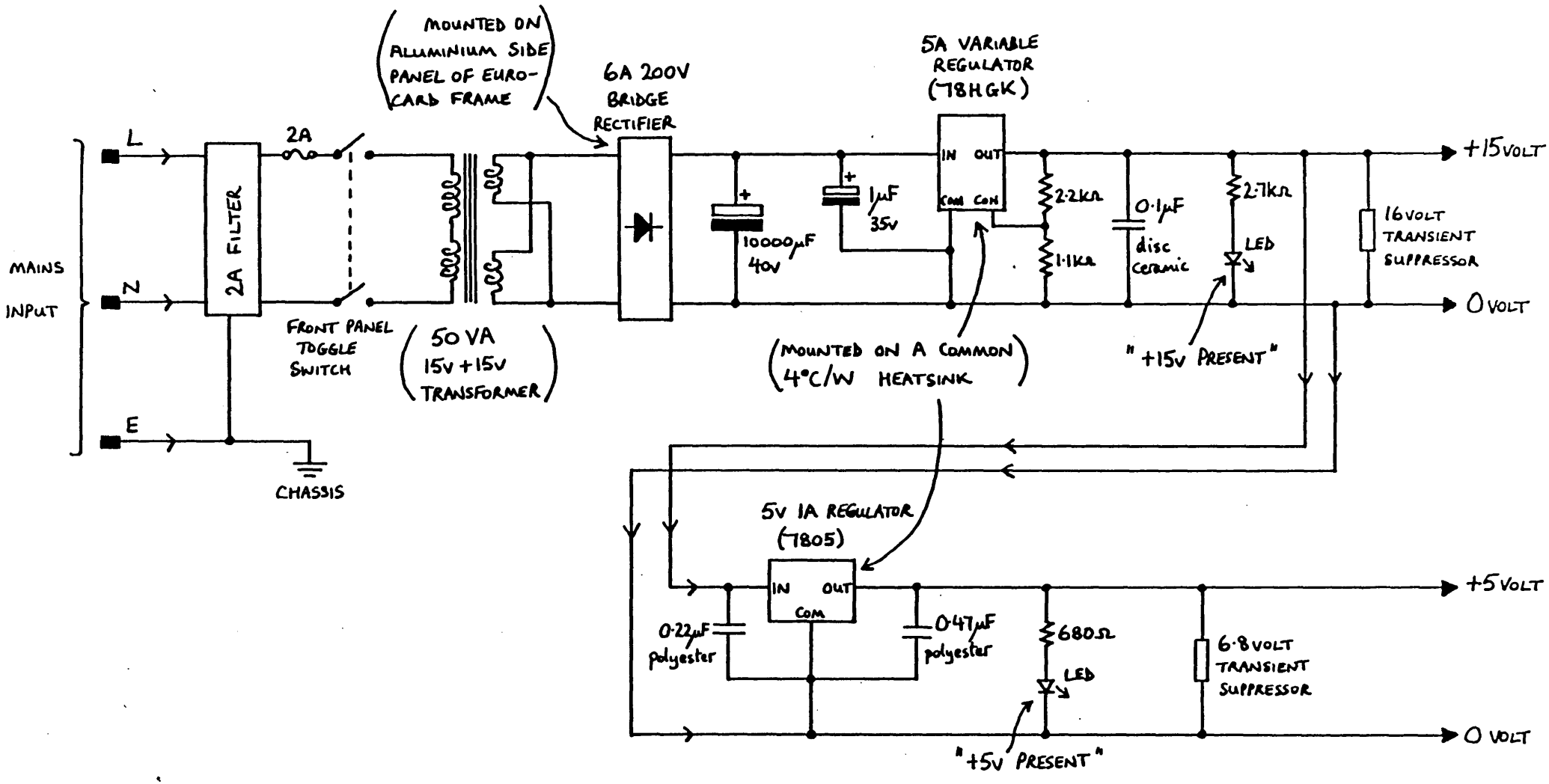


FIG. 5.26. CIRCUIT DIAGRAM OF THE LOGIC POWER SUPPLIES IN THE HEXFET INVERTER RACK UNIT

The inverter control board and the 100 kHz oscillator board were placed in the top half of the rack so that their signals could run directly across the back of the rack to the various inverter board logic connectors. The layout of the back of the inverter is shown in fig. 5.27(a). The signal connections were made by the wirewrap technique. The main power supply rails were formed by several parallel strands of S.W.G. 15 copper wire running across the back of the inverter. The 100 μF decoupling capacitors for each inverter board were connected across the "copper busbars" at appropriate points. The inverter control and inverter enable signals were brought in via two 25 way D connectors at the back of the inverter. The positive and negative variable voltage supplies were brought in via d.i.n. rail mounted 47 amp terminals and the phase outputs were made via 27 amp terminals.

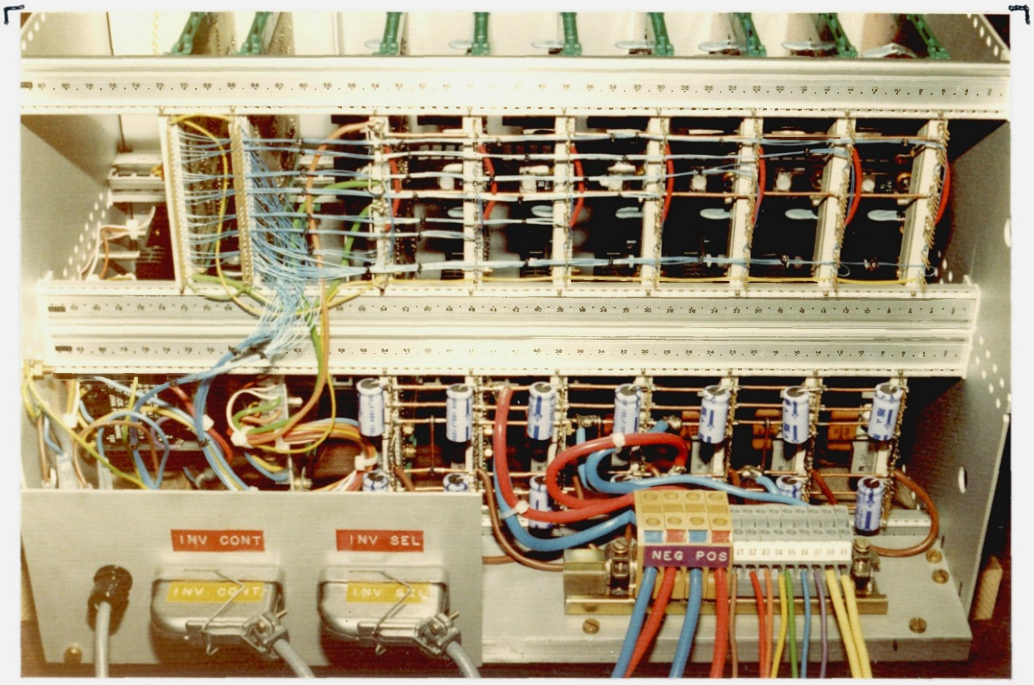


FIG. 5.27a. PHOTOGRAPH OF THE BACK OF THE HEXFET INVERTER EUROCARD RACK UNIT

The toggle switch to enable the 100 kHz oscillator board was mounted on the inverter front panel along with its associated "100 kHz enabled" LED. The LED's indicating the

status of the +5 volt and +15 volt power supplies and the mains switch for the supplies were also mounted on the front panel as shown by the photograph in fig. 5.27(b).

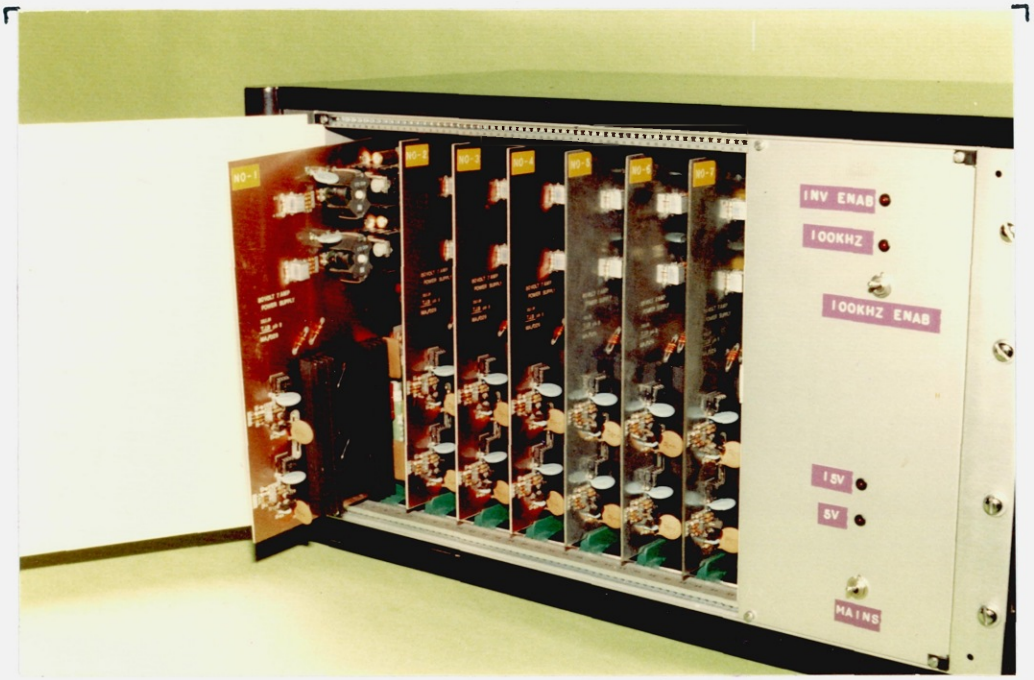
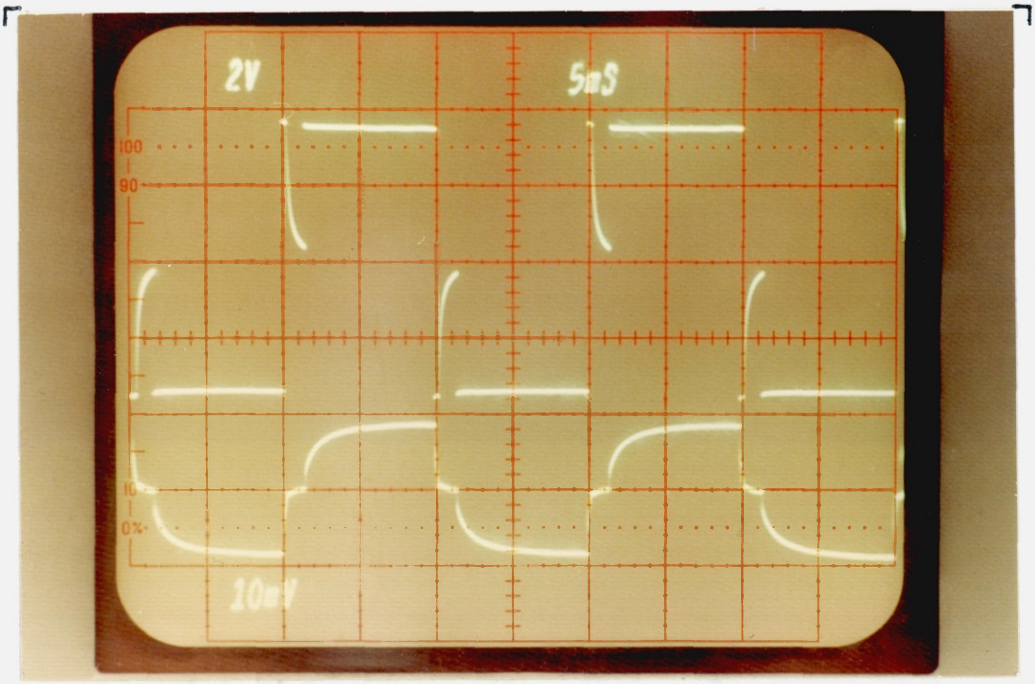


FIG. 5.27b. FRONT VIEW OF THE INVERTER UNIT SHOWING THE HEXFET CIRCUIT BOARDS, THE INDICATOR LEDS, AND CONTROL SWITCHES

When completed, the inverter was tested with various waveforms generated by the TMS 9900 microprocessor and no interaction between the phases could be detected. The inverter was connected up with a rheostat load for each phase. The inverter boards were checked individually by running them for fifteen minutes at ±60 volts with a phase current of 8 amp peak (6 amp rms) at 70 Hz. The TMS 9900 microprocessor was used to generate the inverter drive signals and the waveform shape was of the form required by the square wave motor. The heatsink temperature rose to 83° C by the end of each test. This is slightly higher than that predicted but the ambient temperature during the test was higher than 25° C and the flow of air around the heatsink was somewhat restricted. The rheostat inductance caused a freewheel period to occur when each hexfet switched off and a 5 volt overshoot occurred at this point on most phases when the maximum phase current was being carried. It is

thought that this overshoot is caused firstly by the freewheel current feeding back into the supply decoupling capacitors and secondly by voltage drop across the parasitic transistor as it conducts the freewheel current. Phase 6 had a 20 volt overshoot which oscillated at 6 MHz at the start of the negative freewheel period and decayed in 30 μs. As only one board exhibited this it was decided to ignore it and proceed with the checks. The inverter was run with 7 phases energised for a variety of supply voltages at frequencies from d.c. to 3400 Hz. Fig. 5.28(a) shows the typical voltage waveform for phase 1 and fig. 5.28(b) shows the turn-on time of the positive rail hexfet. The turn-on time of the hexfets was about 40 ns. The turn-on time is fast because the hexfet drain current is limited initially by the load inductance. Therefore as soon as the gate threshold voltage is exceeded, the hexfet can turn on very quickly and has a minimal voltage drop across it because it has very little drain current to support. The gate voltage then continues to rise and reaches the potential required to support the full drain current well before one time constant of the load circuit. The small amount of inductance present in most circuits therefore acts as a series current snubber and allows the hexfet to be fully conducting before the load current builds up.



(5ms/cm)

FIG. 5.28a. TYPICAL WAVEFORMS ON PHASE 1 (INVERTER SUPPLY ±35V): TOP TRACE – OUTPUT VOLTAGE 20V/cm; BOTTOM TRACE – OUTPUT CURRENT 2A/cm

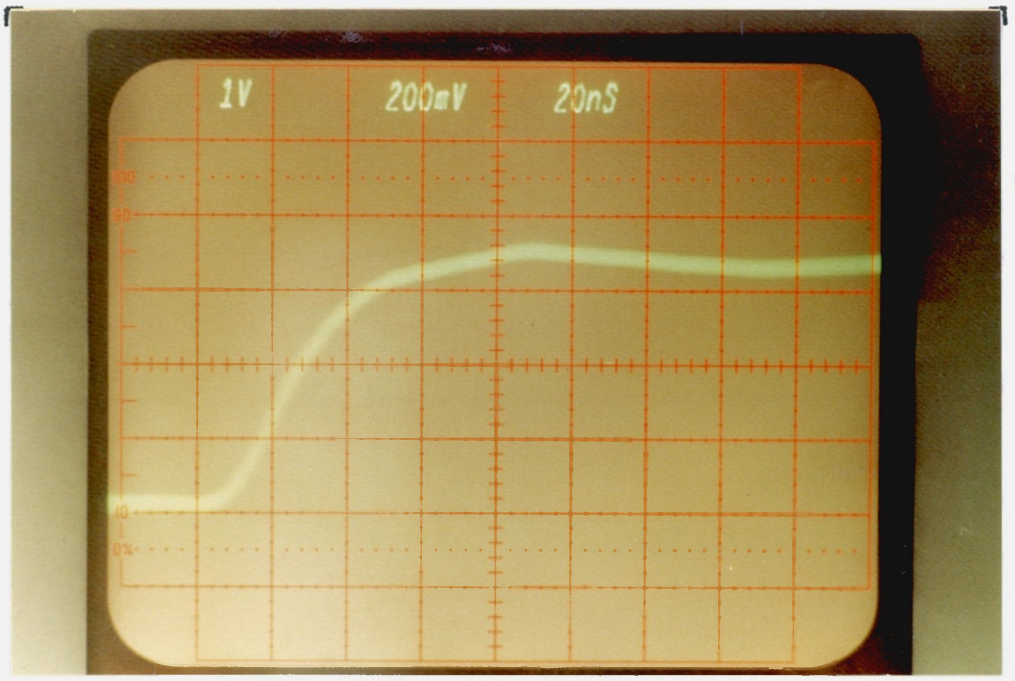


FIG. 5.28b. THE TURN ON TIME OF THE POSITIVE RAIL HEXFET - PHASE 1 VOLTAGE WAVEFORM : 10V/cm, 20ns/cm (INVERTER SUPPLY ±40 VOLTS, SWITCHING FREQUENCY 3450Hz)

The inverter was run at ±60 volts at 7 amps rms per phase for fifteen minutes with no problems. As 60 volts was the maximum supply voltage available it was decided that the inverter was suitable for use with the square wave motor system.

One hexfet was inadvertently destroyed during the tests. It was hoped that the individual drain current of a hexfet might be viewed on an oscilloscope and so a current probe was connected around a wire link inserted into the drain current. Unfortunately the insertion impedance of the current probe must have caused a large voltage spike on the drain because the hexfet and driver circuit were destroyed with only a 10 volt supply connected.

Details of the inverter efficiency and more waveforms are given in the system test results contained in Chapter 9.

5.6 The Variable D.C. Inverter Voltage Supplies

These supplies are required to provide steady regulated outputs in order that the inverter voltage waveforms match the motor waveforms. Crude phase controlled rectifier solutions were rejected immediately as being unsuitable. The inverter requires two microprocessor controllable d.c. supplies, one for the positive rail and one for the negative rail. The output amplitude needs to be variable reasonably quickly in order that the motor speed can be controlled. The voltage waveforms required by the 7 phases of the motor are such that at any time three phases are connected to the positive supply, three to the negative supply, and one is disconnected to allow commutation to occur. Under steady state conditions the difference between the applied square wave phase voltage and the motor square wave back-emf causes a steady current to flow through the winding resistance. Each phase draws a steady current from the supply to which it is connected, and the total current drawn from each supply rail is essentially constant for a particular load and three times larger than a phase current.

It was originally hoped to use a d.c. chopper circuit to produce each of the variable supply rails required, with a controlled output from 0 to 80 volts. A basic form of chopper is shown in fig. 5.29, and the output voltage depends directly on the mark/space ratio at which the switch "S" operates.

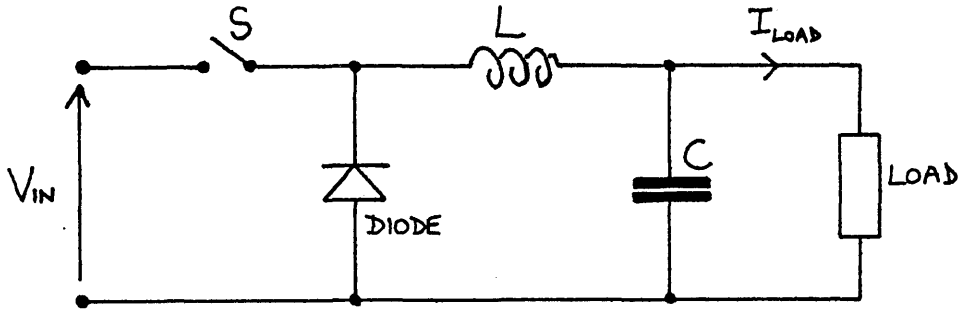


FIG. 5.29. A BASIC FORM OF VOLTAGE CHOPPER CIRCUIT

The switch can be a thyristor or a power transistor. The rating required for each chopper is 1.5 kW, as each supply rail supplies half of the inverter power requirement. This rating can easily be achieved with the semiconductors available today. The fact that the inverter draws a roughly constant current when supplying the square wave motor is a point in favour of using chopper supplies, since the inductor/capacitor smoothing network works best when the load current is constant. However, a chopper requires a d.c. supply to work from, and a 1.5 kW 80 volt d.c. supply is not usually readily available. To generate an 80 volt d.c. rail conveniently, it is possible to step down the mains voltage by transformer and rectify and smooth the output. However, 1.5 kW transformers are large and expensive and an alternative approach is to rectify and smooth the mains voltage directly to get 340 volts. This can then be chopped to get the required 0 to 80 volt output, but using a chopper to give such low variable output voltages from a high input voltage is not efficient, because the switching element only conducts for a relatively short time. The situation can be improved by using two choppers in series, one performing a fixed voltage reduction down to say 100 volts and the other then producing the variable 0 to 80 volt rail from the 100 volts. The finances for the project would have allowed some form of chopper to have been built, but time was very short and it was thought that for the sake of speed, a pair of d.c. machines might have to be used as the positive and negative supplies, with their fields controlled to vary the output voltages. This would not have been very satisfactory since the output voltage is very noisy due to the commutator, and the inductance of the machines must be capacitively decoupled. In addition, the response of a d.c. machine to a change in field current is not very good, and so the output voltage response would be far from desirable.

Fortunately, at this point in time a pair of Farnell "H series" 0-60 volt, 50 amp programmable power supplies were purchased for the laboratory. The output voltage of

each supply is proportional to a control current and a current of 2 mA is required for a 60 volt output. The 60 volt output would not be sufficient to achieve full speed operation of the 80 volt square wave motor. However, it was discovered that the airgap flux of the machine was rather lower than anticipated and full speed operation would only require about 40 volts from each supply. The 60 volt supplies were therefore incorporated into the system. The response of the output voltage is sufficiently fast to enable some speed control to be attempted if required.

The configuration connections in each supply were set up for programmable voltage operation as detailed in the equipment handbook. To enable the TMS 9900 microprocessor to control the supplies it was necessary to design and construct a suitable interface to allow a 16 bit word to generate two independent floating programming currents. The microprocessor outputs the command via the voltage control port ($C00E_{16}$) on the microprocessor interface board, as described in Chapter 4. Eight bits of the control word are used to control the magnitude of the control current and eight bits are used to enable and disable the current. Therefore, if the correct enable byte is not present, the supplies are held in a zero volts condition. The eight bit control byte has sufficient resolution to allow the voltage to be incremented reasonably accurately in one volt steps.

To prevent ground loops and other interaction between the microprocessor and the two supplies, it is sensible to optically-isolate the three units. This removes the chance of damage to the microprocessor in the event of either supply becoming faulty. It is far easier to optically isolate a digital signal than an analogue signal, and so a digital approach was used. The optically isolated voltage control circuit shown in fig. 5.30(a) contains the enable decode logic, the optical isolators and the digital to analogue (D/A) converters, which generate the required control currents.

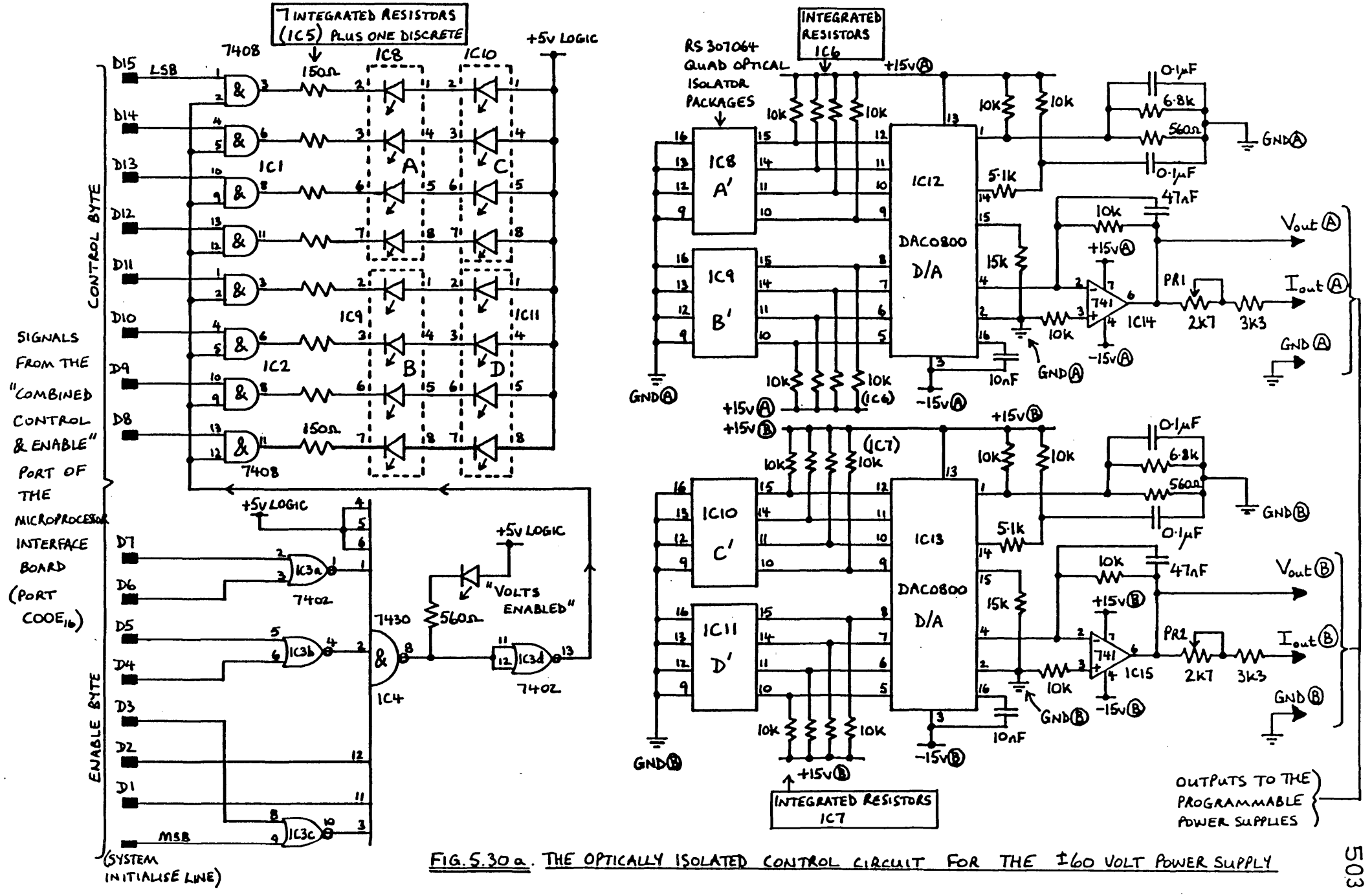


FIG. 5.30a. THE OPTICALLY ISOLATED CONTROL CIRCUIT FOR THE ±60 VOLT POWER SUPPLY

OUTPUTS TO 60 VOLT POWER SUPPLIES FROM AUXILIARY LOGIC UNIT BACK PANEL.		
	BANANA PLUG COLOUR	
	"A" PORT	"B" PORT
FLOATING GROUND	GREEN	BLACK
VOLTAGE OUTPUT	RED	WHITE
CURRENT OUTPUT	BLUE	YELLOW

SIGNALS FROM "COMBINED CONTROL & ENABLE PORT" OF MICROPROCESSOR (FED INTO AUX. LOGIC UNIT VIA D TYPE SOCKET) PIN NUMBERS	SIGNAL ON PARTICULAR PIN NUMBER	
1	"SYSTEM INITIALISE LINE"	
2	D1	
3	D2	
4	D3	VOLTS "ENABLE" BYTE
5	D4	
6	D5	
7	D6	
8	D7	
9	D8	
10	D9	
11	D10	
12	D11	VOLTS "SELECT" (ENABLE) BYTE
13	D12	
14	D13	
15	D14	
16	D15 (LSB)	
17	GROUND	
18 to 25	NO CONNECTION	

FIG. 5.30b CONNECTIONS FOR THE 60 VOLT POWER SUPPLIES ON THE BACK PANEL OF THE KEYPAD AUXILIARY LOGIC UNIT

The enable byte was chosen to be 60_{16}, mnemonically representing the 60 volt power supplies. The most significant bit of the enable byte is taken to logic 1 whenever the microprocessor is reset, and this ensures an immediate shut down of the power supply in a reset condition. When the correct enable byte is present, a LED is illuminated and the output of IC3d goes to logic 1. This enables the and gates (IC1 and IC2), thus allowing the control byte through to the optical isolators and so on to the D/A's. For incorrect enable bytes the outputs of the and gates are at logic 0, and the opto-isolator drive diodes are enabled causing all the output stages of the opto-isolators to pull down to zero volts. Hence the digital input to each D/A converter is zero, and so the analogue output voltages and currents generated by the 741 operational amplifiers (IC14 and IC15) are also zero. Thus if the interface circuit is not connected to the microprocessor, it forces the outputs of the power supplies to go to zero volts.

The output current from each D/A amplifier can be trimmed by the preset potentiometers PR1 and PR2. The 0.047 μF integrating capacitors connected across each 741 amplifier were needed to smooth the output current and voltage. This was necessary because the switching times of the various transistors in the opto-isolators differed greatly, causing the digital input to the D/A's to change in a non-monotonic fashion. The chosen capacitors have a time constant sufficient to remove the problem and the output can still generate a ramp at 100 Hz. The voltage outputs from the D/A amplifiers were required to control the power supplies of the magslip system when it was used to develop the microprocessor motor control software. The circuit was assembled on a eurocard with integrated packages of resistors used to save space.

Each D/A with its associated opto-isolators and amplifier, needs an independent floating ±15 volt power supply, and the circuit diagram of the twin regulated supplies is shown in fig. 5.31. The supplies were also built on a

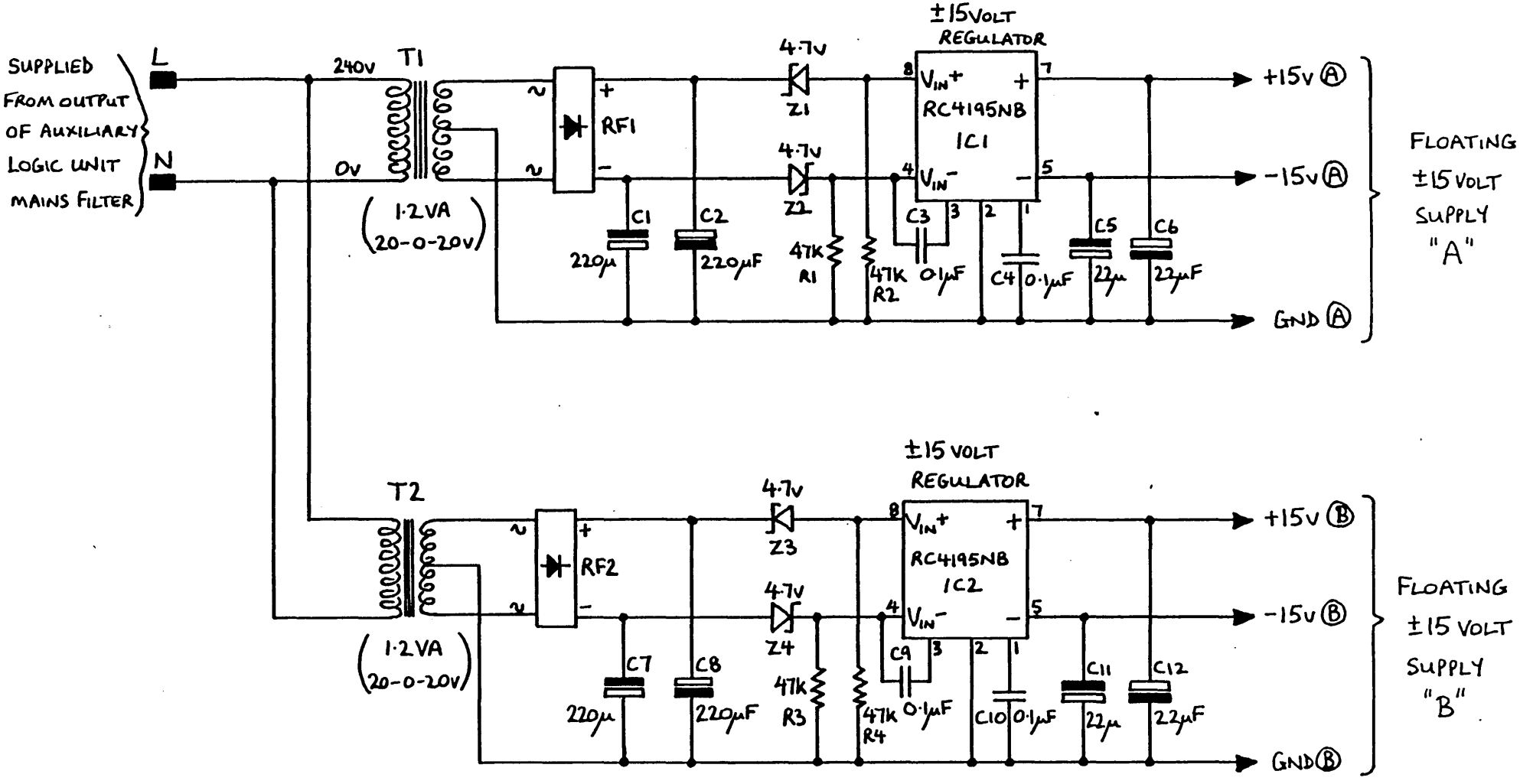
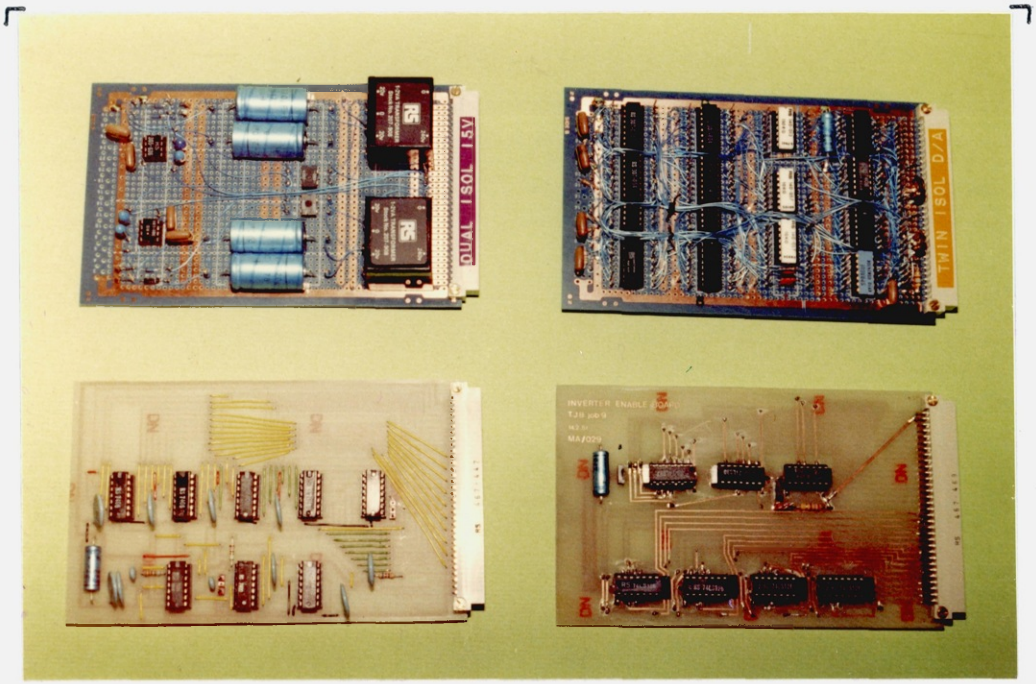


FIG. 5.31. CIRCUIT DIAGRAM OF THE TWIN ISOLATED ±15 VOLT 30mA SUPPLIES REQUIRED BY THE OPTICALLY ISOLATED CONTROL CIRCUIT SHOWN IN FIG. 5.30a

eurocard. Fig. 5.32 is a photograph showing the twin isolated ±15 volt power supplies and the optically isolated voltage control board. (The photograph also shows the inverter control logic and the 100 kHz clock logic printed circuit boards, described in sections 5.5.1 and 5.5.2.2 respectively.) The two boards were mounted in the keypad auxiliary logic unit as spare room was available in its eurocard rack. The interconnections to the microprocessor and 60 volt power supplies were made via a D type connector and banana plugs respectively and are summarised in fig. 5.30(b). The "volts enabled" LED was mounted on the left hand side of the front panel.



TOP LEFT : ±15 VOLT ISOLATED POWER SUPPLIES
TOP RIGHT : OPTICALLY ISOLATED CONTROL CIRCUIT FOR THE 60V PSU'S
BOTTOM LEFT : 100 kHz CLOCK LOGIC BOARD
BOTTOM RIGHT : THE INVERTER ENABLE CIRCUIT BOARD

FIG. 5.32. PHOTOGRAPH SHOWING THE CIRCUIT BOARDS ASSOCIATED WITH THE CONTROL OF THE MOSFET INVERTER AND 60 VOLT PSU'S

The operation of the power supplies under microprocessor control was checked with suitable software and they functioned perfectly. No problems were experienced with the interface circuitry. A set of flexible 42 amp power

leads were made up to connect the power supplies to the inverter.

5.7 Precautionary Current Limit Components

A main objective at the start of the project was to remove the need for current feedback in the inverter-motor system. The phase current in a motor is proportional to the difference between the applied phase voltage and the generated back-emf phase voltage, and so matching the two voltage waveforms should result in an essentially constant phase current in a square wave system, as shown in fig.5.33.

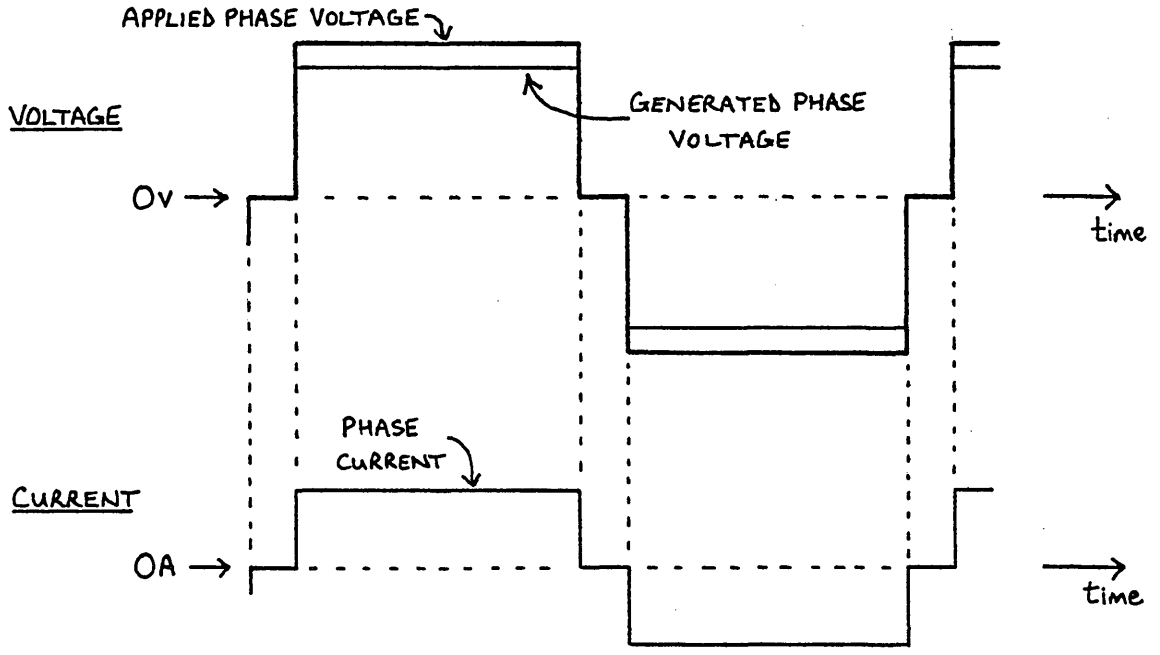


FIG 5.33. IDEAL PHASE WAVEFORMS IN A SQUARE WAVE MOTOR/INVERTER DRIVE SYSTEM

The idealised current waveform assumes that the phase winding inductance is very small, (as is so in the case of the square wave motor), therefore allowing fast rise and fall times for the current. If it were possible to achieve a waveform match as good as that shown in fig. 5.33, the likely phase current could be predicted simply from a know-

ledge of the motor speed and the applied phase voltage, since the generated back-emf is proportional to the motor speed. However, in practice the back-emf waveform does not exactly match the applied voltage waveform, and the typical waveforms are shown in fig. 5.34, where it can be seen that large current spikes occur.

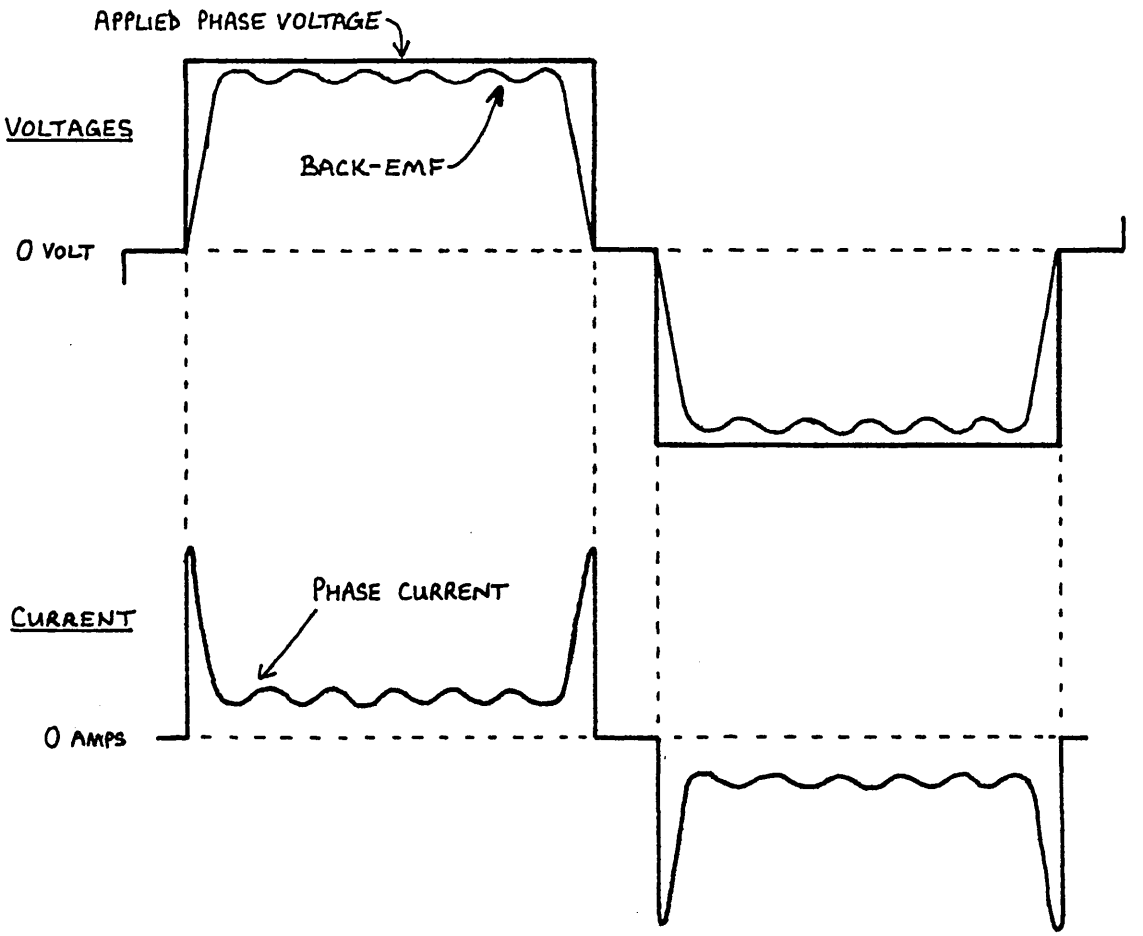


FIG. 5.34. TYPICAL VOLTAGE AND CURRENT WAVEFORMS IN A PRACTICAL SQUARE WAVE MOTOR/INVERTER DRIVE SYSTEM

The large spikes of phase current can only be limited by increasing the phase inductance. The large potential difference between the two voltage waveforms at the rise and fall points is then absorbed by the inductive reactance of the winding.

The phase inductance of the square wave motor is only

about 100 μH. It was decided to add an external inductance into each phase. From observations of the motor back-emf it was known that its rise time at the very worst was equal to a fourteenth of the time for one revolution, (i.e. it equals the time taken for the rotor to pass over one slot pitch). The back-emf constant was found to be about 1.233 volts per 1000 rpm. It was assumed that under no load conditions the resistive voltage drop in the winding would be small enough to be neglected. This is reasonable since the phase resistance is only 0.1Ω. Therefore, choosing an arbitrary no load speed of 6000 rpm, the applied phase voltage would have to be about 7.40 volts. At 6000 rpm the rise time of the back-emf is about 0.71 ms. If it is assumed that the back-emf effectively appears only at the end of its rise time, the total winding inductance has 7.40 volts across it for 0.71 ms. Under these conditions the inductance must limit the phase current to the rated current of the hexfet (i.e. 7 amps), and the calculated value of inductance is the absolute maximum required.

Using $$L_{ph} = (V_{ph})/(dI_{ph}/dt) \qquad 5.38$$

In this case:

$$\frac{dI_{ph}}{dt} = \left(\frac{7.0}{0.71 \times 10^{-3}}\right) \text{ A s}^{-1} \qquad 5.39$$

i.e. $$\frac{dI_{ph}}{dt} = 9860 \text{ A s}^{-1} \qquad 5.40$$

Therefore:

$$L_{ph} = \left(\frac{7.40}{9860}\right) = 0.75 \text{ mH} \qquad 5.41$$

Subtracting the actual phase inductance of 100 μH gives the required external inductance as 0.65 mH. As the required applied voltage is proportional to speed for the conditions assumed here, this method would give the same value no matter what speed was chosen. It was decided to use an

external inductor with an inductance of about 1 mH, as this removes any chance of current spikes.

The phase resistance of the square wave motor is, as mentioned earlier, only 0.1Ω. This small resistance is a problem because the smallest increment of phase voltage that can be selected via the microprocessor keypad is one volt. The phase inductance has no effect for the majority of the applied voltage on time because the phase current is constant. Therefore it is reasonable to say that a rise in phase voltage of one volt increases the potential difference across the phase resistance also by one volt, and so the phase current rises by 10 amps. This current causes the motor to accelerate and so the current level falls as the speed increases, but the hexfets could be damaged in the meantime. It was thought that the addition of an external resistance of 1Ω in each phase would be appropriate since this would limit the rise in current to 0.91 amps for each extra applied volt. Therefore, the chosen values of external inductance and resistance were 1 mH and 1Ω respectively.

The inductors were made using ferrite rings mounted on a 1 B.A. brass rod as shown in fig. 5.35.

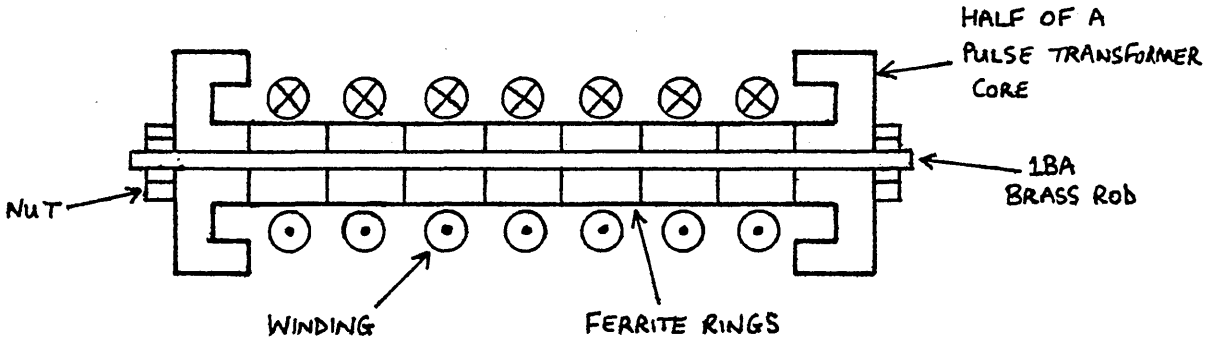


FIG. 5.35. SKETCH SHOWING THE CONSTRUCTION OF THE PRECAUTIONARY CURRENT LIMIT INDUCTORS

The end caps of the inductor were each made up of half a pulse transformer core. Sufficient turns were wound on to give approximately the required inductance. Two strands

of wire were used in parallel to give the winding a 7 amp rating. The windings were firmly secured with silicon rubber compound and insulation tape. The average measured inductance and resistance values of the 7 phase inductors were 1.25 mH and 86 mΩ respectively.

The 1Ω resistors require a power rating of at least 49 watts in order to be suitable for motor phase currents of 7 amps. Therefore, 50 watt 1Ω metal encased resistors were used and they were individually mounted on 2.1° C/W heatsinks. The total "external" resistance due to the 50W resistors and the 1.25 mH inductors was 1.086Ω per phase.

The seven inductors and seven resistors were mounted on a common baseplate eighteen inches square. No problems were experienced with this unit in operation and it is thought reasonable to regard these components as current limiting "motor snubbers" which allow the system to run without current feedback. The fact that they are passive devices makes them reliable and cheap. With a suitable motor redesign they could be incorporated within the motor winding. One advantage of keeping them outside the motor is that the heat generated is easier to dissipate. The resistances could actually be replaced by using high resistance, high dissipation MOSFETs, but this might reduce the reliability of the MOSFETs.

The addition of a series impedance might be regarded as an unsatisfactory solution but it is simple and cheap, just as many commercial systems are.

5.8 Current Level Detection

Although the aim of the project was to avoid the need for current detection in the system, some work was done to produce a circuit which would detect if a current exceeded a set limit.

There are several methods of current measurement and the transducers used include:

(a) resistive shunts including non-inductive coaxial types;

(b) hall effect devices;

(c) magneto sensitive resistors;

(d) current transformers.

The choice of transducer depends very much on the application and the frequency response required. Transducers in groups (b), (c), and (d) have the advantage that they do not need physical connection to the current carrying conductor. The resistive shunts of group (a) do need to be inserted into the current carrying circuit and hence introduce some power loss. However, they are simple, and providing such a device is connected into the circuit with one end connected to ground, it is easy to measure the voltage across it. If the shunt has to be placed so that both its ends are above ground potential, some form of differential amplifier is needed to sense the voltage drop. The differential amplifier requires a large common mode rejection ratio and a high voltage rating for its inputs. A system employing this approach has been built by Siemens (5.34).

An alternative method using group (d) of transducers has been developed by Mullard (5.35). The system uses ferrite toroids to implement a current transformer system in which d.c. currents can be sensed.

Whilst the solutions adopted by Siemens and Mullard are technically impressive, it was felt that they were more complex than the needs of this project warranted. As the need for current detection might only become apparent (if at all) during the running tests of the total system, it was decided that a minimum amount of time would be spent on

a current detector. Hence solutions employing transducers in groups (b), (c) and (d) were not considered. The simplest detector system employs a resistive shunt as the transducer. The shunt can be placed at any point in the circuit by arranging the differential amplifier to have a floating power supply. The circuit diagram is shown in fig. 5.36. A 100 kHz switching power supply (similar to those used in the inverter hexfet driver stages) provides a floating ±15 volt supply. The voltage across the 7 mΩ shunt R_s is amplified by the differential amplifier IC4. Positive peaks are fed via D3 and charge the "peak value" capacitor C5. Negative peaks are inverted by a unity gain amplifier IC5 and similarly charge C5 via D4. The smoothed peak values are compared by comparator IC6 with a trip value voltage set by a preset resistor PR1. When the current through the shunt is sufficient to cause the voltage on C5 to exceed the trip voltage, the comparator changes state, and its output is transmitted via an opto-isolator to a ground referenced transistor-buffer TR5. This provides a TTL level interrupt which can be used to signal a current overload to the microprocessor. The circuit functioned well and could detect current overloads on d.c. and a.c. currents. Seven such circuits were built (one for each phase of the inverter) with the 100 kHz clock shown in fig. 5.36 driving all seven floating power supplies.

The circuit shown in fig. 5.37 was used to combine the seven interrupts. If an interrupt occurs, the schmitt trigger oscillator IC2c is enabled and this produces a constant stream of interrupts which the microprocessor services in an appropriate manner until they disappear, at which point the current overload has been cleared. The microprocessor can determine which phase is overloaded by sampling the seven status lines. As an alternative to interrupt operation the microprocessor can sample the combined current status line whenever possible to determine if an overload is present.

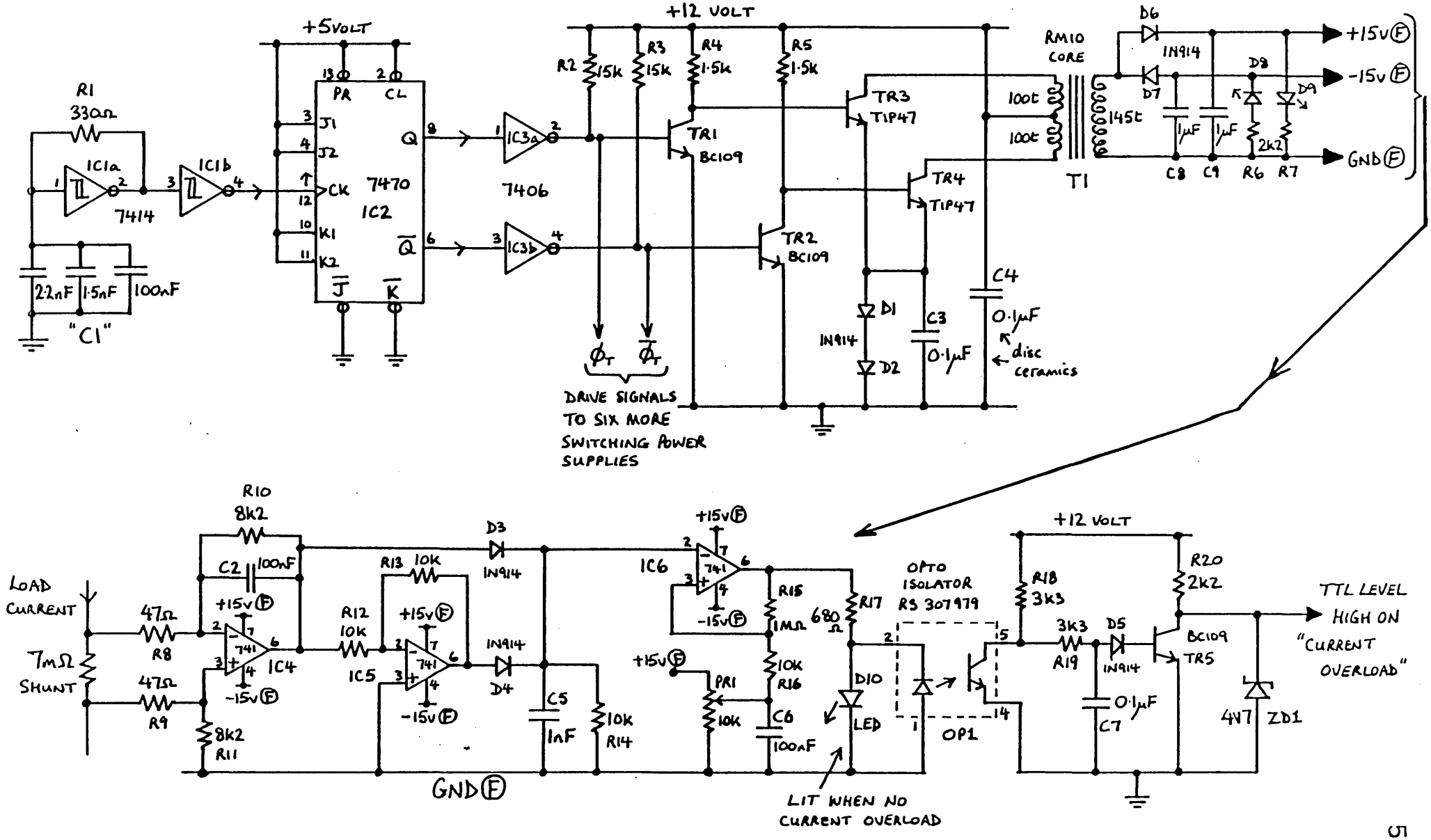


FIG. 5.36. CURRENT OVERLOAD DETECTION CIRCUIT

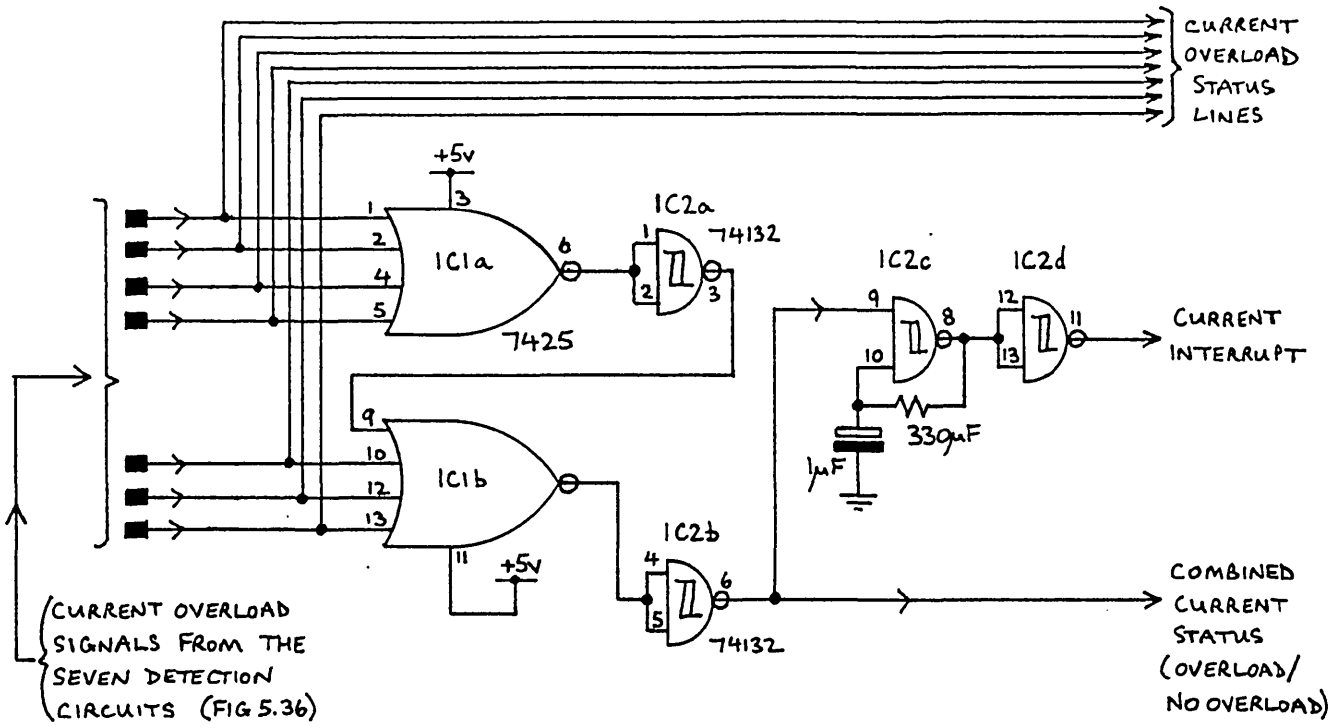


FIG. 5.37. CURRENT INTERRUPT GENERATION CIRCUITRY

It was not found necessary to employ the current detector circuits in practice, but their development is felt to have been worthwhile. A useful extension would be a modification to allow the microprocessor to change the trip voltage at present set by the preset resistor PR1. This could be achieved by means of opto-isolators and a D/A converter.

5.9 Conclusions

The work covered in this chapter has demonstrated that MOSFET transistors are relatively easy to incorporate into inverter designs. Their low power gate drive requirements reduce the amount of actual "power electronics" required in an inverter, because all the circuitry up to the MOSFET is standard low power electronics. The construction of each phase on a self contained printed circuit board has shown that inverters can be built with a modular plug in design, which in industrial applications leads to reduced

costs and easier servicing.

It is true to say that MOSFETs were used for the inverter because of their low power gate drive requirements, and the fact that they do not need critically designed snubber circuits. They were not really required as far as their fast switching speed is concerned but this was a useful bonus.

The microprocessor did limit the test frequencies possible to 3400 Hz, and in fact the 7 amp load tests were only conducted at frequencies up to 1300 Hz. However, bearing in mind that the inverter was designed specifically to drive the square wave motor, a frequency of 1300 Hz was well in excess of that required because it is equivalent to a motor speed of 78000 rpm.

The operating performance of the inverter in combination with the motor is discussed in the system results covered in Chapter 9.

CHAPTER 6

THE DESIGN AND CONSTRUCTION OF A "SQUARE WAVE" SYNCHRONOUS MOTOR AND ITS ASSOCIATED POSITION SENSOR

6.1 Introduction

Standard induction and synchronous motors have sinusoidally distributed windings in order to minimise their losses when they are run on the sinusoidal mains supply. If such motors are used in autopiloted drive systems they should therefore be fed from inverters producing sinusoidal waveforms. However, sinusoidal inverters tend to be relatively complicated and in Chapter 5 it is reasoned that the simplest form of inverter is a voltage source type producing a square waveform. Whilst the combination of a square wave inverter with a commonly available sinewave motor is simple, it does have the disadvantage that the losses in the motor are increased because of the harmonics in the supply waveform. In order to achieve a better match between inverter and motor and yet retain a simple square wave inverter it is possible to use a square wave motor. The iron losses in such a motor are likely to be somewhat increased because of the high order harmonics present in the supply waveform, but the extent of the increase can be minimised by careful design.

This chapter describes the design and construction of a square wave synchronous motor suitable for use with the MOSFET inverter described in Chapter 5. The position sensor needed for the detection of rotor position, to enable autopiloted operation of the motor, is also described.

6.2 Initial Motor Design Decisions

It was decided that a synchronous motor should be built in preference to an induction motor and the reasons for the choice are summarised in Chapter 1. Having chosen the type of motor, the next decisions were to choose:

(a) a maximum operating speed for the motor;

(b) the form of rotor excitation;

(c) the power output of the motor.

It is argued in Chapter 1 that the economic costs of auto-piloted motor systems can be more readily justified if a reduction in motor size is possible by improving the power to weight ratio. Therefore, it was decided that the motor speed should be at least as fast as the speed attained by commonly available domestic commutator motors. Commutator motors are used widely in vacuum cleaners, washing machines, food mixers and electric drills, and they operate reliably at speeds up to about 16000 rpm. The armature winding is on the rotating part of the motor and even though the machines are mass produced as cheaply as possible, they rarely disintegrate at speed. Therefore, it was decided that it should be possible to design a synchronous motor to operate at speeds up to twice those achieved by high speed domestic commutator motors. A top speed of 30000 rpm was selected as a suitable design target.

The choice of such a high speed then led to the decision that a brushless form of rotor excitation should be used. The life and reliability of brushes and slip rings are significantly reduced by high speed operation and so they must be eliminated. The simplest method of achieving rotor excitation without brushes is to mount permanent magnets (PM) on the rotor. This does mean that no control of the motor excitation is possible but the simplicity of the method makes it extremely attractive. The lack of

excitation control normally means that the motor power factor cannot be controlled when its speed and load vary, but the minimisation of stator phase inductance virtually enables a unity power factor to be achieved as is explained later in this chapter. The choice of PM excitation does depend on a suitable construction being designed to prevent the rotor bursting at the desired top speed of 30000 rpm, but this was not thought to be a problem.

The power output of the machine was not a quantity that could be specified initially. The power output of a motor depends on several factors including the rotor flux density, the rotor speed, and the total stator current. The rotor flux density depends on the chosen PM material, the air gap of the machine, and the magnetic circuit configuration. The total stator current is limited by the physical size of the stator, which in turn depends on the diameter of the rotor. The only known quantity was the desired rotor speed, and so it was decided that the rotor diameter would be chosen to allow safe operation at that speed and then the dimensions of the stator would be fixed accordingly. Once the dimensions of a motor are fixed, an estimate of its power output can be made, and the required rating of the supply inverter is then known. It was thought that, providing the inverter rating could be achieved with a reasonable number of MOSFET transistors, this would be a satisfactory design procedure. A redesign would only be necessary if the inverter rating was impractical.

However, before proceeding with the motor design it is necessary to outline the basic construction and operation of a square wave synchronous motor.

6.3 The Basic Requirements For and the Operation Of a Square Wave Synchronous Motor

The general form of voltage waveform produced by a phase of a voltage source square wave inverter is shown in

fig. 6.1.

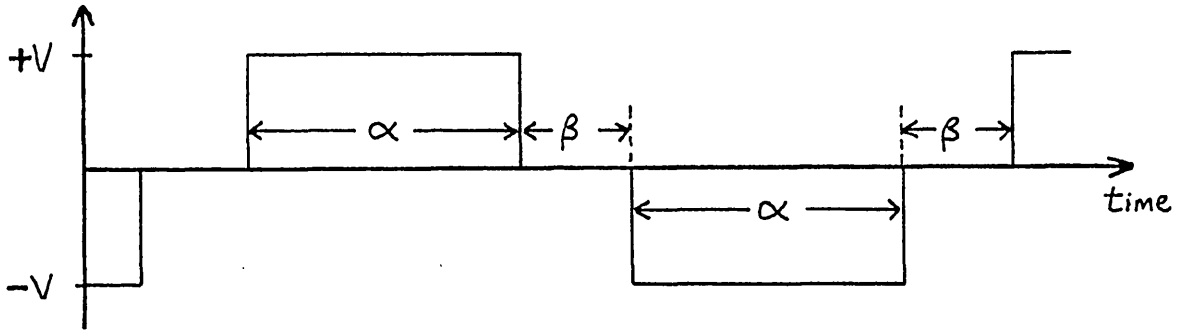


FIG. 6.1. GENERAL FORM OF OUTPUT PRODUCED BY A SQUARE WAVE VOLTAGE SOURCE INVERTER

The output is alternately positive or negative for a period α and changes from positive to negative or vice versa via a zero volts state of duration β. In the limit β is zero, and the output is then alternately positive and negative for a period equal to one half of the cycle time.

In order to match a synchronous motor to such a supply waveform, it is necessary to design the motor so that the back-emf of each phase winding has a shape as close as possible to that of the supply. The resulting phase current then depends on the relative magnitudes of resistance and inductance in the phase winding. Typically a winding can be resistance dominated, inductance dominated, or have a reasonable amount of both, and the resulting idealised phase currents are shown in fig. 6.2. It can be seen that in the case of a resistance dominated winding, the phase current is constant during the on periods and is in phase with the applied voltage. The constant square phase current is useful because it can result in a greater torque output from the motor, and the unity power factor that the winding has allows the supply inverter VA to be minimised. In practice it is not possible to eliminate the inductance of a phase winding completely because some current driven flux is inevitable. However, the inductance can be mini-

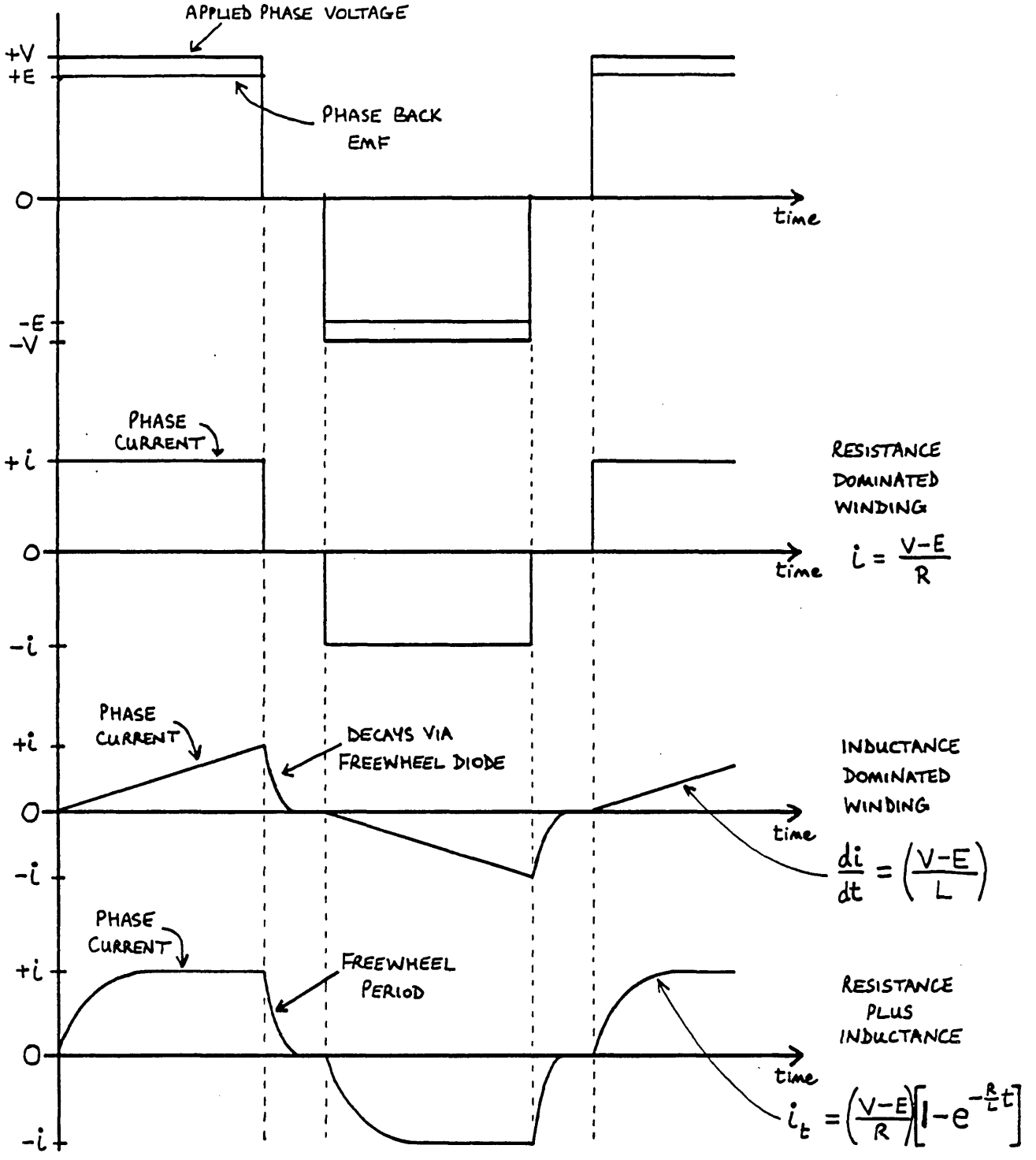


FIG. 6.2. POSSIBLE IDEALISED PHASE CURRENTS THAT CAN OCCUR IN A SQUARE WAVE SYNCHRONOUS MOTOR SYSTEM

mised to keep the current rise time as short as possible, and a power factor close to unity is then possible over a wide operating frequency range.

A quasi-square back-emf of the form shown in fig. 6.1 can be generated in a phase winding by arranging the rate of change of flux linkage to be constant (and of appropriate sign) when a back-emf is required, and to be zero during the desired zero volt back-emf periods. (i.e. The flux linkage is constant during these times.) This can be achieved by using a rotor which has a constant flux density "B" across each pole face (i.e. a square wave of flux across each pole arc) in combination with an unskewed, concentrated phase winding (one slot per pole per phase). The necessary flux density distribution across each rotor pole pair is shown in fig. 6.3(a) and a cross section of a 2 pole "square wave" rotor is shown in fig. 6.3(b). When the rotor is turning at constant speed the emf generated in a single turn of a phase winding is constant during the time that a pole face passes underneath the coil. If the conductors C_1 and C_2 (shown in fig. 6.3(b)) form the sides of a single turn phase coil, the total induced voltage E in the coil is given by:

$$E = 2.B.v.l \qquad 6.1$$

where v is the velocity of the conductors with respect to the airgap flux density, and l is the length of each conductor.

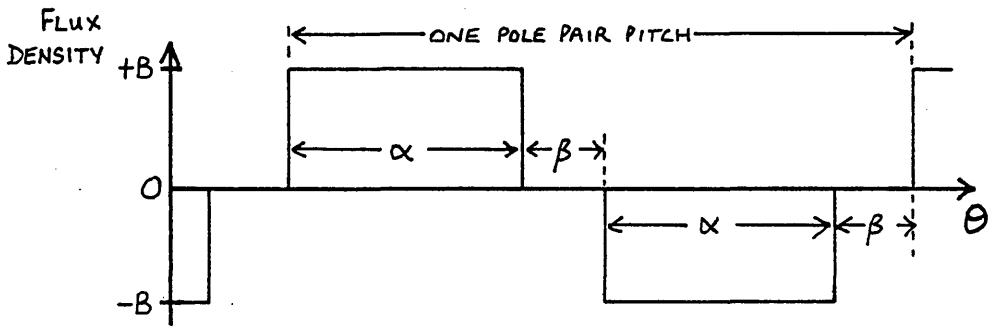


FIG. 6.3a. THE NECESSARY FLUX DENSITY DISTRIBUTION ACROSS EACH ROTOR POLE PAIR IN A SQUARE WAVE SYNCHRONOUS MOTOR

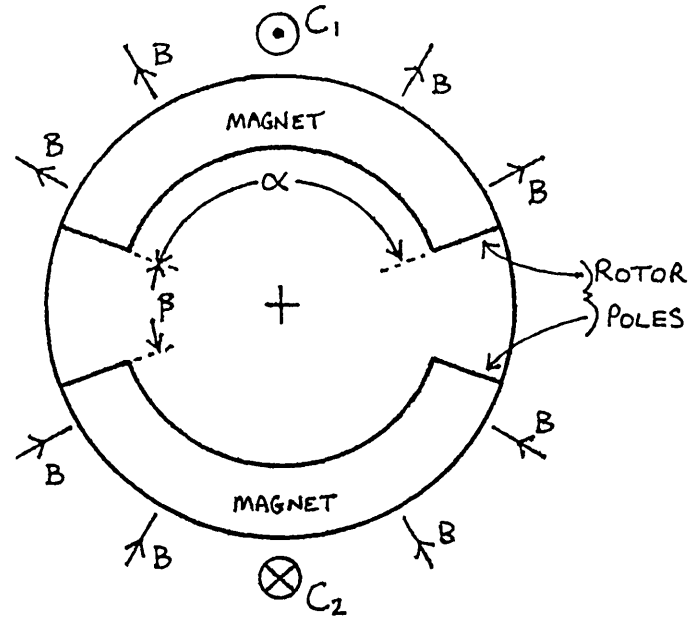


FIG. 6.3b CROSS SECTION OF A TWO POLE "SQUARE WAVE" ROTOR

A phase winding is usually made up of many individual turns and the way in which the individual coil emfs add together affects the generated voltage waveform. If the winding is distributed rather than concentrated, or the stator is skewed, there is a tendency for the resultant generated emf to be sinusoidal. This is because various parts of the winding meet the leading and trailing edge of the pole flux at different times, and so the vector sum of the individual emfs is by no means a constant with time. The fact that the stator must not be skewed is unfortunate because it may result in a rather large cogging torque for the machine. One way of reducing this is by the use of closed stator slots. However, it is not possible to have closed stator slots and still achieve a low stator inductance, (as is explained in section 6.6.2 dealing with the stator slot design). Another way of minimising cogging torque is to carefully choose the relative widths of slot opening, rotor pole arc, and slot pitch. This choice is made difficult by fringing phenomena and since both experimentation and field plotting can be lengthy procedures, this aspect of the motor design was left uninvestigated. Therefore, in order to achieve a square wave back-emf and a power factor close to unity, it was necessary to accept

a large cogging torque caused by unskewed, unclosed stator slots.

Having explained the basic requirements of a square wave synchronous motor, the mode of operation can be described. A 3 phase 2 pole motor is shown in fig. 6.4.

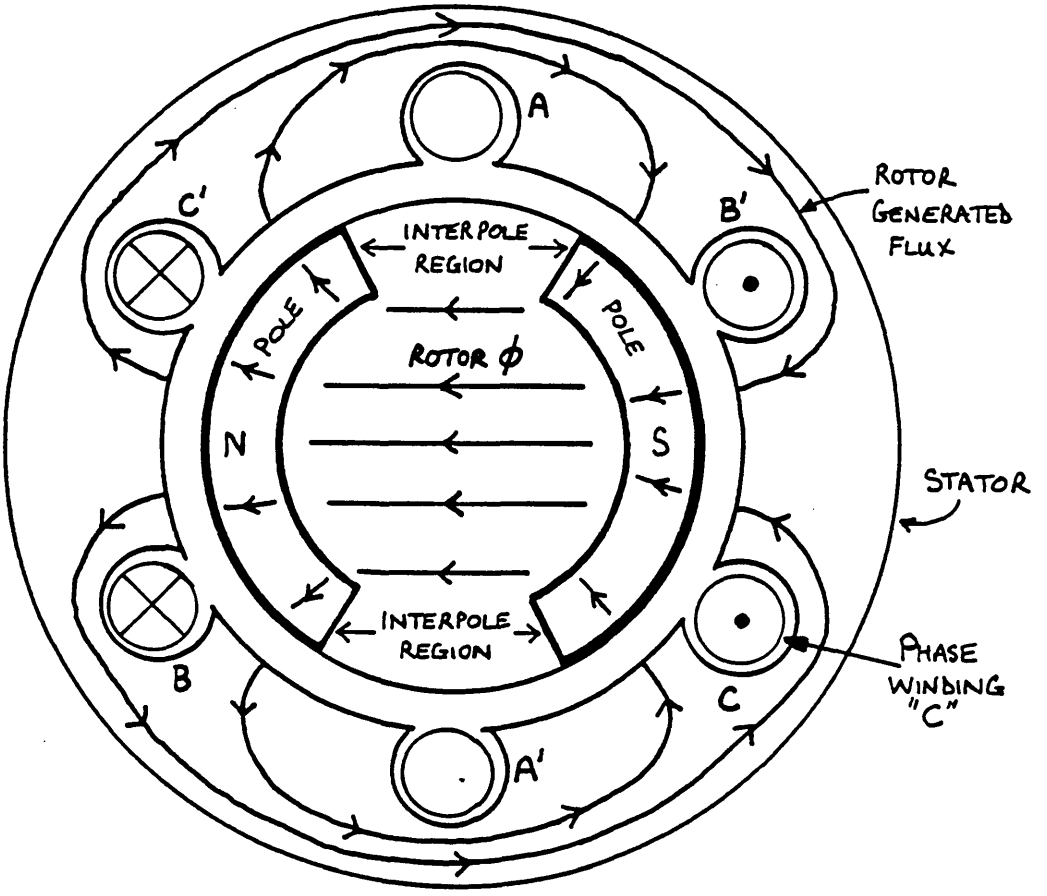


FIG. 6.4. CROSS SECTION OF A 3 PHASE 2 POLE SQUARE WAVE MOTOR

It can be seen in fig.6.4 that the pole arcs do not cover 180 mechanical degrees. The interpole regions are necessary to avoid excessive pole to pole leakage flux. The interpole regions are useful in a further way. If it is assumed that the flux density in the interpole regions is zero, then the induced emf in a phase coil facing the interpole areas is also zero. This is a helpful feature since during the zero emf period the inverter can more easily switch off and reverse the coil current.

In fig. 6.4 (illustrating a 3 phase design) it can be seen that phase A is commutating and phases B and C are carrying currents which can react with the rotor flux to produce torque. At any instant two phases are conducting and one commutating. If the fact that the phase current can be reversed is taken into account, there are six combinations of phases conducting and commutating and so the rotor rotates in six steps. Each time that a phase coil moves out from under the rotor poles, the relevant phase is switched off (via position sensing circuitry) and allowed to commutate. At the same time the next phase coil, just coming under the rotor poles, is switched on in order to maintain two conducting phases under the rotor poles at any time.

The pole arc can be greater than 120^{o} for a 3 phase 2 pole motor. However, in general, for an n phase machine (where n is an ODD integer) it is useful to make the pole arc span an angle θ_p given by:

$$\theta_p = \left(\frac{n-1}{n}\right).\pi \quad \text{radians} \qquad 6.2$$

If the pole arc is set according to equation 6.2 then:

(i) the n^{th} harmonic is eliminated with a consequent reduction in losses and motor heating. No benefit is obtained if n is made EVEN because the quasi square waveform shown in fig. 6.1 contains only ODD harmonics. (i.e. There are no EVEN harmonics to be removed.)

(ii) The generated phase emf and the required matching applied voltage waveform have a duration that is a simple sub-multiple of the rotor rotation time. As there are 2n steps per revolution the position sensor must have a minimum resolution of π/n radians. If the pole arc exceeds the minimum length there are more than 2n steps per revolution and in order to utilise the flux fully, the switchings are not necessarily equally spaced; hence the

need for a more complicated position sensor. For a 3 phase 2 pole machine with a pole arc of 180° there are 12 steps per revolution which are equally spaced.

(iii) There are always (n-1) phases conducting at any time. (As n tends to infinity, the winding is virtually fully utilised.) As in (i), this only applies when n is ODD. If n is EVEN there are two phases off at any time and the phase windings are therefore utilised for a smaller percentage of the total time. Since each phase ideally draws an essentially constant current when on and there are (n-1) phases conducting, the total supply current is virtually constant for a given load condition. In addition, as each phase is displaced from the adjacent phases by $2\pi/n$ radians, their generated emfs are of opposite polarity. Therefore, at any time, (n-1)/2 phases need connecting so that they have positive volts applied to them and (n-1)/2 need negative volts applied to them. If each phase winding is driven by a half bridge inverter stage as described in Chapter 5, the positive and negative voltage supply rails are equally loaded at all times. This fact is demonstrated in fig. 6.5 where the back-emf and required applied voltage for each phase of a 3 phase motor during one rotation are summarised.

	PHASE A		PHASE B		PHASE C		POSITIVE VOLTAGE SUPPLY		NEGATIVE VOLTAGE SUPPLY	
	BACK EMF	APPLIED VOLTS	BACK EMF	APPLIED VOLTS	BACK EMF	APPLIED VOLTS	NUMBER OF PHASES ON	TOTAL CURRENT DRAWN	NUMBER OF PHASES ON	TOTAL CURRENT DRAWN
STEP 1	+E	+V	-E	-V	O	O	1	+I	1	-I
STEP 2	O	O	-E	-V	+E	+V	1	+I	1	-I
STEP 3	-E	-V	O	O	+E	+V	1	+I	1	-I
STEP 4	-E	-V	+E	+V	O	O	1	+I	1	-I
STEP 5	O	O	+E	+V	-E	-V	1	+I	1	-I
STEP 6	+E	+V	O	O	-E	-V	1	+I	1	-I

FIG. 6.5. SUMMARY OF THE VOLTAGES AND CURRENTS IN A 3 PHASE SQUARE WAVE MOTOR DURING ONE REVOLUTION

The constant supply current from each supply rail is useful since it facilitates the use of a chopper to regulate the supply rail voltages.

It is thought that points (i), (ii) and (iii) are sufficiently beneficial to warrant the use of equation 6.2 in selecting the motor pole arc. With the pole arc so selected, an autopiloted synchronous motor runs in a manner similar to a commutator motor. The motor has an "electronic commutator" in the form of an inverter and position sensor. The equivalent number of bars on the "commutator" are 2n and there are (2n-2) brushes which ensure that (n-1) phases are connected at any time. However, this arrangement causes each phase to be connected in parallel across the voltage supply, whereas in a d.c. commutator motor the armature coils are connected in series and there are fewer brushes. The parallel arrangement ensures that each on phase has the same voltage applied across it to match the generated back-emf and so the phase currents are essentially balanced. The current in each phase is voltage forced and so it is important to minimise the phase inductance to allow rapid current rise times.

The final point needing discussion is the number of phases actually required for a successful square wave motor. The phase number should be odd, for the reasons explained earlier, and so it might be thought that a 3 phase machine would be a good choice. However, because of the need for concentrated coils it is advantageous to increase the number of phases. For a n phase machine, each phase has to deal with 1/n of the total power. The number of turns per phase needed to achieve the power rating is relatively large if the phase number n is small. This leads to a requirement for large slots to accomodate each phase and this is not desirable because:

(a) the dissipation of heat from the middle of a coil is poor in a large slot;

(b) the large slot size reduces the amount of backing iron in the stator for a given frame size and so saturation can occur behind the slots;

(c) slot leakage is high with a small number of slots.

By increasing the number of phases it is possible to reduce the slot dimensions and spread the power loss dissipation around much more of the stator periphery. As the number of phases becomes very large the situation tends towards that shown in fig. 6.6 in which a square wave of rotor flux interacts with a square wave of stator current.

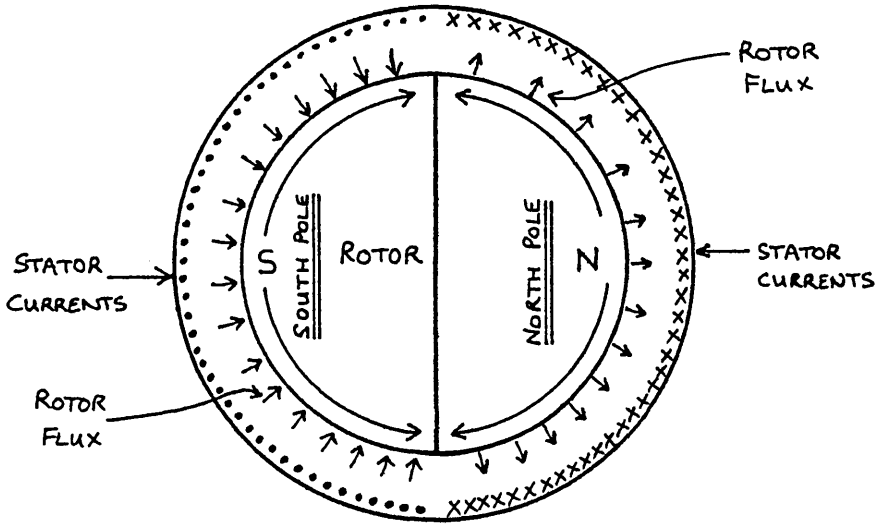


FIG. 6.6. SITUATION IN A MOTOR WITH A VERY LARGE NUMBER OF PHASES: SQUARE WAVE OF ROTOR FLUX INTERACTS WITH SQUARE WAVE STATOR CURRENT

The torque produced by the machine is then greater than that available from an equivalent phase number sinusoidal machine. The calculation of the torque is also very simple. With a very large phase number the phase windings are energised almost continuously in a positive or negative manner, and the winding utilisation is therefore very good. At the same time the phases in the inverter are almost continuously in a conducting state and so the utilisation of the switching elements is good.

Having explained the basic operation of a square wave synchronous motor it is now possible to outline the design

procedure used for the motor described in this chapter.

6.4 Choice of the Number of Phases

For the reasons discussed in the previous section it was known that a large phase number was desirable and it should be odd. In addition, a paper by Jahns (6.1) has considered the improved reliability that can be achieved by the use of a high phase number. However, there are practical limits to the number of phases that can be used and they are:

(i) the number of connections that are reasonably acceptable between the motor and the drive inverter;

(ii) the number of switching elements and drive circuits that are needed in the inverter;

(iii) the number of control signals needed for the inverter.

Limit (i) is not a serious problem today, given the acceptance of multi-phase stepper motors, and it can be overcome by mounting the inverter directly on the motor end plate. Limit (ii) is very relevant because the cost of switching elements forms a significant portion of the inverter/motor price. However, the required rating of each switch falls as the number of phases is increased, and this can allow the use of relatively cheaper low power devices. Limit (iii) can be a problem if a microprocessor is used to provide the inverter drive signals. In general, two bits are required to control each phase completely because it can exist in three states, (OFF, ON positive or ON negative). Therefore, with microprocessor systems giving fully independant control of each phase, an n phase inverter requires a control word of 2n bits. Most microprocessors available today have 8 (1 byte) or 16 bit (2 bytes) data buses, and

can therefore control up to 4 or 8 phases respectively with single data transfers. An 8 bit microprocessor can obviously control more than 4 phases but the information must then be fed out in several bytes with a resulting limitation on the maximum frequency at which the inverter and motor can operate. It was thought best to use a 16 bit machine to enable up to 8 phases to be controlled with a single data transfer. It was originally intended to build a 9 phase motor so that some comparison could be made with the 9 phase square wave induction motor built by Professor Enslin of Cape Town University (6.2), but the desire to use a microprocessor as the controlling device and the cost of the MOSFET transistors used in the inverter made the use of more than 8 phases unattractive. The largest odd number of phases that can be used is then 7, and so this was the phase number chosen.

The idealised back-emfs for a 7 phase motor during one cycle are shown in fig. 6.7. If Fourier analysis is performed on a typical waveform shown in fig. 6.7, it is found that it can be represented by the expression:

$$f(t) = 1.24E \sin(t) + 0.33E \sin(3t) + 0.11E \sin(5t)$$

$$- 0.06E \sin(9t) - 0.09E \sin(11t) - .... \qquad 6.3$$

There is no seventh harmonic present because the pole arc span is set by equation 6.2. Each phase winding carries current for (6/7) of the total cycle time and so the winding utilisation is very good considering that only 7 phases are used.

Having decided on a 7 phase motor it was possible to design a suitable rotor, and this is described in the following section.

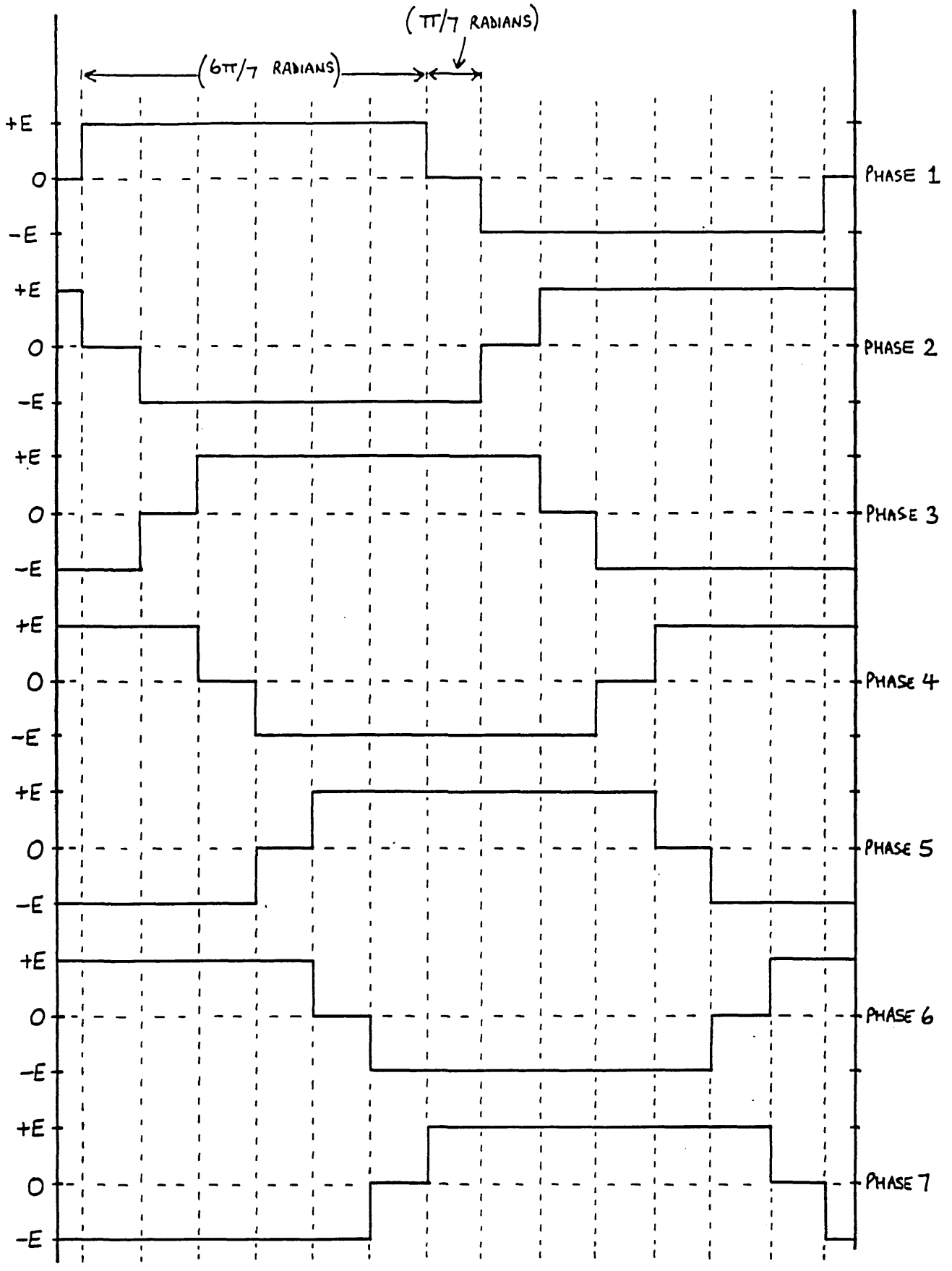


FIG. 6.7. IDEALISED BACK-EMFS FOR A SEVEN PHASE SQUARE WAVE SYNCHRONOUS MOTOR DURING ONE CYCLE

6.5 The Design of the Rotor

The first stage in the design of the PM rotor was to choose a suitable diameter for it. The size is constrained by:

(a) the centrifugal forces generated at the maximum speed of 30000 rpm;

(b) the minimum width of tooth acceptable in the stator lamination;

(c) the size of PM magnets available.

Observation of the rotor diameters of existing high speed machines suggested that a rotor diameter less than 75 mm (3") was required to limit centrifugal forces to mechanically tolerable levels. The minimum rotor diameter was also fixed by mechanical constraints and constructional considerations. (A very small diameter rotor is relatively difficult to assemble.) A 7 phase stator with concentrated phase windings has 14 slots. If a 50 mm diameter rotor is used with a small airgap between it and the stator, the internal circumference of the stator is approximately 160mm. The slot pitch is then 11.4mm. This is probably a reasonable minimum slot pitch to consider because a smaller slot pitch results in very thin stator teeth, which are mechanically fragile and likely to saturate. It was therefore decided to fix the rotor diameter somewhere between 50 and 70mm.

The next decision that had to be made about the PM rotor was whether it should be a 2 pole or 4 pole construction. In general it can be stated that:

(a) a two pole rotor is simpler to construct than a four pole rotor and can be more robust;

(b) with a 7 phase winding having one slot per pole

per phase, twice as many slots are needed with a 4 pole design. The tooth width would therefore be halved;

(c) the end winding in a four pole machine is shorter than in a two pole machine, resulting in a significant reduction in copper wire and losses;

(d) some flux density multiplication is possible with a four pole rotor if the magnets are suitably configured;

(e) the fundamental frequency in a four pole machine is double that of a two pole machine for the same rotor speed. Hence the iron losses in a two pole stator should be significantly less than those in a four pole stator;

(f) the flux in the stator backing iron of a four pole design is half that existing in an equivalent two pole design, and this can result in reduced losses.

The phase frequency required for a 2 pole motor to spin at 30000 rpm is 500 Hz, whereas a 4 pole motor requires 1 kHz. On balance it seemed best to choose a 2 pole rotor design.

6.5.1 Rotor Magnet Configurations

There are several configurations that can be used to construct a 2 pole rotor with a pole arc of $(6\pi/7)$ radians and a diameter in the required range. They include the use of:

(a) arcuate magnet segments as shown in fig. 6.8;

(b) standard bar magnets arranged in an arc as shown in fig. 6.9;

(c) shaped bar magnets arranged in an arc as shown in fig. 6.10;

(d) ring magnets as shown in fig. 6.11;

(e) shaped magnet blocks as shown in fig. 6.12;

(f) magnet slabs with shaped soft iron pole pieces arranged with or without a central shaft, as shown in figures 6.13 and 6.14 respectively.

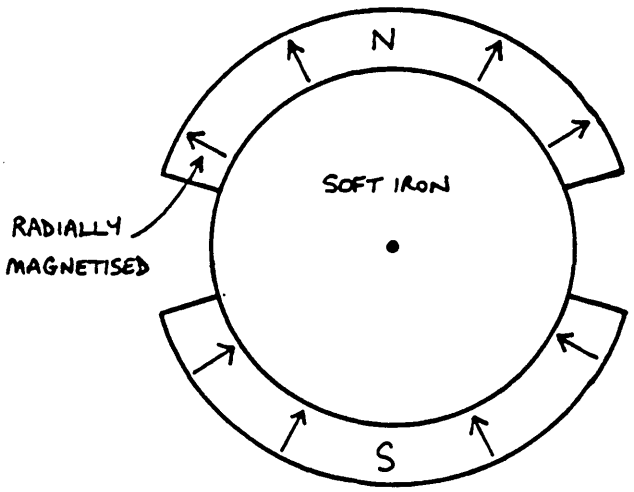


FIG. 6.8. TWO POLE ROTOR CONSTRUCTED FROM ARCUATE MAGNET SEGMENTS.

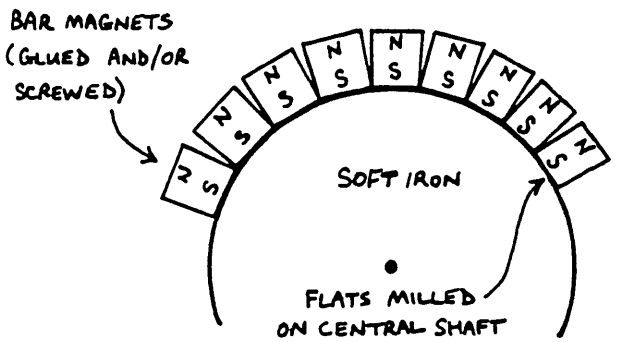


FIG. 6.9. TWO POLE ROTOR CONSTRUCTED FROM STANDARD BAR MAGNETS ARRANGED IN AN ARC.

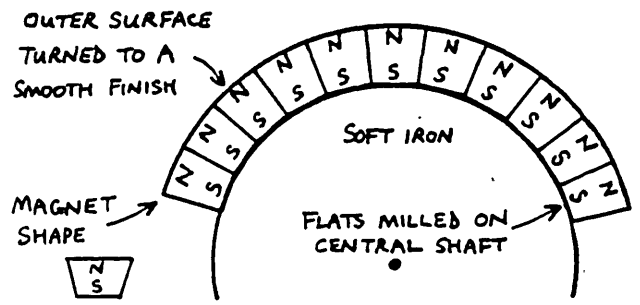


FIG. 6.10. TWO POLE ROTOR CONSTRUCTED FROM SHAPED BAR MAGNETS ARRANGED IN AN ARC.

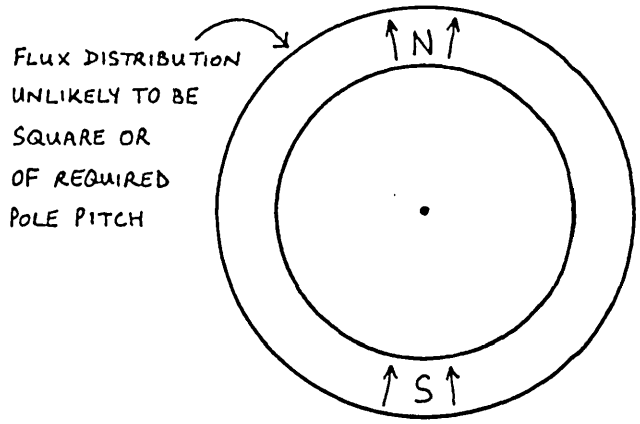


FIG. 6.11. TWO POLE ROTOR CONSTRUCTED FROM A SUITABLE DIAMETER RING MAGNET.

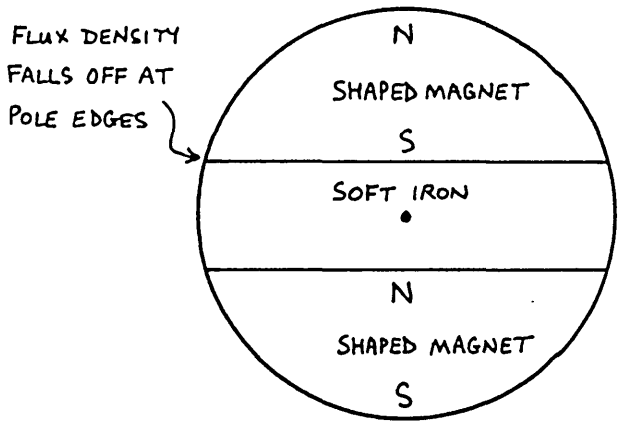


FIG. 6.12. TWO POLE ROTOR CONSTRUCTED FROM SHAPED MAGNET BLOCKS.

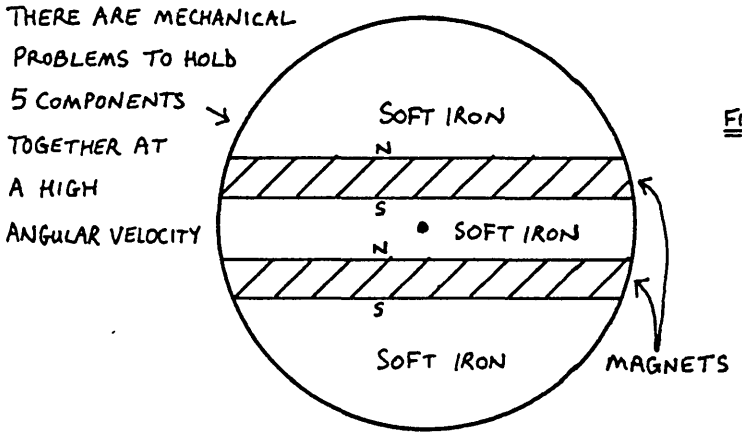


FIG. 6.13. TWO POLE ROTOR CONSTRUCTED FROM MAGNET SLABS WITH SHAPED IRON POLE PIECES. (ROTOR SHAFT PASSES BETWEEN MAGNET SLABS)

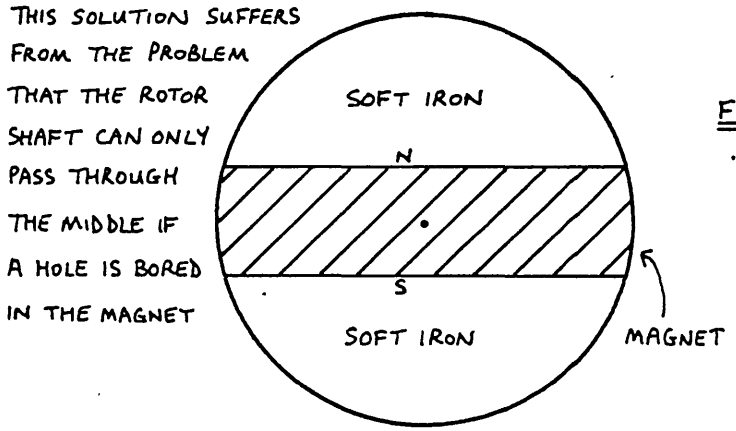


FIG. 6.14. TWO POLE ROTOR CONSTRUCTED FROM ONE MAGNET SLAB WITH SHAPED IRON POLE PIECES (NO PROVISION FOR ROTOR SHAFT TO PASS THROUGH MAGNET BLOCK)

The solutions offering the best chances of achieving a smooth square wave of flux over the required pole arc are (a) and (c). In these solutions the magnets form the pole faces of the rotor. Good magnetic materials have permeabilities close to that of air, and so with these configurations the stator effectively sees an airgap with a thickness equal to the magnet plus the physical airgap. This results in a reduction in the stator flux and so the rotor flux suffers less distortion and the phase windings have a low inductance as required. A variety of magnet materials can be used in these configurations and the four considered were:

(a) metallic magnets (Alcomax III);

(b) ceramic ferrite magnets (Magnadur);

(c) flexible rubber bonded ferrite magnets (Plastiform or Ferriflex);

(d) rare earth magnets (Samarium Cobalt-sintered or polymer bonded).

These four types were chosen because they were relatively easy to obtain commercially. Their typical demagnetisation characteristics are shown in fig. 6.15.

As the magnets are positioned directly next to the airgap and the stator windings, they must be able to withstand the full demagnetising effect of the stator current. Alcomax magnets do produce good remanent flux densities (B_r) in excess of 1T but have low coercive forces (H_c) of about 50kA, and so they can be easily demagnetised. In addition they are not readily available in anything but very standard shapes and care is required to machine them. As they are conductive, eddy currents can be induced in them. This is a distinct possibility in the "pole face" configuration considered here. Consideration was therefore given to the flexible rubber bonded magnets which are used widely in

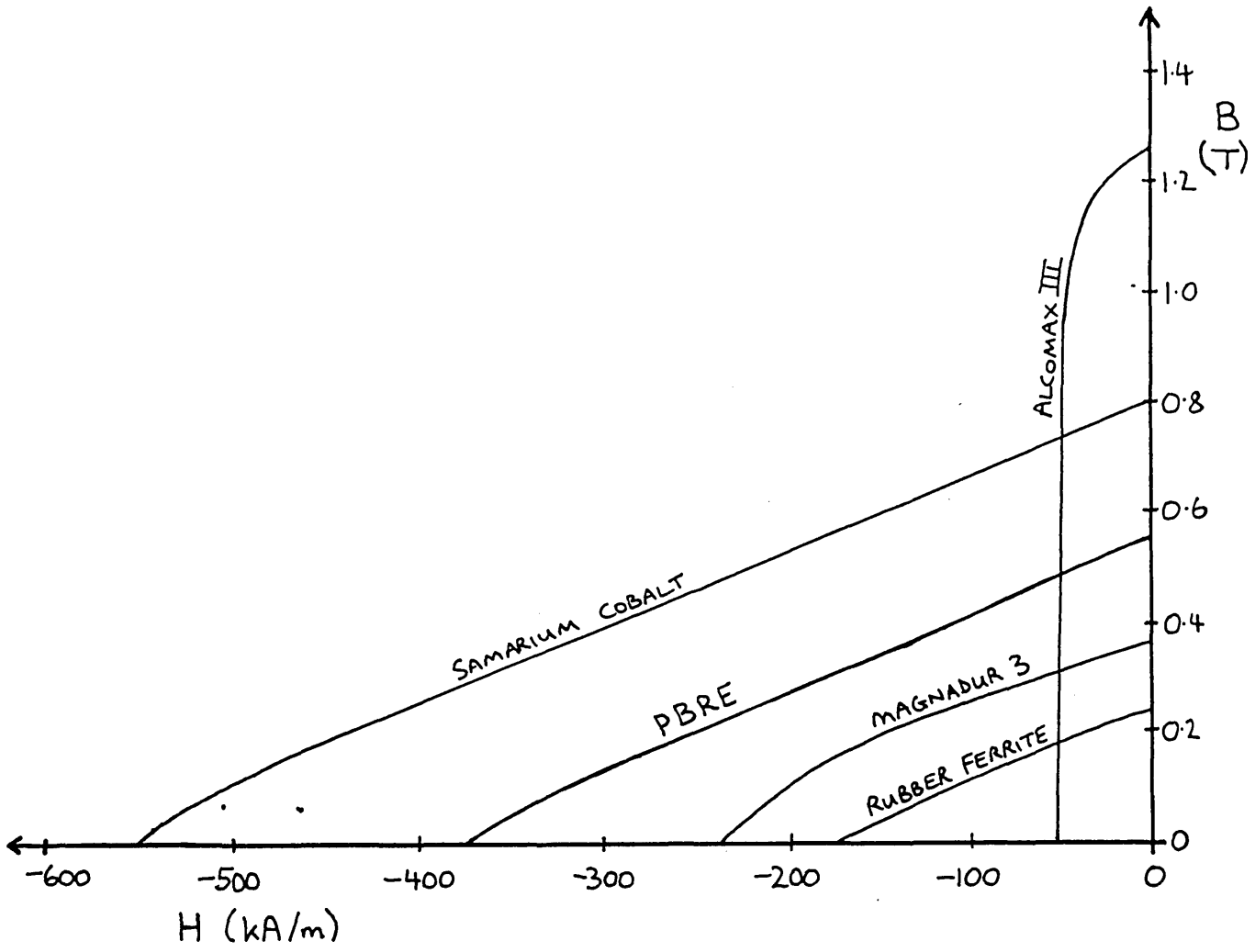


FIG. 6.15. DEMAGNETISATION CHARACTERISTICS OF VARIOUS MAGNETS

applications such as self sealing refrigerator doors. These materials are cheap, are available in a variety of thicknesses, and are non-conducting. They have a B_r of about 0.24T and an H_c of about 195kA/m. Samples of Ferriflex and Plastiform were obtained with a view to using them to form arcuate segments, but it was found that a 5mm thick sheet of either material was difficult to bend, and a 10mm thickness was virtually impossible to bend around the required rotor radius. It was thought that a 10mm thickness of magnet was the minimum that could be considered in order to ensure a reasonable airgap flux, and so the material was rejected. Therefore consideration was given to the use of

ferrite or rare earth magnets. No suitable arcuate shapes were available and so it was decided that the required pole pieces would have to be constructed using method (c), in which the magnets have a wedge shaped cross section. Ferrite magnets have a B_r of about 0.33T, and an H_c of about 240kA/m. They are rather brittle and so must be machined with care but they are cheap and non-conductive. Sintered samarium cobalt magnets have a B_r of about 0.8T and an H_c of about 550kA/m. They are extremely difficult to demagnetise but they are brittle and very expensive. The cheaper polymer bonded rare earth (PBRE) magnet has a B_r of about 0.55T and an H_c of about 380kA/m. It is more expensive than ferrite magnets but is easy to machine. The only disadvantage is its maximum continuous operating temperature of 60°C and peak temperature capability of 125°C. Considering all the factors, it was felt that PBRE was the best choice because:

(a) it has a high H_c making it virtually impossible to demagnetise in normal use;

(b) it is easily machined;

(c) it is relatively cheap;

(d) it has an adequate B_r;

(e) it has a permeability close to μ_0.

A data sheet for HR8 PBRE material is included in Appendix 6A.

6.5.2 Final Dimensions of Rotor

The final dimensioning of the rotor depends on the actual size of the PBRE magnets used. The PBRE bar magnets available are listed in Appendix 6A and a typical magnet is shown in fig. 6.16. The magnetic axis is usually directed along the shortest dimension of the bar, and this is the

magnets "length". The fact that the magnets are situated on the pole faces and must remain in position at high speed, necessitates the use of a strength cylinder around the outside of the rotor. This cylinder must be able to contain the bursting pressure set up by the magnets as the rotor rotates. It must also be non-magnetic to avoid providing a path for the rotor flux to go directly from the North to the South pole without passing through the stator.

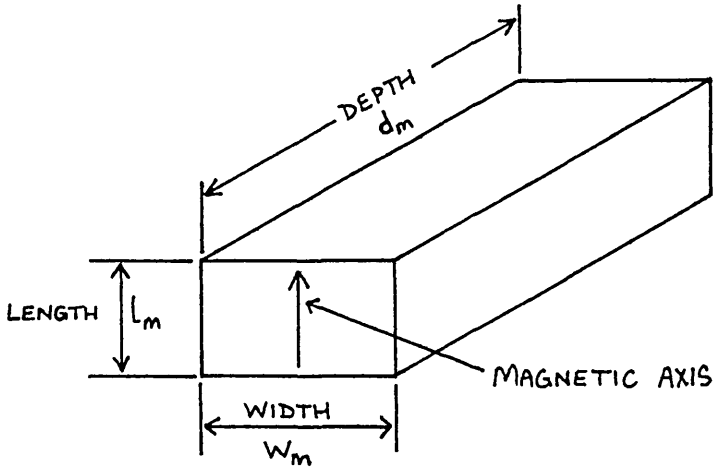


FIG. 6.16. TYPICAL POLYMER BONDED RARE EARTH (PBRE) BAR MAGNET

The wall thickness of the strength cylinder increases the effective airgap between the magnets and the stator, and it should be minimised in order to maximise the airgap flux density. Prior to the magnet size being chosen, it was impossible to calculate what forces the cylinder would have to withstand. It was estimated that a suitable cylinder would require a wall thickness of about 1.5mm to 2.0mm. In addition, it was decided to choose a relatively large airgap of 0.5mm between the ring and the stator to remove any possibility of rubbing at high speed. The chosen magnet length was therefore required to generate a reasonable rotor flux density for a total airgap of between 2.0mm to 2.5mm.

The magnet width W_m was chosen to minimise the amount of machining needed to make an integer number of magnets fit together on each pole face. The magnet depth d_m (the

axial length) was chosen simply to give a "square" rotor. (i.e. axial length of the pole equal to rotor diameter.) This ratio of pole axial length to diameter is popular in motors for well documented reasons and there was no pressing requirement to choose any other ratio. It can be seen in Appendix 6A that there were two magnets available with which it was possible to construct an approximately square rotor within the required diameter limits. They are types M051 and M015 with dimensions of 60 x 10 x 15mm and 50 x 12 x 10mm respectively, (depth d_m x width W_m x length l_m). There was therefore a choice between using a magnet with a magnetic axis length l_m of 15mm or 10mm.

The arrangement of rotor and stator is shown in fig. 6.17(a) and the equivalent magnetic circuit is shown in fig. 6.17(b). The circuit assumes that the rotor and stator iron is infinitely permeable and so the only reluctance in the circuit is that due to the airgaps. The effect of the stator slots on the effective airgap length is also neglected.

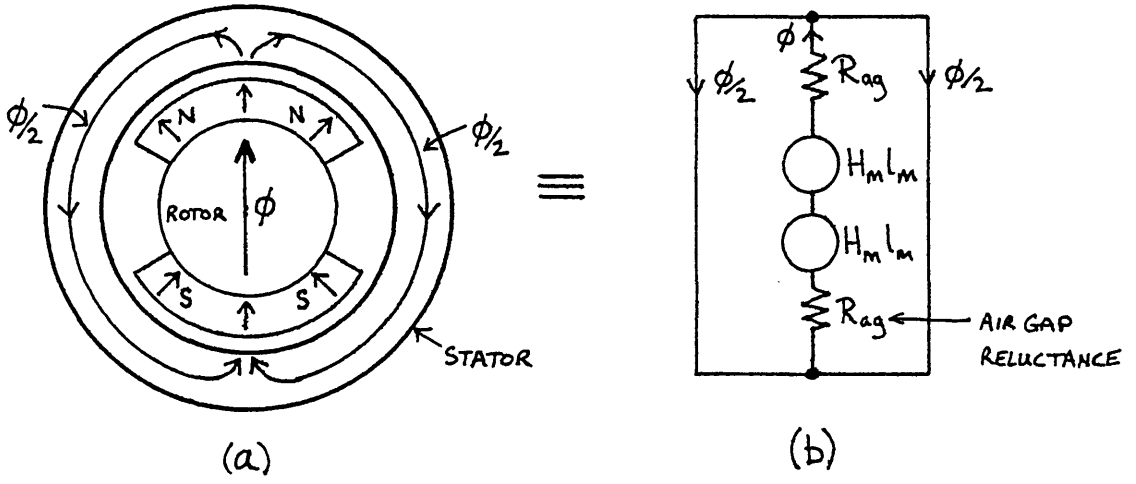


FIG. 6.17. THE ARRANGEMENT OF THE ROTOR AND STATOR (a), WITH THE EQUIVALENT MAGNETIC CIRCUIT (b).

Summing the magnetomotive forces (MMF's) around the magnetic circuit gives:

$$2.H_m.l_m + 2.H_{ag}.l_{ag} = 0 \qquad 6.4$$

where H_m and H_{ag} are the magnetising forces in the magnet and airgap respectively, and l_m and l_{ag} are the respective lengths of the magnet and airgap. The airgap flux density, B_{ag}, is given by:

$$B_{ag} = \mu_0 . H_{ag} \qquad 6.5$$

where μ_0 is the permeability of free space.

Hence: $$B_{ag} = K . H_m \qquad 6.6$$

where $$K = - \frac{\mu_0 . l_m}{l_{ag}} \qquad 6.7$$

The normal demagnetisation characteristic of HR8 PBRE material is linear and can be expressed as:

$$B_m = 1.3145 \times 10^{-6} H_m + 0.55 \qquad 6.8$$

The magnets and airgaps are in series and have the same cross sectional area (if fringing and leakage are neglected). Hence:

$$B_{ag} = B_m \qquad 6.9$$

Combining equations 6.7, 6.8 and 6.9 yields:

$$B_{ag} = \left[\frac{0.55 \, \mu_0 . l_m}{\mu_0 . l_m + 1.3415 \times 10^{-6} \, l_{ag}} \right] \text{ Tesla} \qquad 6.10$$

The resultant flux densities for the two possible magnet lengths with airgaps of 2.0mm or 2.5mm are summarised in fig. 6.18. It can be seen that very little increase in flux density is obtained by using the 15mm magnet rather than the 10mm magnet. In addition, the flux density produced by the 10mm magnet only falls by 4.2% as the airgap increases from 2mm to 2.5mm. The cost of the 15mm magnet

is 50% greater than the cost of the 10mm magnet (for any quantity) and so it is not economically justifiable to spend 50% more to achieve an increase in flux density of only 6.0 to 8.0%, depending on actual airgap. Therefore, it was decided to use the M015 10mm magnet.

		MAGNET LENGTH l_m		% increase in flux density due to extra 5mm magnet length
		10mm	15mm	
AIRGAP	2mm	0.453T	0.481T	6.2
	2.5mm	0.434T	0.467T	7.6

FIG.6.18. TABLE SHOWING RESULTANT AIRGAP FLUX DENSITY FOR VARIOUS COMBINATIONS OF MAGNET LENGTH AND AIRGAP LENGTH

To achieve a reasonably smooth rotor flux, the magnets can be machined into wedge shapes as shown in fig. 6.19(a) so that they fit closely together as shown in fig. 6.19(b).

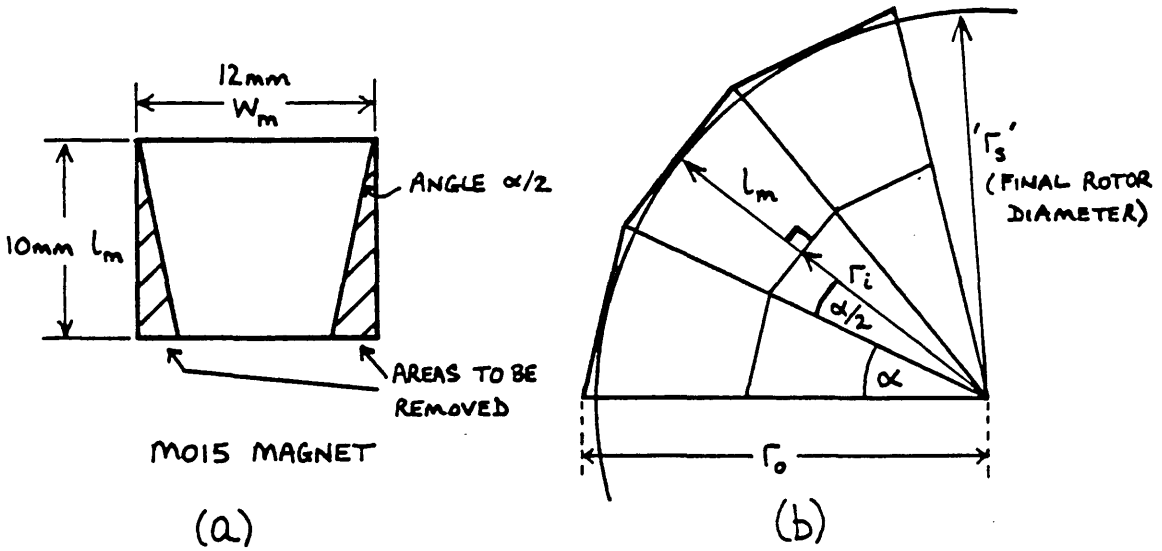


FIG.6.19. CROSS SECTION OF A MACHINED WEDGE SHAPED M015 MAGNET (a) AND THE WAY THE WEDGES FORM AN ARCUATE POLE (b)

If n M015 magnets are used to fill the $(6\pi/7)$ radian pole span, then the angular span "α" of each magnet is given by:

$$\alpha = \frac{6\pi}{7n} \text{ radians} \qquad 6.11$$

The maximum radius of the rotor before it is turned to a smooth finish is r_0, where:

$$r_0 = \frac{6}{\sin\left(\frac{3\pi}{7n}\right)} \text{ mm} \qquad 6.12$$

If the rotor is skimmed just sufficiently to produce a smooth pole surface, the resulting radius r_s is given by:

$$r_s = r_0.\cos\left(\frac{\alpha}{2}\right) \text{ mm} \qquad 6.13$$

$$\text{i.e. } r_s = \left[\frac{6}{\tan\ \frac{3\pi}{7n}}\right] \text{ mm} \qquad 6.14$$

The smooth rotor diameters obtained for values of n of 6, 7 or 8 are 52.56mm, 61.62mm and 70.62mm respectively. These diameters are all within the chosen limits of 50 to 75mm. The quantity of magnet that has to be machined off is greatest for n = 6, but the additional cost of using seven or eight magnets per pole is hardly justified by the reduction in magnet wastage that can be achieved. It is also more sensible to choose the diameter closest to 50mm in order to minimise centrifugal forces on the rotor. In addition, n = 6 results in a "square" rotor having an axial length of 50mm and a diameter of 52.6mm (excluding the strength cylinder). Therefore, it was decided to use six magnets per pole. The steel shaft on which the magnets are mounted requires 14 flats milled on it. With reference to fig. 6.19(b), the diameter D_f across the flats is given by:

$$D_f = 2.r_i \qquad 6.15$$

$$\text{i.e. } D_f = 2(r_s - l_m) \qquad 6.16$$

For six M015 magnets, D_f = 32.6mm. The cross section of the rotor is shown in fig. 6.20. The interpole gaps are filled with paxolin spacers which have the same dimensions as the

magnets. A strength cylinder with a wall thickness of 1.5mm is shown in fig. 6.20. The next stage in the design was to check if a strength cylinder with a wall thickness of 1.5mm to 2.0mm would in fact possess sufficient strength.

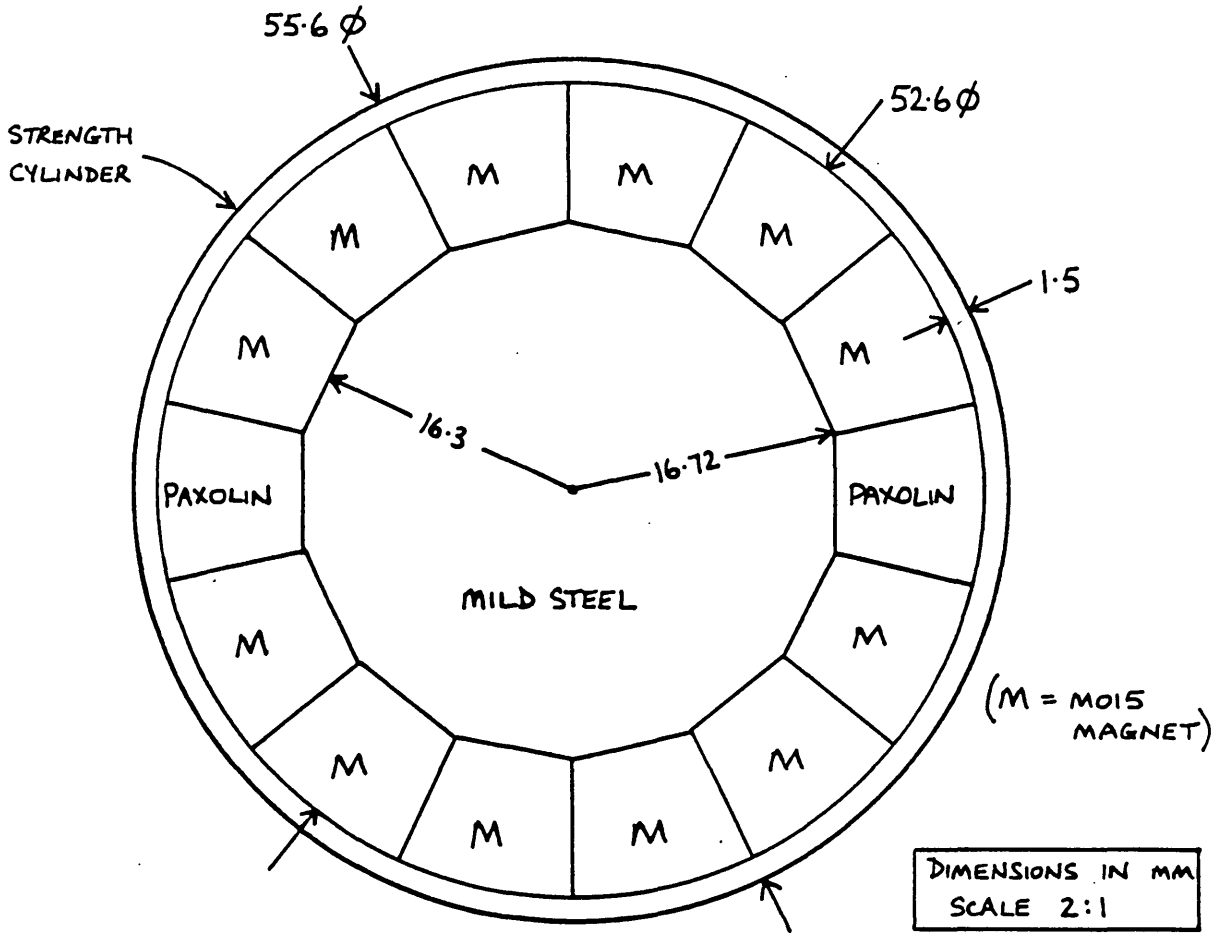


FIG. 6.20. CROSS SECTION OF THE POLYMER BONDED RARE EARTH MAGNET ROTOR

6.5.3 Choice of a Suitable Strength Cylinder for the Rotor

During assembly of the rotor there is a need to glue the magnets onto the steel shaft in order to hold them in place whilst the outer diameter is turned down to a smooth surface. (If glue is not used, the six magnets forming a pole, repel each other.) The glue may assist the strength cylinder during subsequent high speed rotation of the rotor, but it cannot be completely relied upon. Therefore, it was

proposed that the strength cylinder should be strong enough to provide all of the necessary restraining forces on the rotor magnets.

To simplify the stress calculations it was assumed that the magnets and paxolin spacers had the same density. Then, at any rotational speed, the 14 wedge shaped blocks on the surface of the rotor all exert the same force on the inside surface of the strength cylinder. A circumferential tensile stress is set up in the wall of the cylinder in order to contain the bursting pressure. A typical wedge block is shown in fig. 6.21.

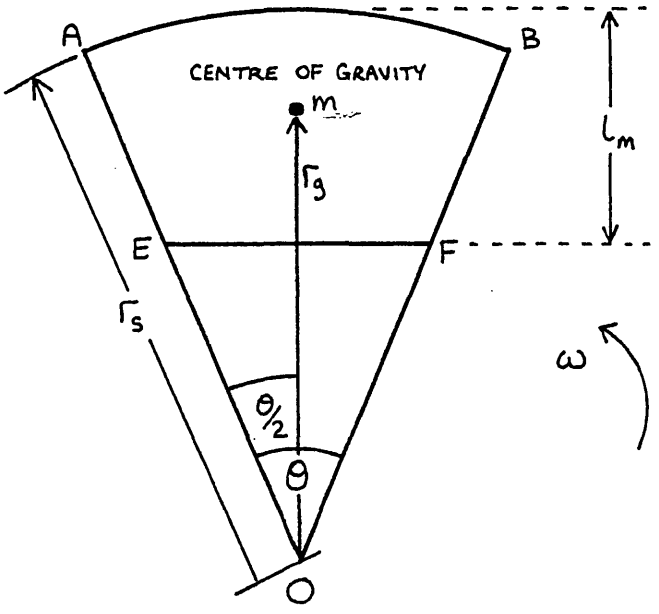


FIG. 6.21. CROSS SECTION OF A TYPICAL MAGNET WEDGE (ABFE)

The centrifugal force F_c required to maintain the wedge of mass m in circular orbit about an axis going vertically (into the page) through point O is given by:

$$F_c = m.r_g.\omega^2 \qquad 6.17$$

where r_g is the distance of the wedge's centre of gravity from O, and ω is the rotational speed about O in radians

per second. It can be shown that r_g is given by the expression:

$$r_g = \frac{4}{3}\left[\frac{r_s.\sin(\theta/2) - (r_s-l_m)^3\tan(\theta/2)}{r_s^2.\theta - 2(r_s-l_m)^2\tan(\theta/2)}\right] \qquad 6.18$$

For r_s = 26.3mm, l_m = 10mm and $\theta = \pi/7$ radians, equation 6.18 gives r_g = 21.57mm. The volume V_w of each wedge is given by:

$$V_w = d_m\left[\frac{1}{2}r_s^2.\theta - (r_s-l_m)^2\tan(\theta/2)\right] \ m^3 \qquad 6.19$$

For d_m = 50mm, equation 6.19 gives a volume $V_w = 4.7286 \times 10^{-6}$ m^3. The density of HR8 PBRE material is 5100kg/m^3 and so the mass of each wedge is 0.0241kg. At a rotational speed of 30000 rpm (3142 radians/sec) the centrifugal force calculated by use of equation 6.17 is found to be 5130N. This force is exerted over an area A_r of the strength cylinder internal surface given by:

$$A_r = (r_s.\theta.d_m) \ m^2 \qquad 6.20$$

For the dimensions given, A_r is found to be 5.9×10^{-4} m^2. The pressure p on the internal surface of the cylinder is simply the force divided by the area, and for the conditions imposed it is found to be 8.7MN/m^2.

The circumferential and radial stresses set up in the cylinder can be calculated using thick cylinder theory based on Lamé's theory as explained by Stephens (6.3). If the cylinder wall thickness is thin ((outside diameter/inside diameter) < 1.3) the radial stress is very small and it is sufficient to use thin cylinder theory (6.4) to calculate the circumferential stress. For the strength cylinder considered here, thin cylinder theory may be used. The circumferential stress σ_c in the cylinder is comprised of a centrifugal force component σ_f due to the mass of the cylinder, and a pressure component σ_p due to the magnets being

restrained by the cylinder. The stress is given by:

$$\sigma_c = \sigma_f + \sigma_p \qquad 6.21$$

where $$\sigma_f = \rho . r_s^2 . \omega^2 \qquad 6.22$$

and $$\sigma_p = \frac{p . r_s}{t} \qquad 6.23$$

where ρ is the density of the cylinder material, and t is the cylinder wall thickness. For complete accuracy, the mean radius of the cylinder should be used in equation 6.22, but for a thin walled cylinder the internal radius r_s gives sufficient accuracy.

To be able to use equation 6.21 it is necessary to know the density of the cylinder material. Several non-magnetic materials were considered and they are summarised in fig. 6.22.

MATERIAL	POISSONS RATIO	DENSITY	MAXIMUM WORKING STRESS	RESISTIVITY	YOUNGS MODULUS	σ_c (SEE TEXT)
		Kg/m^3	MN/m^2	Ωm	GN/m^2	MN/m^2
BRASS	0·35	8500	450	8×10^{-8}	100	210·6
TITANIUM	0·36	4540	960	53×10^{-8}	110	183·5
MAGNESIUM	0·29	1740	95	4×10^{-8}	44	164·0
ALUMINIUM L65	0·34	2800	470	5×10^{-8}	75	171·0
GLASS FIBRE COMPOSITE	—	$\simeq$ 2000	1000	—	42	166·2
CARBON FIBRE COMPOSITE *	—	1500	1900	16×10^{-6} TO 30×10^{-6}	125	162·8

* GRAFIL XA-S

Typical Properties of Possible Strength Cylinder Materials

fig. 6.22

The last column in the table of fig. 6.22 shows the circumferencial stress σ_c that would be present at 30000 rpm in the different strength cylinders with 1.5mm wall thicknesses. It can be seen that most of the materials considered have a sufficiently high maximum working stress to withstand the circumferencial stress σ_c. However, it is necessary to choose a material that has a high resistivity in order to limit the eddy currents induced by ripple in the rotor airgap flux. Such induced eddy currents cause a braking torque on the rotor and so absorb some of the output torque. The absorbed power is dissipated in the cylinder in the form of heat.

The ripple in the rotor airgap flux is accentuated when "open slot" stator designs are used (as in the present case). The situation is shown in fig. 6.23.

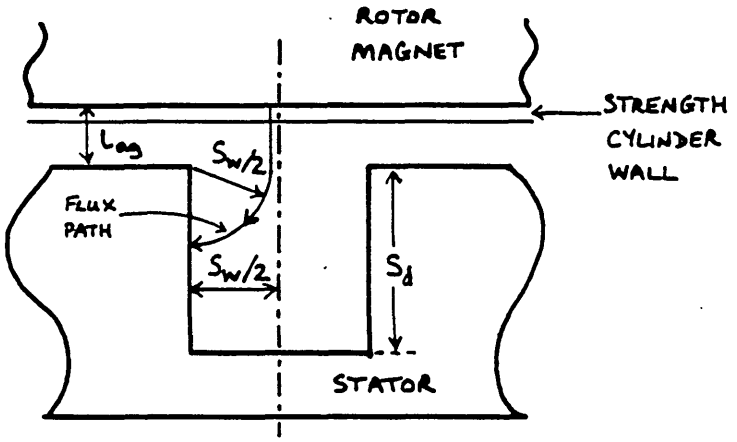


FIG. 6.23. DIAGRAM ILLUSTRATING THE "OPEN SLOT" STATOR DESIGN USED IN THE SQUARE WAVE SYNCHRONOUS MOTOR

A reasonable approximation for the minimum slot flux density is obtained by employing Roter's method (6.5) in which flux is assumed to travel in straight lines and circular paths to the nearest iron. The stator slot width S_w and slot depth S_d are calculated in sections 6.6.2 and 6.6.3, and their values are 6.35mm and 12.7mm respectively. The stator tooth to rotor magnet airgap, l_{ag}, is 2mm and the flux density in this region is 0.453T as calculated by equation 6.10. At the mid point of the stator slot the

airgap l_{ag}' can be approximated by:

$$l_{ag}' = l_{ag} + \tfrac{1}{4}(2\pi \frac{S_w}{2}) \qquad 6.24$$

i.e. $$l_{ag}' = l_{ag} + \frac{\pi S_w}{4} \qquad 6.25$$

For l_{ag} = 2mm and S_w = 6.35mm, equation 6.25 gives l_{ag}' = 7.0mm. Substitution of this airgap length into equation 6.10 yields a minimum mid slot flux density of 0.315T. The rotor airgap flux density therefore varies between values of 0.453T and 0.315T over each slot pitch. The ripple in the flux density has a peak value B_{pk} of 0.07T and the flux pattern is stationary with respect to the stator. However, the rotor strength cylinder and magnets move with respect to the flux ripple and so eddy currents can be induced in them if they are conductive. There is no problem with eddy currents in the PRBE magnets because they have a high electrical resistivity. However, eddy currents in the strength cylinder must be minimised to limit losses and also to prevent the cylinder temperature from rising above 60°C and so damaging the rotor magnet.

The eddy current losses in the cylinder are equivalent to the pole face losses in alternating current machines caused by tooth-ripple flux. Say (6.6) gives a formula for the calculation of such losses but it is difficult to choose the corrects values for substitution into the equation. The eddy current loss in the cylinder can be roughly estimated by assuming that the rotor flux varies sinusoidally over a slot pitch as shown in fig. 6.24. A piece of the conducting sheet forming the cylinder is indicated in fig. 6.24 and it has a circumferencial length equal to one slot pitch. To simplify the description it will henceforth be called the "plate".

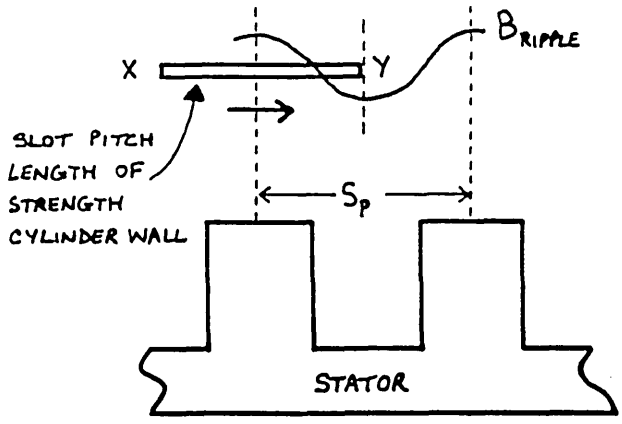


FIG. 6.24 THE INTERACTION OF THE STRENGTH-CYLINDER-WALL WITH THE ROTOR-FLUX-RIPPLE-WAVEFORM "B_{RIPPLE}"

The total flux passing through the plate changes with time and it is assumed that the resulting eddy currents flow in symmetrically located loops as shown in fig. 6.25.

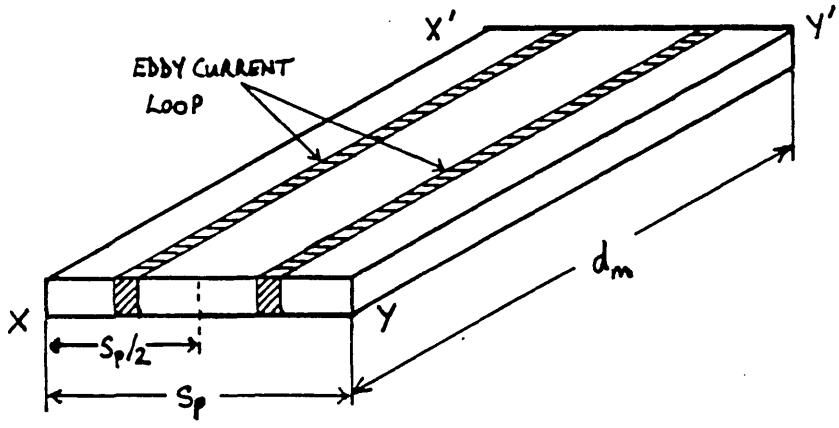


FIG. 6.25. THE SYMMETRICALLY LOCATED LOOPS OF EDDY CURRENT ON THE PLATE

The axial length of each loop is assumed to be equal to the depth d_m of a PBRE magnet. This dimension is much greater than the slot pitch and so the end sections of each loop can be neglected. The flux linking a general loop at any instant can be defined by r, θ coordinates, with the origin located at the centre of the rotor, as shown in fig. 6.26.

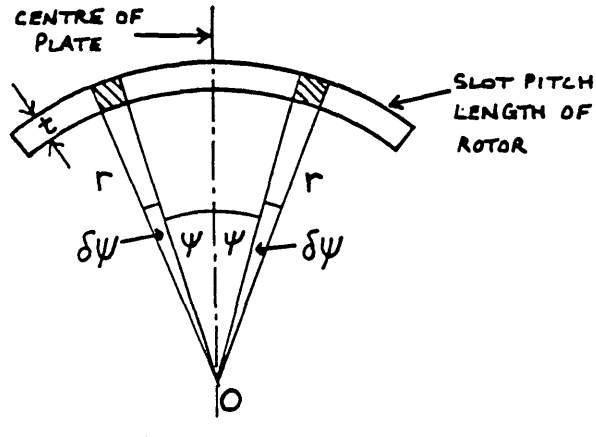


FIG. 6.26 THE USE OF POLAR COORDINATES TO DESCRIBE THE POSITION OF A GENERAL EDDY CURRENT LOOP IN THE PLATE

The flux ripple wave around the airgap has its positive maxima half way across each stator tooth and the location of one maximum can be defined as the starting point (r,0). The ripple flux density, assumed to be negligibly damped by the eddy currents, is then given by:

$$B_{rip} = B_{pk}.\sin(14\theta + \pi/2) \qquad 6.26$$

where θ is the angular displacement from the starting point. It is assumed that the centre of the plate is at the starting point at the instant, t = 0. Then if the plate is rotating at an angular velocity of ω radians per second with respect to the ripple flux wave, the position θ of the centre of the plate at a time t_1 is given by:

$$\theta = \omega.t_1 \qquad 6.27$$

The flux $\phi_{t_1}^{\psi}$ linking a typical eddy current loop of width $2r\psi$ at time t_1 is given by:

$$\phi_{t_1}^{\psi} = \int_{\theta-\psi}^{\theta+\psi} B_{rip}.\,dA \qquad 6.28$$

but $dA = r.d_m.d\theta$, hence:

$$\phi_{t_1}^{\psi} = d_m.r\int_{\omega t_1-\psi}^{\omega t_1+\psi} B_{pk}.\sin\left(14\omega t + \frac{\pi}{2}\right).d\theta \qquad 6.29$$

This yields:

$$\phi_{t_1}^{\psi} = \frac{2r.d_m.B_{pk}}{14}\sin\left(\frac{28\omega t_1 + \pi}{2}\right)\sin(14\psi) \qquad 6.30$$

The voltage $V_{t_1}^{\psi}$ induced in the loop is obtained by differentiating equation 6.30 with respect to time giving:

$$V_{t_1}^{\psi} = 2\omega r.d_m.B_{pk}\cos\left(\frac{28\omega t_1 + \pi}{2}\right)\sin(14\psi) \qquad 6.31$$

The resistance R^{ψ} of any loop is given by:

$$R^{\psi} = \frac{2.\varrho.d_m}{t.r.\delta\psi} \qquad 6.32$$

where ϱ is the resistivity of the plate material. The instantaneous eddy current power dissipation in a loop is therefore given by:

$$\delta P = \frac{(V_t^{\psi})^2}{R^{\psi}} \qquad 6.33$$

and the total instantaneous power dissipation P for all the loops in the plate is given by:

$$P = \int_{\psi=0}^{\psi=\frac{\pi}{14}} \delta P.\, d\psi \qquad 6.34$$

where the limit $\psi = \pi/14$ covers one slot pitch. Integration of equation 6.34 yields the expression

$$P = \frac{\pi}{28}\left(\frac{2.\omega^2.r^3.d_m.t.B_{pk}^{\,2}}{\varrho}\right)\cos^2\left(\frac{28\omega.t_1 + \pi}{2}\right) \qquad 6.35$$

and an equation of this form has an average value P_{av} of one half of its peak value.

$$\text{i.e.}\quad P_{av} = \frac{\pi}{56}\left(\frac{2.\omega^2.r^3.d_m.t.B_{pk}^{\,2}}{\varrho}\right) \qquad 6.36$$

Since there are 12 "plates" on the rotor exposed to rotor flux, the total average power loss P_t is thus 12 times greater than P_{av}. In addition, the radius r is related to the slot pitch S_p by the equation

$$r = \frac{7S_p}{\pi} \qquad 6.37$$

yielding:

$$P_t = \frac{147}{\pi^2}\left(\frac{\omega^2 . S_p^{\,3} . d_m . t . B_{pk}^{\,2}}{\rho}\right) \qquad 6.38$$

This formula is of similar form to that derived by Carter (6.7) for eddy current losses in a metal block. Substitution of the appropriate values into equation 6.38 yields eddy current losses of 1380W for brass and 209W for titanium at 30000 rpm. (Cylinder wall thickness t = 1.5mm; slot pitch S_p = 12.7mm, as calculated in sub-section 6.6.1; cylinder axial length d_m = 50mm; and B_{pk} = 0.07T.) These power dissipations are obviously far too big to tolerate in a practical system. It could have been decided to limit the maximum rotor speed to say 20000 rpm (where the dissipation powers would have been 610W and 93W respectively; the latter figure perhaps being just acceptable). It was realised that the calculation method was giving a high prediction because no account had been taken of skin effect in the plate, or of plate damping. In addition, the actual ripple flux magnitude may be much smaller than that assumed for the calculation. Nevertheless, it was thought that the calculation was sufficiently indicative to justify a decision not to use a metallic strength cylinder. However, some work was done to see if a worthwhile reduction in losses was possible by splitting the metallic strength cylinder into a series of rings. (i.e. a laminated construction with several "plates" making up the total axial cylinder length d_m.) When this is done the resistance at the ends of each eddy current loop becomes significant and there can be some reduction in the losses. Equation 6.38 does not include the loop end resistances and so a modified version was used to investigate the benefits of shorter rings.

The calculations were performed for eddy current loops of the form shown in fig. 6.27(a) and 6.27(b). The loops in fig. 6.27(b) were tried in an attempt to get closer to the current paths that actually occur in practice.

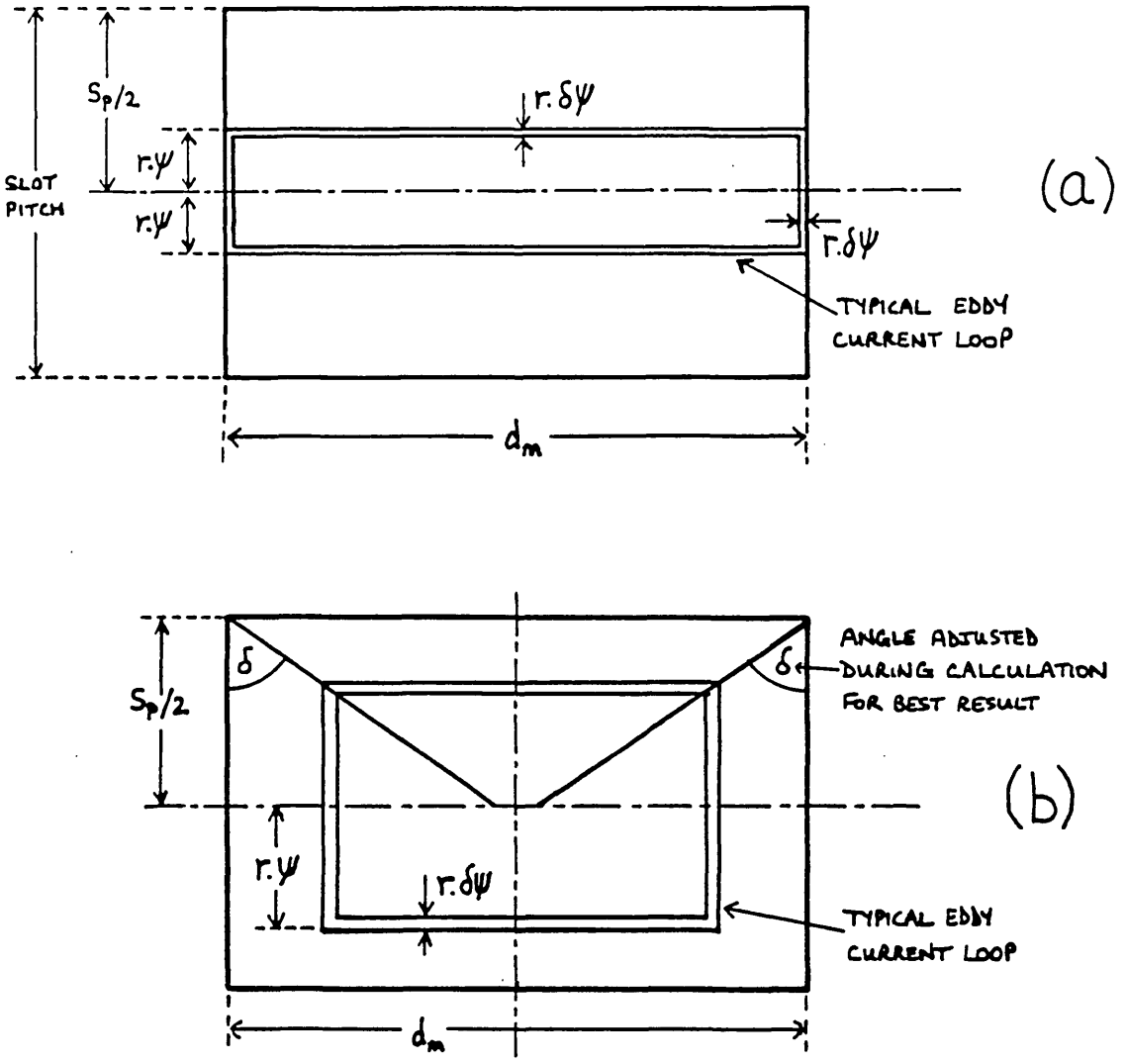


FIG.6.27. EDDY CURRENT LOOP MODELS USED TO INVESTIGATE THE BENEFITS OF SPLITTING THE STRENGTH CYLINDER INTO RINGS. BOTH (a) AND (b) INCLUDE THE EFFECT OF RESISTANCE AT THE LOOP ENDS. THE LOOPS IN (b) ARE CLOSER TO ACTUAL EDDY CURRENT LOOPS.

The calculations performed indicated that a significant reduction in losses could be achieved by splitting the cylinder into a series of rings but the number of rings needed to reduce the loss to less than 10 Watts was of the order of fifty, which leads to constructional difficulties. It was not thought worthwhile pursuing the calculation of the losses any further, although if time had permitted an attempt to use the method published by Russell and Norsworthy (6.8) would have been made. Very recently a paper has been published by Boules (6.9) dealing with the calculation of losses in magnet retaining cylinders but unfortunately it

was far too late to be used in the rotor design.

Having looked at the difficulties caused by the use of a metallic strength cylinder, it was decided to investigate the possibilities of using non-metallic materials. The main contenders are fibre glass and carbon fibre. Fortunately, a generous free sample of Grafil E/XA-S carbon fibre from Courtaulds saved any further deliberations. Appendix 6B contains data on the E/XA-S carbon fibre. Its resistivity when used in a laminated structure, is between $16 \times 10^{-6}\Omega m$ and $30 \times 10^{-6}\Omega m$ and so the eddy current loss in a cylinder made from this material is negligible (about 7W absolute maximum). Details of the final rotor assembly are given in section 6.7.1.

During the assembly of the rotor the individual magnets have to be glued onto the central steel shaft to hold them in position as the carbon fibre is wound on. The adhesive bond can provide some of the magnet restraining force. However, if the entire restraining force is supplied by the carbon fibre cylinder, the magnet is subjected to a maximum compressive stress of $8.7MN/m^2$ (as calculated in section 6.5.3). This stress is well below the compressive strength limit of $31.40\ MN/m^2$ for PBRE magnets. If the adhesive bond provides all of the restraining force, the magnet is subjected to a maximum tensile stress of $13.79MN/m^2$ and this is below the tensile stress limit of $17.66MN/m^2$. Therefore the magnet should not physically fail.

The strain in the carbon fibre strength cylinder caused by the maximum imposed circumferential stress of $162.8MN/m^2$ is 1.302×10^{-3}. The increase in the cylinder circumference is accompanied by an increase in rotor radius from 27.80mm to 27.836mm. Thus the static airgap clearance of 0.5mm is more than adequate to accomodate the rotor expansion at top speed.

6.6 The Design of the Stator

The major dimensions to be decided in the design of the stator were:

(a) the slot pitch S_p;

(b) the slot and tooth widths, S_w and t_w respectively;

(c) the slot depth S_d;

(d) the length of the stator, l_s;

(e) the thickness of the stator backing iron, S_b;

(f) the lamination thickness, t_{lam};

(g) the number of turns per phase, N_{ph}.

6.6.1 Slot Pitch

The slot pitch has already been fixed by the choice of rotor diameter and airgap. The total rotor diameter is 55.6mm including a 1.5mm thick strength cylinder, and so with an airgap of 0.5mm the stator bore has a diameter d_{si} of 56.6mm. The slot pitch S_p for a stator with 14 slots is given by:

$$S_p = \pi . d_{si}/14 \qquad 6.39$$

i.e. $S_p = 12.70\text{mm}$

6.6.2 Tooth and Slot Widths

The requirement for a low inductance stator winding placed a restriction on the form of slot that could be used in the stator. Very often slots are semi-closed (or sometimes even fully closed) as shown in fig. 6.28, in order to limit the ripple in the airgap flux density caused by the

slot openings.

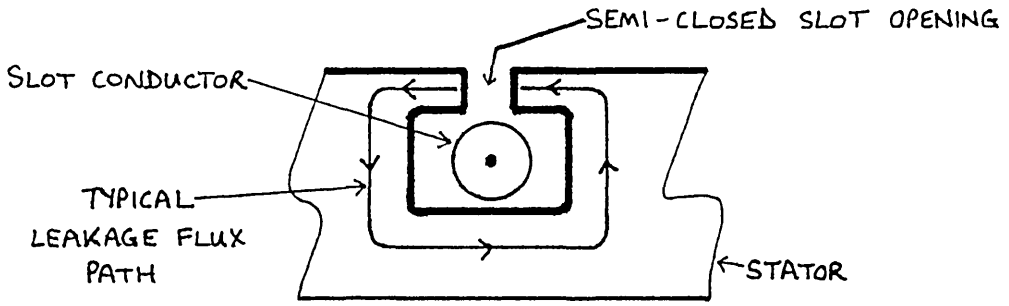


FIG. 6.28. DIAGRAM SHOWING THE FORM OF A SEMI-CLOSED STATOR SLOT AND A TYPICAL LEAKAGE FLUX PATH.

However, semi-closed slots were rejected in this case because:

(a) they provide a good path for the leakage flux and so increase the stator inductance. This was particularly undesirable in the project machine;

(b) they make the insertion of ready wound stator coils very difficult.

It was therefore decided to use an open slot shape even though this results in a large ripple component in the airgap flux density. The tooth width, slot width, and slot pitch in an open slot stator are related by the equation:

$$t_w + S_w = S_p \qquad 6.40$$

It can be assumed that all of the rotor flux over one slot pitch is concentrated into the width of one tooth as shown in fig. 6.29. If the magnets produce a flux density B_{ag} in the airgap, the flux density B_t in the tooth is given by:

$$B_t = \left(\frac{S_p . B_{ag}}{t_w} \right) \quad \text{Tesla} \qquad 6.41$$

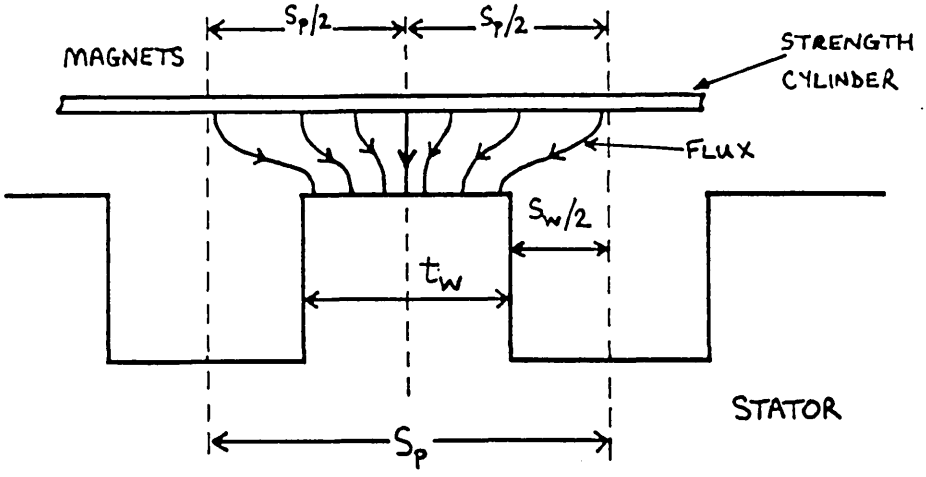


FIG. 6.29. THE CONCENTRATION OF ROTOR FLUX OVER ONE SLOT PITCH INTO THE WIDTH OF ONE TOOTH

From equation 6.40 it can be seen that for a practical machine the tooth width must be less than the slot pitch. The tooth width must be chosen to prevent the flux density exceeding the saturation point of the tooth iron. In addition, it is sensible to modify equation 6.41 to take account of any leakage flux circulating around the slot currents. It was assumed that a 10% increase in B_t would be sufficient to allow for any possible leakage flux, and so equation 6.41 becomes:

$$B_t = \left(\frac{1.1 \; S_p . B_{ag}}{t_w} \right) \qquad 6.42$$

Equation 6.42 can be used to calculate the minimum tooth width required providing all the other variables are known. However, it is also important to consider the ratio of slot width to tooth width because this has a direct bearing on the harmonics present in the rotor airgap flux density. The ripple in the airgap flux density causes eddy currents to be induced into the conductive components of the rotor, and the calculations in section 6.5.3 were based only on the fundamental of the ripple flux. The eddy current losses are worse for the higher harmonics and so it is logical to try to minimise the presence of such harmonics. It was thought reasonable to assume that the higher harmonic content would

be minimised by making the slot width equal to the tooth width: i.e. the tooth width is half the slot pitch. Substitution of $B_{ag} = 0.453T$, $S_p = 12.7mm$ and $t_w = S_p/2$ into equation 6.42 gives:

$$B_t = \left[\frac{1.1 \times 12.7 \times 0.453}{(12.7/2)}\right] = 1.0T$$

A flux density of 1.0T is well below the saturation point of all the commonly used lamination steels. Therefore the tooth and slot widths were fixed at 6.35mm each.

Having fixed the tooth width, slot width and airgap length, it is possible to make a correction to the airgap length to take account of the slotting in the stator. The method was devised by F.W. Carter (6.10) and its use for various stator and rotor slot configurations is summarised by Say (6.11). When the equations are rewritten so that they contain the notation used in this chapter, it is found that the corrected airgap l_{ag}' is given by:

$$l_{ag}' = C.l_{ag} \qquad 6.43$$

C is a factor known as Carter's coefficient and it is defined by:

$$C = \frac{t_w + S_w}{t_w + S_w(1 - \sigma)} \qquad 6.44$$

and σ is calculated from:

$$\sigma = \frac{2}{\pi}\left[\tan^{-1}\left(\frac{S_w}{2l_{ag}}\right) - \frac{l_{ag}}{S_w}\ln\left\{1 + \left(\frac{S_w}{2l_{ag}}\right)^2\right\}\right] \qquad 6.45$$

Equation 6.45 shows that σ is only dependent on the slot width and the physical airgap length. When the appropriate dimensions are substituted into equations 6.43, 6.44 and 6.45, it is found that the corrected airgap length has a

value of 2.484mm as compared to the physical rotor magnet to stator tooth distance of 2.0mm. If the corrected airgap length is substituted into equation 6.10, the airgap flux B_{ag} is found to be 0.4347T. This is not significantly lower than the uncorrected value of 0.453T. However, the corrected value is used in all further calculations in this chapter unless otherwise stated.

6.6.3 The Slot Depth

The depth of the stator slots has a direct effect on:

(a) the power output of the motor. A deep slot can accomodate more conductors than a shallow slot. Hence the total slot current is proportional to the slot depth and this has a corresponding effect on the power handling capability of the stator.

(b) The leakage inductance of the winding in the slot and the eddy currents induced in the winding conductors.

The slot leakage inductance L_{l1} for a single conductor in a slot is:

$$L_{l1} = \frac{\mu_o . l_s}{S_w} \left[S_d - \left(y + \frac{2d}{3} \right) \right] \quad \text{Henry} \qquad 6.46$$

where l_s is the axial length of the slot and y and d are as shown in fig. 6.30. For a slot containing N_{ph} conductors in series the leakage inductance is increased by a factor equal to the square of the turns. In addition there are two slots per phase winding and so the total phase leakage inductance L_{ln} is given by:

$$L_{ln} = \frac{2 . N_{ph}^2 . \mu_o . l_s}{S_w} \left[S_d - \left(y + \frac{2d}{3} \right) \right] \qquad 6.47$$

(Note slots per pole per phase = 1.)

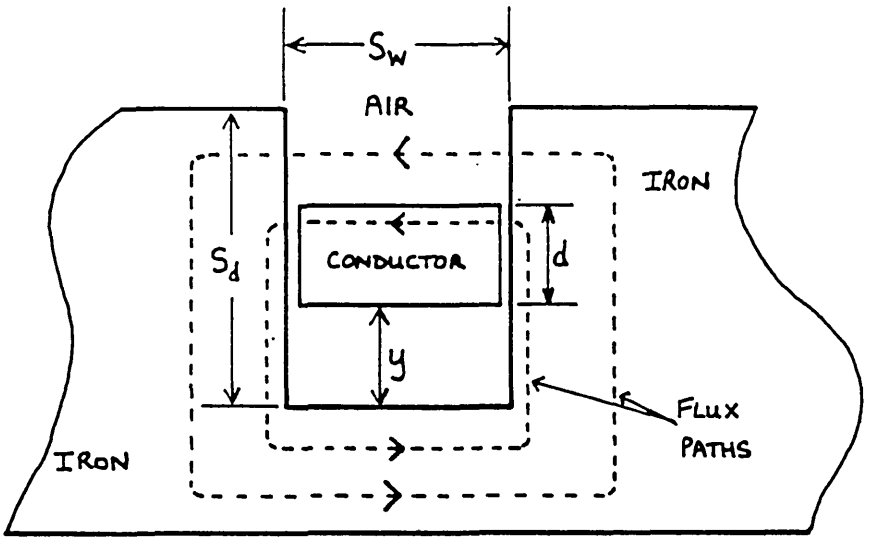


FIG.6.30. DIAGRAM SHOWING THE RELEVANT DIMENSIONS FOR THE CALCULATION OF SLOT LEAKAGE INDUCTANCE

Full use is made of the slot cross sectional area when the conductors totally fill the slot. Substituting $d = S_d$ and $y = 0$ into equation 6.47 gives:

$$L_{ln} = \frac{2.N_{ph}^2.\mu_o.l_s.S_d}{3S_w} \qquad 6.48$$

i.e. $$L_{ln} \propto S_d \qquad 6.49$$

It was therefore decided to choose the slot depth such that a reasonable power output could be achieved without incurring an excessively large leakage inductance. The power output P_{out} of the motor is related to the rotor torque T by the expression:

$$P_{out} = T.\omega \qquad 6.50$$

where ω is the angular velocity of the rotor in radians per second. The steady-state motor torque can be calculated by use of the formula giving the force F on a conductor of length l carrying a current I in a magnetic field of flux density B:

i.e. $$F = B.I.l \qquad 6.51$$

For the machine considered here, B represents the airgap flux density set up by the excitation system (i.e. the rotor magnets). The flux interacts with the stator current I to produce a force. The stator current is assumed to act in the airgap. The instantaneous "airgap" current per slot, I_{slot}, is given by :

$$I_{slot} = \delta . S_d . S_w . K_p \qquad 6.52$$

where δ is the maximum permissible current density in the conductors, and K_p is the slot utilization factor. The force F_r^1 on the rotor due to the current in one slot of length l_s is equal and opposite to the force on the stator. Substitution of equation 6.52 in equation 6.51 yields:

$$F_r^1 = B_{ag} . \delta . S_d . S_w . K_p . l_s \qquad 6.53$$

When the motor is operating correctly there are 12 slots carrying current at any instant. The polarities of the various slot currents are arranged so that the total force on the rotor is 12 times that produced by a single slot current. Hence:

$$F_r^{12} = 12 . F_r^1 \qquad 6.54$$

The total force on the rotor is acting at the mean airgap radius r_{gm} given by:

$$r_{gm} = \frac{1}{2}\left[d_{si} - l_{ag}\right] \qquad 6.55$$

The rotor torque T is then:

$$T = F_r^{12} . r_{gm} \qquad 6.56$$

Therefore substituting equations 6.53, 6.54, 6.55 and 6.56 into 6.50 yields:

$$P_{out} = 12 . B_{ag} . \delta . S_d . S_w . K_p . l_s . r_{gm} . \omega \qquad 6.57$$

Reasonable values of δ and K_p are 4.65A/mm^2 (3000A/in^2) and 0.4 respectively. An approximate power output can be estimated by assuming that the slot length l_s is the same as the rotor axial length (i.e. 50mm) and there is no fringing flux at the ends of the rotor and stator which can react on the end winding of the stator. The mean airgap radius is 27.3mm and so at 30000 rpm the power output is approximately given by:

$$P_{out} = (12 \times 0.435 \times 4.65 \times 10^6 \times S_d \times 6.35 \times 10^{-3} \times 0.4 \times 50 \times 10^{-3} \times 27.3 \times 10^{-3} \times 1000\pi) \text{ Watts}$$

$$\text{i.e. } P_{out} = 2.644 \times 10^5 . S_d \text{ Watts} \qquad 6.58$$

The output powers for a slot-depth to slot-width ratio of a half, one, and two are summarised in fig. 6.31.

$S_d : S_w$ RATIO	S_d (mm)	P_{out} (Watts)
0·5	3·175	839
1·0	6·35	1679
2·0	12·70	3358

FIG. 6.31. THE OUTPUT POWER OF THE SQUARE WAVE MOTOR FOR VARIOUS $S_d : S_w$ RATIOS

It was felt that a power output of 3.358KW (4.50 hp) was the order of power level that should be obtainable from a machine of the dimensions considered. Such a power level was within the capabilities of the MOSFET transistors that were to be used in the inverter. Calculations later on in this chapter show that the phase leakage inductance is very small for a slot depth of 12.7mm and so this slot depth was chosen. The power output predicted by equation 6.57 is optimistic because the eddy current losses in the motor have been totally neglected. The torque predicted by equation 6.56 is 1.07Nm for the rated current density. A knowledge of the maximum motor torque is needed for the selection of a suitable dynamometer.

6.6.4 The Stator Axial Length, l_s

There are three possibilities for the stator length. It can be longer, shorter, or the same length as the rotor. If the stator is longer than the rotor there is a good chance that all the rotor flux links with the stator current, and so the rotor flux is fully utilised. If the stator is shorter than the rotor the cost and weight of the stator is reduced, but some rotor flux is wasted. A good compromise is to make the stator the same length as the rotor; i.e. l_s = 50mm.

6.6.5 The Thickness of the Stator Backing Iron S_b

In a 2 pole machine the back of the stator must carry one half of the total pole flux ϕ_p without saturating, as shown in fig. 6.32.

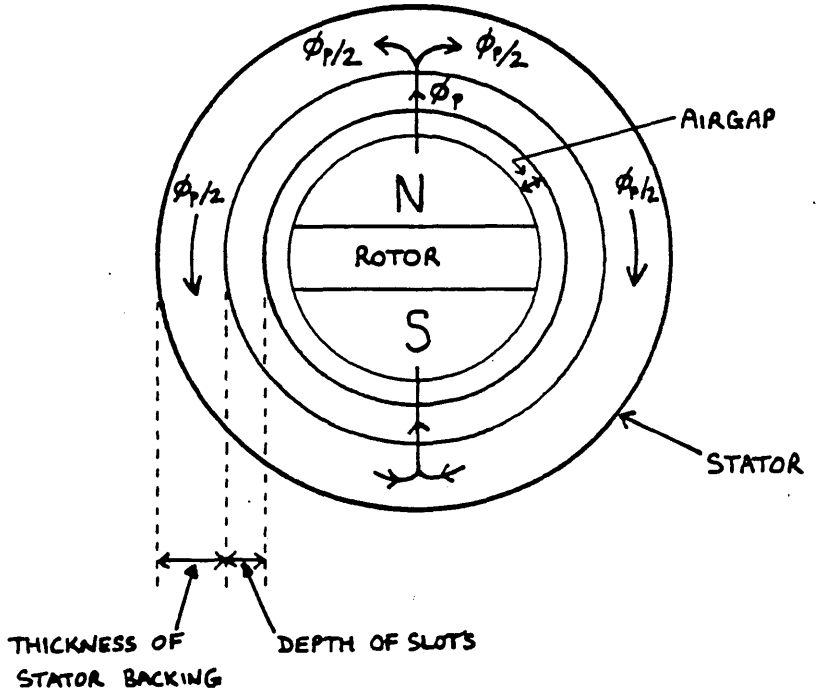


FIG. 6.32. DIAGRAM SHOWING THE ROUTES TAKEN BY THE ROTOR FLUX THROUGH THE STATOR BACKING IRON

The pole flux ϕ_p is given by:

$$\phi_p = B_{ag}.A_p \quad \text{Weber} \qquad 6.59$$

where A_p is the magnet pole area given by:

$$A_p = \left[6\pi.r_s.d_m/7\right] \quad m^2 \qquad 6.60$$

The cross sectional area A_{sb} of the stator back is given by:

$$A_{sb} = S_b.l_s \quad m^2 \qquad 6.61$$

The flux density in the stator backing, B_{sb}, is given by:

$$B_{sb} = \frac{\phi_p/2}{A_{sb}} \quad T \qquad 6.62$$

Combining equations 6.62, 6.61, 6.60 and 6.59 gives:

$$B_{sb} = \left[\frac{3\pi.B_{ag}.r_s.d_m}{7.S_b.l_s}\right] \quad T \qquad 6.63$$

To allow for any flux in the stator back due to stator currents it was decided to increase B_{ag} by 20%. Then with $d_m = l_s$, equation 6.63 becomes:

$$B_{sb} = \left[\frac{3.6\pi.B_{ag}.r_s}{7S_b}\right] \quad T \qquad 6.64$$

If it is assumed that B_{sb} can have a value of 1.5T without the stator iron saturating, the minimum permissible value of S_b is found to be 12.31mm. The outer diameter d_{so} of the stator is given by:

$$d_{so} = d_{si} + 2(S_d + S_b) \qquad 6.65$$

i.e. $d_{so} = 106.62\text{mm}$

It was therefore decided to make d_{so} equal to 110mm.

6.6.6 The Thickness of the Stator Laminations

The presence of a square wave of rotor flux rotating within the stator at high speed makes it desirable to choose low loss laminations. The fundamental frequency within the square wave machine is 500Hz at 30000 rpm and hence it is really necessary to use much thinner laminations than is standard. (The laminations used in 400Hz aircraft electrical equipment are of the order of 0.05mm to 0.13mm (2 to 5 thou") in thickness.) However, at the time that the design of the motor was completed there were insufficient funds available for the purchase of suitable thin lamination material. To avoid the loss of valuable construction time it was decided to use whatever lamination material was readily available. The need for a 14 slot stator meant that standard ready made laminations could not be used since they all have slot numbers that are multiples of three. A batch of 100 four inch square pieces of lamination material with a thickness of 0.635mm (25 thou") was donated by a colleague. Whilst the thickness was far from ideal, it was thought worthwhile to build a stator with these laminations. The reasoning behind this decision was that if the square wave motor was found to work adequately, there would then be good cause to believe that a thin lamination stator would result in appreciable performance improvements.

6.6.7 The Number of Turns per Phase, N_{ph}, and the Minimum Wire Diameter

The stator slot dimensions were chosen in section 6.6.3 to enable a maximum motor rating of about 3kW to be achieved. This power requirement was within the capabilities of a MOSFET inverter and the phase voltage and current ratings were chosen to suit the available MOSFETs. The values chosen were ±80 volts and 6.25 amps, and the reasons for their choice are described in Chapter 5.

During motoring operation, the back-emf generated in each phase winding is very nearly equal to the applied phase

voltage, providing the phase winding impedance is small. The motor is required to rotate at 30000 rpm with an 80 volt amplitude applied phase winding voltage, and at this speed the back-emf must therefore also have an amplitude of about 80 volts. The generated emf E_{ph} in a phase winding of N_{ph} turns can be calculated by modifying equation 6.1 to give:

$$E_{ph} = 2.B_{ag}.N_{ph}.r_{gm}.\omega.l_s \qquad 6.66$$

The conductors are all assumed to be at the mean airgap radius r_{gm}. In addition, fringing flux between the ends of the rotor and stator is ignored, and so the active length of each conductor cut by the rotor flux is equal to the stator axial length l_s. Substitution of the relevant values into equation 6.66 gives the number of turns per phase:

$$N_{ph} = 80/(2 \times 0.435 \times 27.3 \times 10^{-3} \times 1000\pi \times 50 \times 10^{-3})$$

$$N_{ph} = 21.4 = 21.5$$

Unfortunately, this result was not obtained at the time of the initial motor design because an incorrect value for r_{gm} was used. The error was only detected when the stator construction had been completed and so it could not be corrected. The dimension used for r_{gm} was actually the radius corresponding to a point halfway down the stator slots, and this gave a value for N_{ph} of 14.1 turns. A value of 14.5 turns per phase was chosen so that the ends of each phase winding would come out of the stator at opposite ends. This arrangement prevents the end winding at one end of the stator from becoming too bulky due to an excessive number of connections. The incorrectly chosen number of turns obviously results in a reduction in available motor output power and torque with a feed of given voltage.

Equations 6.56 and 6.57 can be rewritten in terms of the number of turns per phase and the instantaneous phase current I_{ph} to give:

$$T = 12.B_{ag}.N_{ph}.I_{ph}.l_s.r_{gm} \qquad 6.67$$

and $$P_{out} = T.\omega \qquad 6.68$$

For N_{ph} = 14.5 and I_{ph} = 6.25 amps, equations 6.67 and 6.68 give T = 0.646Nm and P_{out} = 2.03kW. The calculated torque and power for 21.5 turns per phase are 0.957Nm and 3.006kW respectively. Therefore, the error unfortunately resulted in a reduction of the motor rating by 33%.

The quasi-square current waveform flowing in each phase winding has an average value I_{av} over a half cycle given by:

$$I_{av} = \frac{6}{7} I_{ph} \qquad 6.69$$

The corresponding rms current I_{rms} in the winding is:

$$I_{rms} = \sqrt{\frac{6}{7}} I_{ph} \qquad 6.70$$

Equation 6.67 can therefore be rewritten in terms of the rms phase current:

$$T = 12.\sqrt{\frac{7}{6}}.B_{ag}.N_{ph}.I_{rms}.l_s.r_{gm} \qquad 6.71$$

This equation is useful for calculating the torque when only the rms phase currents are known, such as in the case of the experimental results discussed in Chapter 9.

The gauge of copper wire required for the phase windings is determined by the maximum phase current I_{ph}. The minimum cross sectional area of the wire A_w can be calculated by:

$$A_w = \frac{I_{ph}}{\delta} \qquad 6.72$$

The area A_w is sufficient for the wire to be able to support the phase current indefinitely, such as in the case of the

motor being stalled. For I_{ph} = 6.25 amps and δ = 4.65 x $10^6 Am^{-2}$, the minimum area is found to be 1.34mm^2, and for a round section conductor this corresponds to a wire with a diameter of 1.31mm. This is between S.W.G. 17 and S.W.G. 18. A plentiful supply of 0.9mm diameter wire was available and so it was decided to use a twin strand of this wire wound in bifilar form. The current rating of each strand is 3.0 amps. The combined rating of 6.0 amps is sufficiently close to the desired rating to be adequate, especially since during normal motor operation the 6.25 amp phase current has an rms value given by equation 6.70 of only 5.78 amps.

6.6.8 Calculation of the Resistance and Self Inductance of a Typical Phase Winding

The resistance of a phase winding, R_{ph}, formed by two parallel strands of wire is given by:

$$R_{ph} = 2.\varrho.N_{ph}.l_t/(\pi.d^2) \qquad 6.73$$

where ϱ is the resistivity of copper, d is the diameter of the copper wire, and l_t is the mean length of a typical turn in the winding. If all the turns are assumed to be located at a radius r_{cm} half way down the slots, and then to travel across the end of the machine at the same radius, the mean length l_t is given by:

$$l_t = 2(l_s + \pi.r_{cm}) \qquad 6.74$$

Substitution of the appropriate values into equations 6.74 and 6.73 gives a mean length of turn of 0.318m and a phase resistance of 0.065 ohm. In practice the winding has to be longer than the stator slots to allow room for the end winding, and so the actual phase resistance is larger.

The self inductance of a phase winding is calculated in a similarly approximate manner. A cross section of the stator and rotor is shown in fig. 6.33. Conductors repres-

enting a phase winding with N_{ph} turns are indicated on the diagram.

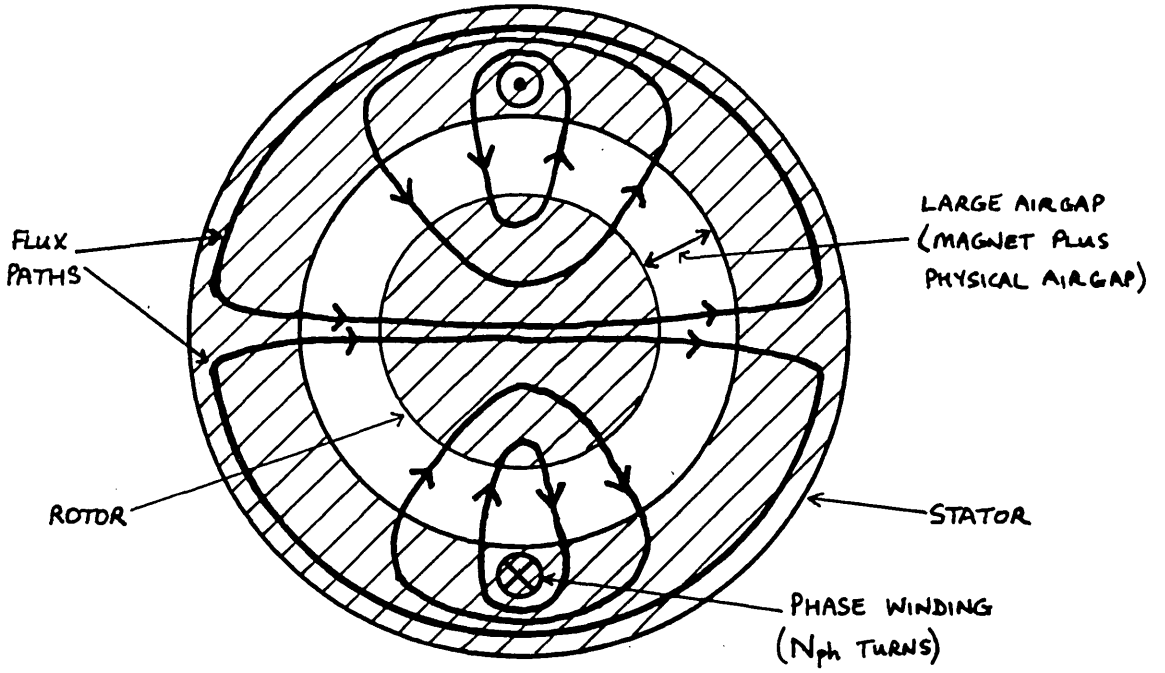


FIG. 6.33. CROSS SECTION OF THE ROTOR AND STATOR SHOWING THE FLUX GENERATED BY A PHASE WINDING

If the reluctance of the iron is neglected, the phase inductance depends on the reluctance of the airgap. The effective airgap length L_{eff} is equal to the sum of the magnet length (as PBRE has a permeability of μ_o), the strength cylinder wall thickness, and the physical airgap length. Consequently, l_{eff} is relatively large and has a value of 12mm. It can be shown that the self inductance L_s of a concentrated coil is approximately given by:

$$L_s = \left[\frac{\mu_o . \pi . N_{ph}^2 . l_s . r_{rs}}{2 . l_{eff}} \right] \qquad 6.75$$

where r_{rs} is the mean radius between the rotor steel and stator iron given by:

$$r_{rs} = \frac{1}{2} \left[d_{si} - l_{eff} \right] \qquad 6.76$$

Substitution of the relevant values into equations 6.76 and 6.75 gives L_s = 38.6μH. The total phase winding inductance

is the sum of the self inductance L_s and the phase leakage inductance L_{ln}. The leakage inductance is calculated using equation 6.48 and has a value of 17.6μH. Some allowance must be made for the leakage inductance of the end winding of each phase and it was suggested that L_{ln} should be increased by about 60% to take account of this. The total phase inductance is then found to be 67μH. This is a very small value of inductance for a motor phase winding. The time constant of the phase winding is about 1.03ms and it was thought that this was good enough for use in a square wave motor application. The equivalent circuit for a phase winding is shown in fig. 6.34.

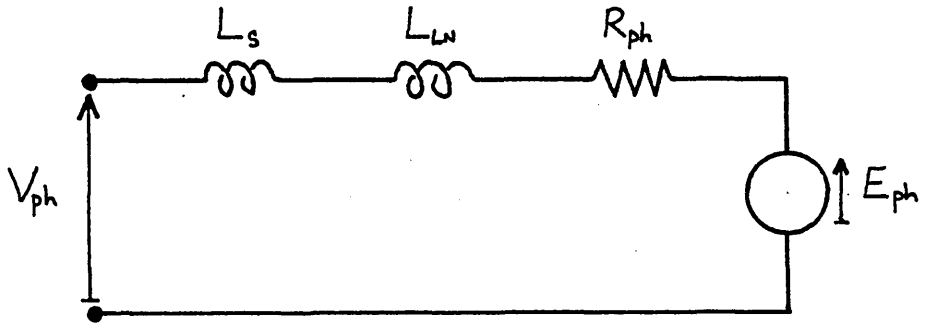


FIG. 6.34. THE EQUIVALENT CIRCUIT FOR A PHASE WINDING OF THE SEVEN PHASE SQUARE WAVE SYNCHRONOUS MOTOR

6.6.9 Stator Copper Loss

The total stator copper loss P_c is given by:

$$P_c = 7.I_{rms}^2.R_{ph} \qquad 6.77$$

Substituting equation 6.70 into 6.77 yields:

$$P_c = 6.I_{ph}^2.R_{ph} \qquad 6.78$$

For I_{ph} = 6.25 amps and R_{ph} = 0.065Ω, the predicted stator copper loss is 15.2 watts. The actual loss is larger

because the practical value of R_{ph} is bigger than that predicted in section 6.6.8.

6.6.10 Final Proposed Stator Lamination Cross Section

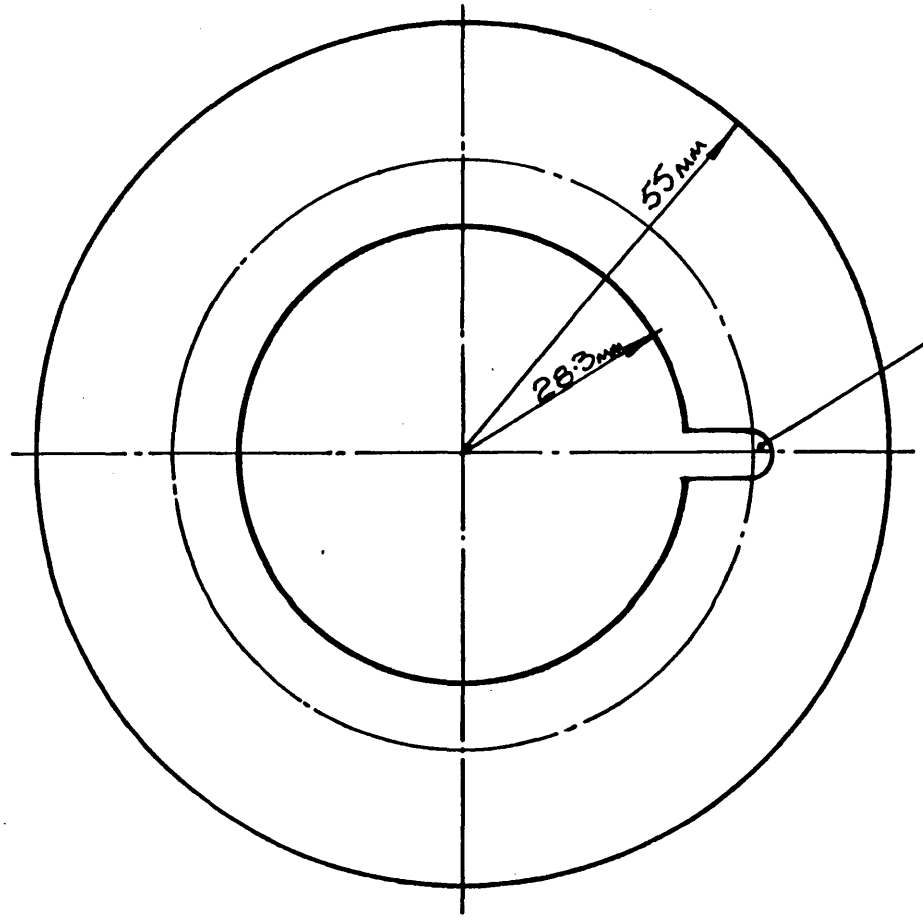
Having considered all of the points discussed in sections 6.6.1 to 6.6.9 it was decided that the dimensions calculated for the stator were suitable, and so the production of laminations with a cross section of the form shown in fig. 6.35 was started. (see fig. 6.41 for photograph of the lamination stack.)

6.7 Constructional Details of the Motor

It was originally intended to mount the stator in a tubular aluminium case. However, before construction began, a survey was made of the old scrap motors available in the laboratory. A 3 phase 0.5 horsepower induction motor was found which had an internal case diameter of 110mm exactly, with a rotor diameter of 54.5mm and a stator axial length of 48.16mm. It was decided to mount the square wave stator in the case of the induction motor.

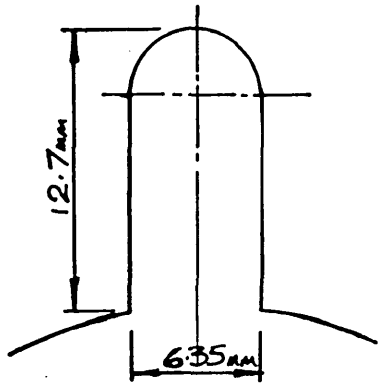
6.7.1 The Rotor Construction

The rotor shaft was designed so that the bearings were correctly located in the bearing housings of the reclaimed case. The dimensions of the mild steel shaft are shown in fig. 6.36 and a photograph of the shaft is shown in fig. 6.37. Twelve MO15 PBRE magnets and two paxolin blocks were then machined into the required wedge shapes with the dimensions as chosen in section 6.5.2. The magnets and paxolin spacers were then glued onto the 14 sided central portion of the steel shaft by using a powerful adhesive called Loctite Multi-bond. The magnets forming each rotor pole have to be glued on because the individual magnets repel each other. Once the magnets were securely bonded, the outside diameter of the rotor was turned down to 52.6mm.



14 SLOT STATOR

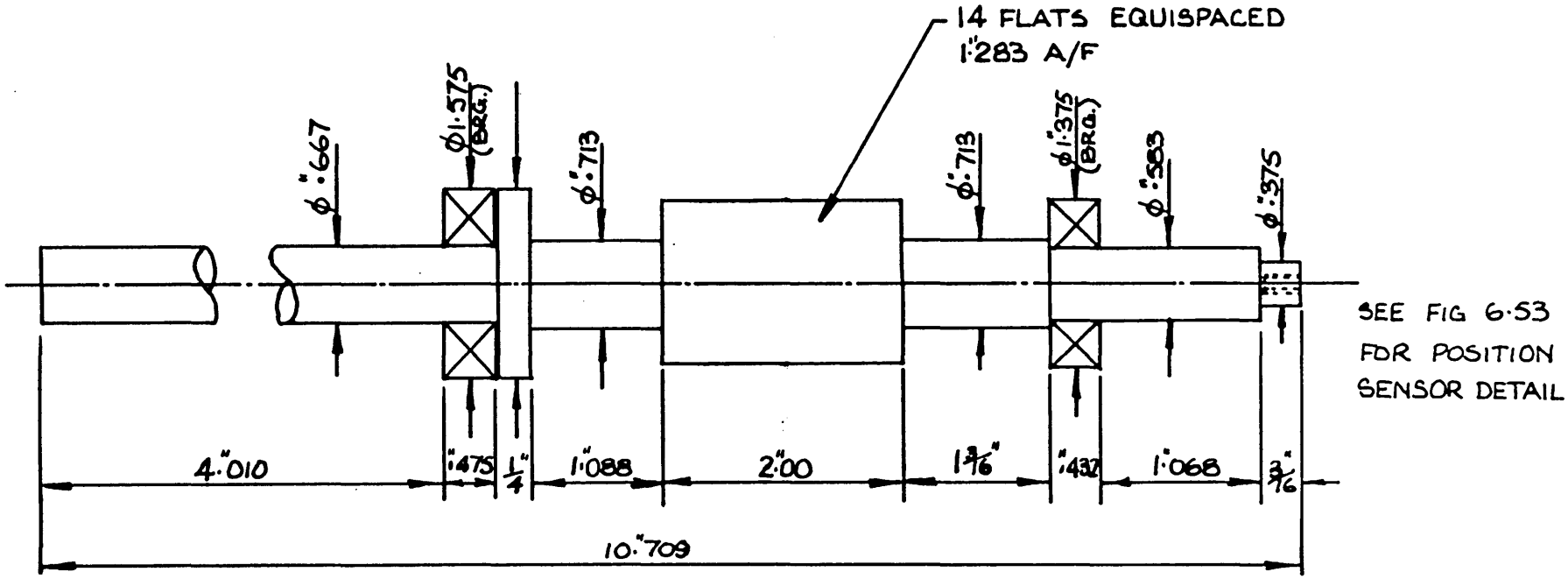
14 SLOTS,
AS DETAIL,
EQUISPACED



SLOT DETAIL

APPROX 78 x 25 THOU. LAMINATIONS

FIG 6.35 PROPOSED STATOR LAMINATION DETAIL



MATERIAL: STEEL

FIG. 6·36 ROTOR DETAIL

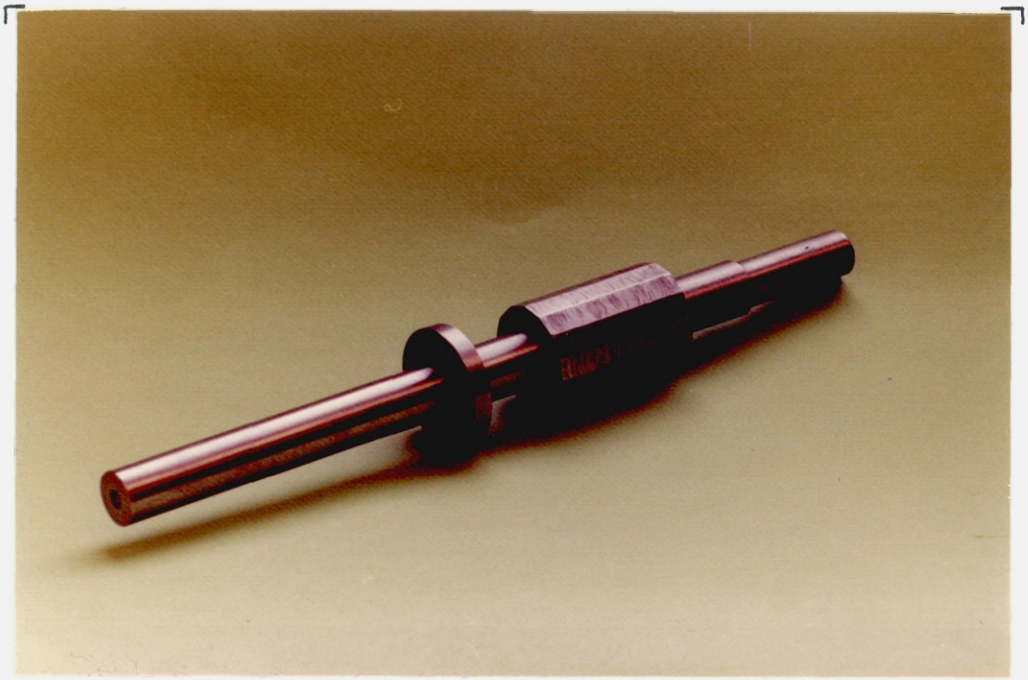


FIG. 6.37. PHOTOGRAPH OF THE MILD STEEL ROTOR SHAFT

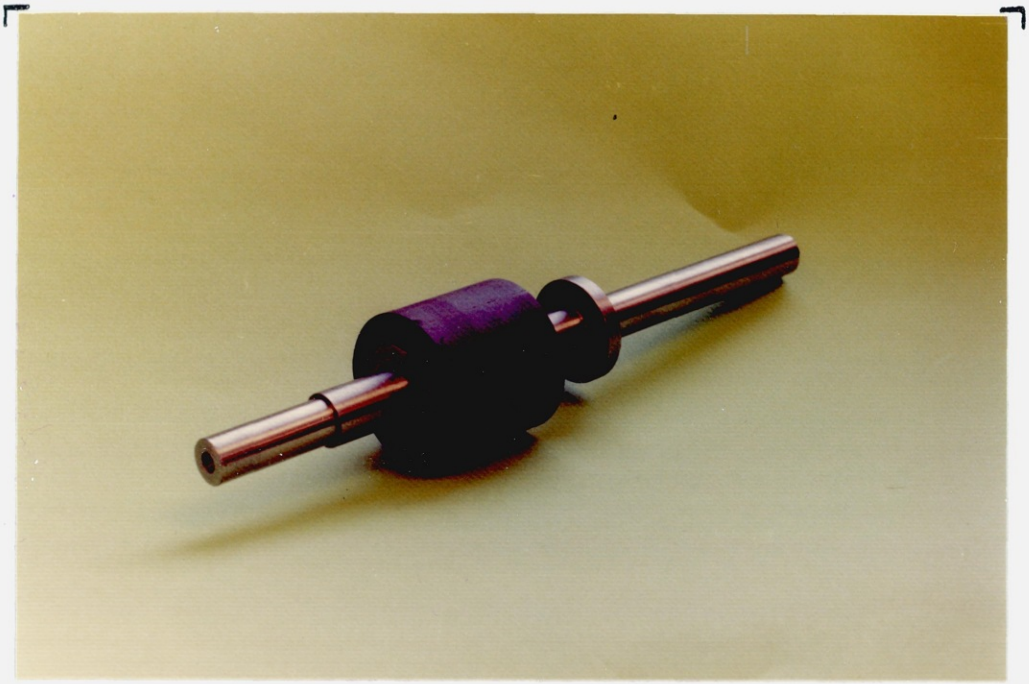


FIG. 6.38. PHOTOGRAPH OF THE ROTOR SHOWING THE POLYMER BONDED RARE EARTH MAGNET POLES

During the turning operation some nylon cable ties were fixed around the magnets to prevent them breaking free. The ends of the magnets were trimmed and their overall axial length (their depth d_m in fig. 6.16) was then 47mm. A photograph of the turned rotor is shown in fig. 6.38.

The carbon fibre sample supplied by Courtaulds was Grafil E/XA-S in a 2 x 3000 filiament tow form. Appendix 6B contains a data sheet for the carbon fibre. It was decided to bond the carbon fibre with a low temperature curing epoxy so that the PBRE magnets would not be damaged. A suitable epoxy was found to be Araldite CY219 which is used in combination with a hardener HY219 and an accelerator DY219. This epoxy cures in about eight hours when it is held at a temperature of 50°C. The carbon fibre tow was carefully wound onto the rotor magnets to form a layer. Copious amounts of CY219 were painted onto the layer of fibre. Another layer of fibre was then wound on and coated with CY219. A total of four layers of carbon fibre were wound onto the rotor to achieve the required thickness of 1.5mm. The fibre was then knotted around the shaft to hold it in place during the curing process. A piece of plastic tubing with an internal diameter of 55.56mm (35/16") was pushed over the carbon fibre to prevent the CY219 adhesive running off whilst curing. Two circular paxolin cheeks had previously been glued at the ends of the rotor magnets so that a sealed cylinder was formed around the carbon fibre and magnets when the plastic tubing was put into place. The assembly was kept at 50°C for eight hours to cure the CY219 adhesive. The plastic tubing and paxolin end cheeks were then broken off and it was found that the surface finish on the rotor was very good except for a few small blemishes caused by air bubbles. Some extra CY219 adhesive was painted on to fill up the small imperfections. When the rotor had fully cured it was found necessary to machine off a very small quantity of excess adhesive and carbon fibre in order to achieve a final concentric diameter of 55.7mm. (Courtaulds had confirmed that machining of a carbon fibre laminate does not seriously weaken it.) The rotor was

finally coated with several protective layers of polyurethane varnish to prevent any damage of the machined carbon fibre surface. A photograph of the finished rotor is shown in fig. 6.39.

6.7.2 The Stator Construction

As stated earlier in this chapter, the available lamination material was supplied in the form of four inch squares. The squares were clamped together and a central hole drilled through them. A locating bolt was passed through the hole to ensure that the laminations remained concentric during the following operations. In addition, four holes were drilled around the edges of the block and these were to be used for the stator clamping bolts in the final assembly. Fourteen 6.5mm holes were then bored in the appropriate positions to form the bottoms of the slots. Bolts were passed through some of the holes to keep the whole assembly rigid when the clamps were removed. The block was then turned on a lathe to achieve an outside diameter of 100mm. Finally the central bolt was removed and the middle of the stator bored out to 56.1mm.

The next step was to separate all of the laminations so that formation of the slots could be completed. Before doing this a mark was made along one side to enable the laminations to be correctly orientated when reassembled. Having separated the laminations, a 6.35mm (1/4") punch was used to remove the material between the holes forming the slot bottoms and the central bore. A simple jig was used to simplify the positioning of each lamination during punching. 89 laminations were required to produce a 50mm long stator and this entailed the punching of 1246 slots by hand on a fly-press. Having punched all of the slots, the stator was reassembled and the slots were filed to remove any bad irregularities. The laminations were then separated so that they could be deburred, degreased and varnished. The laminations were finally bolted together with suitable insulation on the bolts to prevent the laminations being shorted

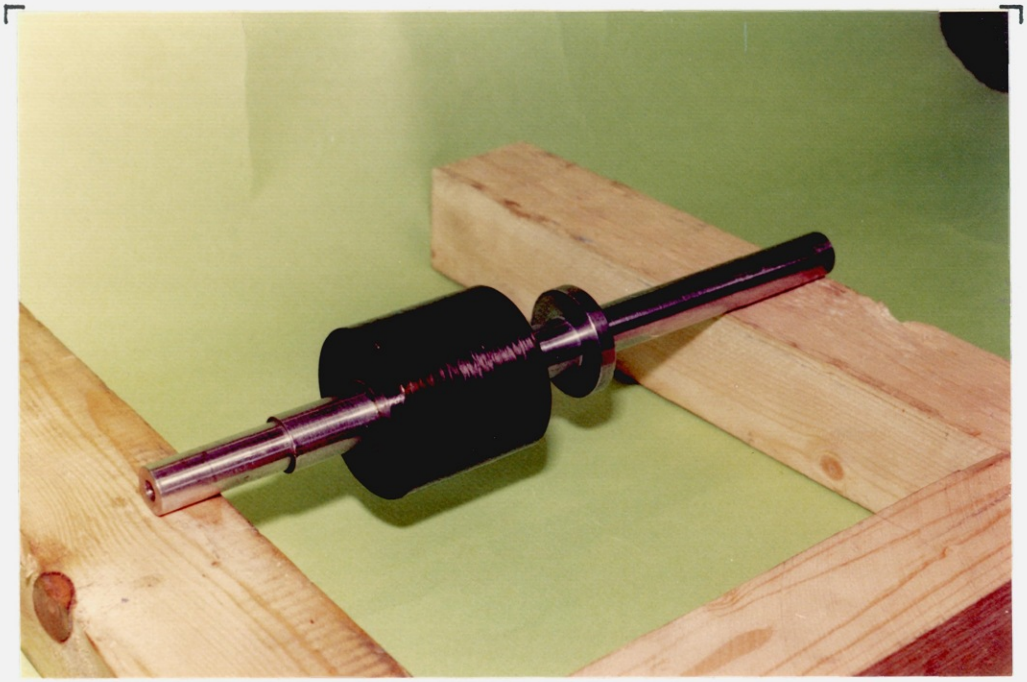


FIG. 6.39. PHOTOGRAPH OF THE FINISHED ROTOR

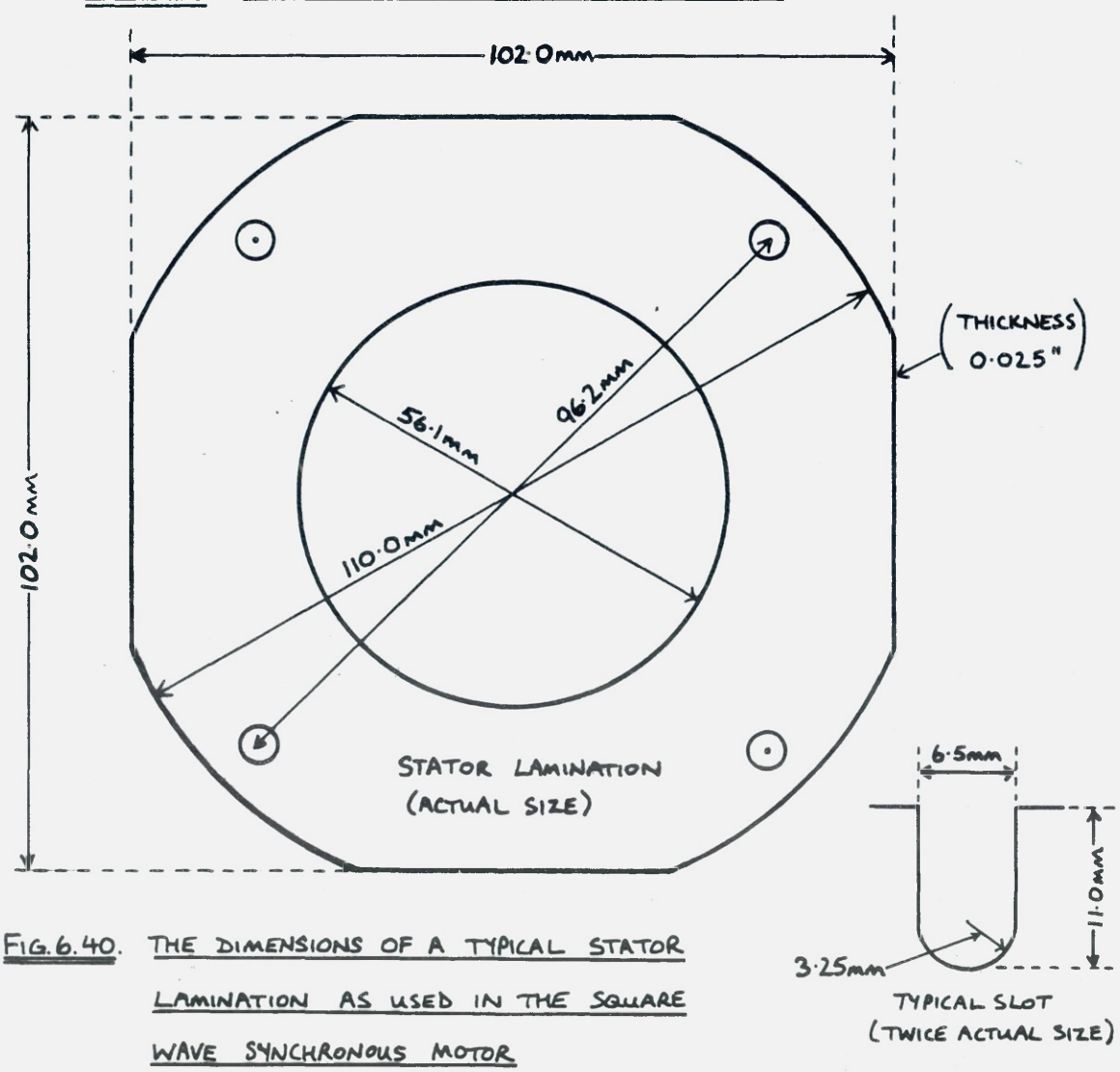


FIG. 6.40. THE DIMENSIONS OF A TYPICAL STATOR LAMINATION AS USED IN THE SQUARE WAVE SYNCHRONOUS MOTOR

together. A 3.175mm thick steel clamping plate was used at each end of the stator to spread the clamping pressure over the whole lamination area. The internal diameter of each clamping plate was 83.2mm so that it cleared the bottom of all of the slots.

The dimensions of a typical lamination are shown in fig. 6.40. The slot depth of 11.0mm is slightly less than the planned size but there is still plenty of room for the actual phase winding. The depth of the stator backing iron is also slightly smaller than planned due to the four flats on the lamination edges, but it was not thought to be of any significance. A photograph of the assembled stator is shown in fig. 6.41.

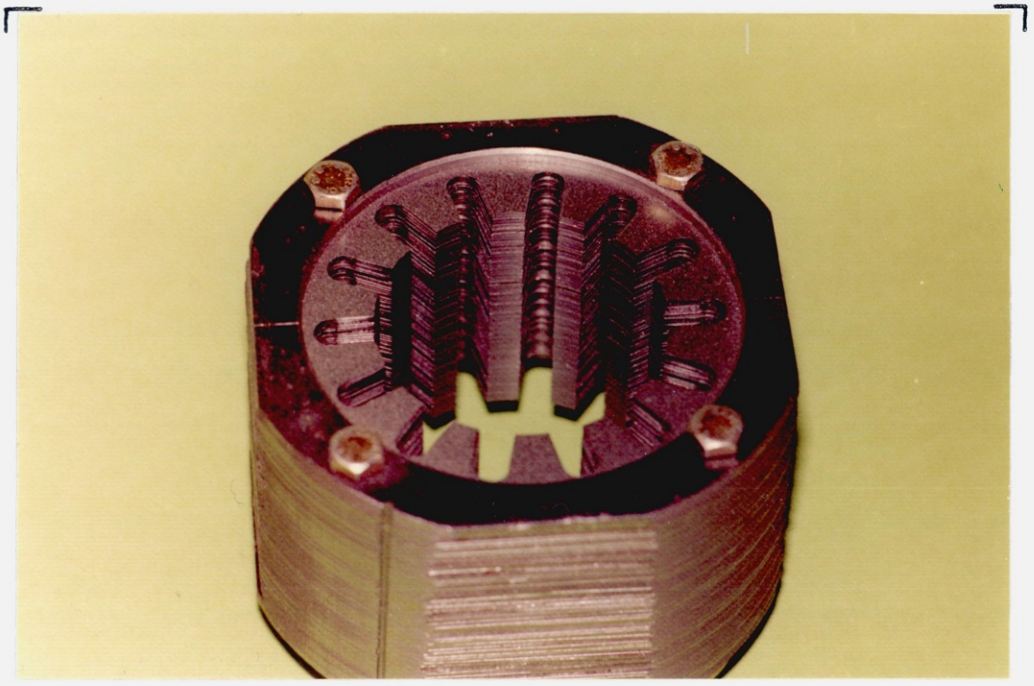


FIG. 6.41. PHOTOGRAPH OF THE ASSEMBLED STATOR LAMINATIONS

Having completed the stator metalwork, melanex slot liners were glued into the slots so that the phase windings would be protected from any sharp metal edges. In order to make the end windings of the stator reasonably symmetrical, it was decided to make each phase winding from two coils connected in series, with the end winding of each coil bent in opposite directions around the stator periphery. One

coil had 7 turns and the other coil 7.5 turns. The use of a 7.5 turn coil meant that the neutral connection between the seven phases could be made at one end of the stator, whilst the interconnections between the 7 and 7.5 turn coils along with the phase connections to the 7 turn coils, could be made at the other end of the stator. A typical phase winding connection is shown in fig. 6.42.

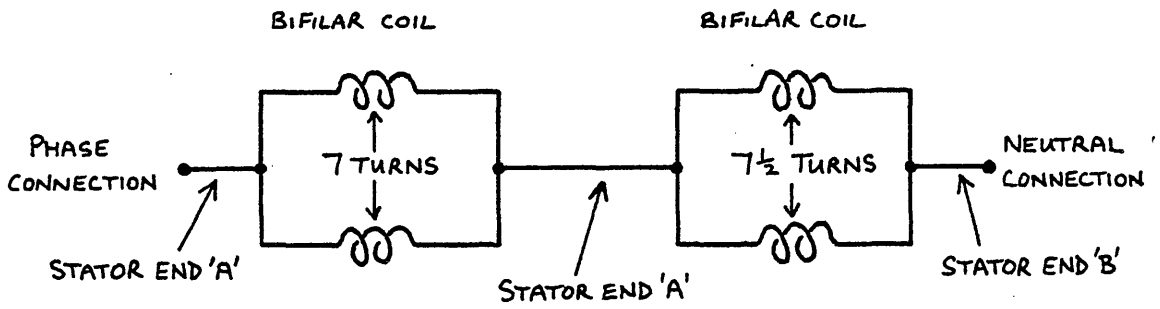


FIG. 6.42. THE TYPICAL PHASE WINDING CONNECTION ARRANGEMENT

It was thought best to common all the neutral connections within the motor to reduce the number of connecting leads needing to be brought out of the body. As already discussed in section 6.6.7, the coils were wound in bifilar form using 0.9mm diameter enamelled copper wire. Each coil was wound on a wooden former with a hexagonal cross sectional shape as shown in fig. 6.43. The mean length of each turn was longer than that predicted in section 6.6.8 and so it was expected that the phase resistance would also be larger. The mean length per turn is about 0.408m which gives a resistance R_{ph} of 0.083Ω when substituted into equation 6.73.

The 14 coils making up the seven phases were placed into the stator and the necessary power leads were soldered on. The end windings were laced up and the conductors were fixed into the slots by "potting" them in rapid araldite. The winding was then liberally soaked with polyurethane varnish to prevent the individual turns vibrating and possibly shorting together. A photograph of the finished stator

is shown in fig. 6.44.

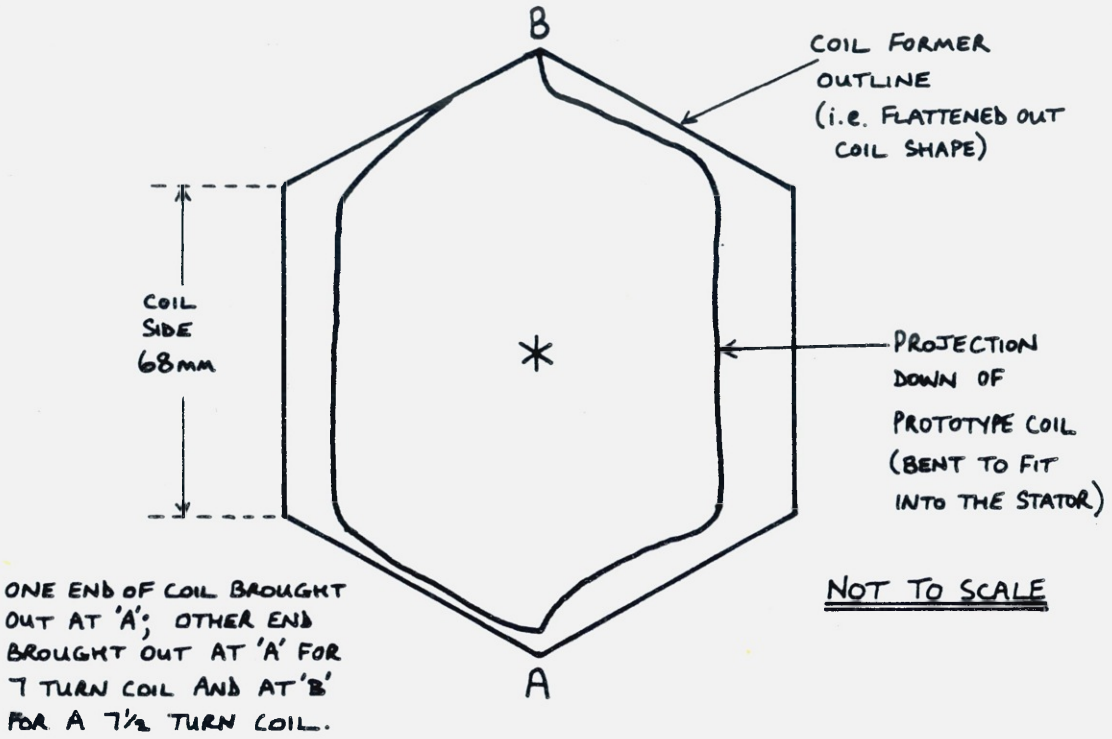


FIG. 6.43. TYPICAL HEXAGONAL FORM OF EACH PHASE COIL

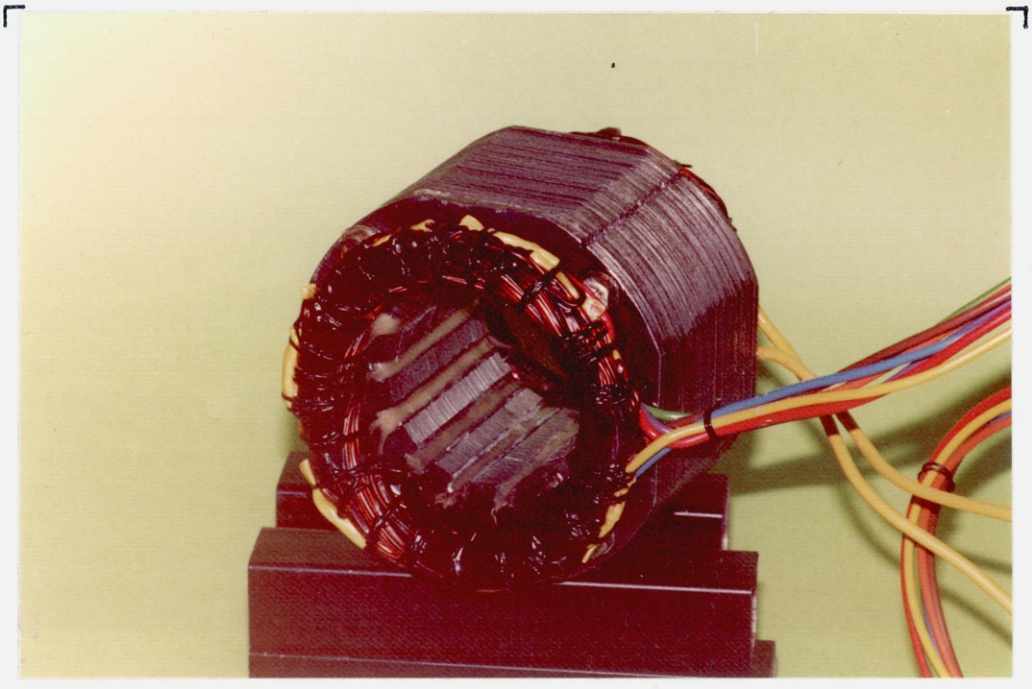


FIG. 6.44. PHOTOGRAPH OF THE FINISHED STATOR.

The stator was a tight push fit in the induction motor body. It was positioned so that the rotor magnets were located centrally within the stator ironwork. A locking screw was arranged on the motor case to hold the stator firmly. To ensure that the stator was concentric with the rotor bearings it was necessary to slightly bore out the stator internal diameter (to 56.43mm). Therefore, with the rotor diameter of 55.7mm, there is a radial airgap of 0.365 mm. This is fractionally smaller than the designed value but it is adequate. Standard ball bearings were used for the rotor since it was thought that their life would be sufficient for the duration of the tests. The phase and neutral connections were brought out via a terminal box and a photograph of the finished motor is shown in fig. 6.45.

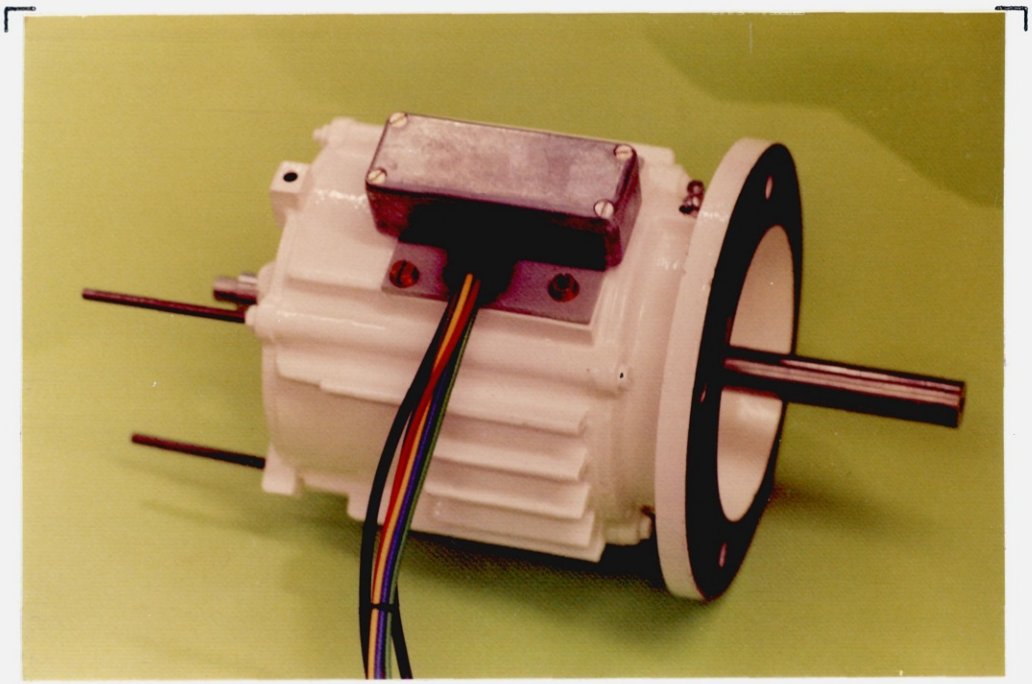


FIG. 6.45. PHOTOGRAPH OF THE FINISHED SQUARE WAVE SYNCHRONOUS MOTOR

6.8 Motor Test Rig

Having built the motor there was a need for a dynamometer capable of loading the motor shaft at speeds of up to 30000 rpm. No such dynamometer was available but it was thought that an eddy current brake type of unit would be

suitable. Several magnet and disc configurations were considered. It was finally decided to use the stator from a disc rotor central heating pump to induce eddy currents into a dural disc mounted on the motor shaft. It was not known whether the unit would be able to load the motor sufficiently to obtain a full set of torque-speed curves. However, minimum effort was expended on the dynamometer design due to the limited amount of time available, and the fact that the likely motor performance could not be predicted with any certainty.

The stator was taken from an SMC Commander 'S' motor. It has two phase windings and is normally operated from a single phase supply by means of a capacitor connected in series with one phase winding. The rotating field produced by the stator was useful in the required application because the test motor could be driven, thus allowing the open circuit phase emf's to be observed. It also enabled the position sensor circuitry to be checked prior to the square wave motor and inverter being run together.

It is not proposed to describe the dynamometer in detail. The motor mounting bracket and the dynamometer assembly layout are illustrated in figures 6.46 and 6.47. The torque was measured by means of a torque arm acting on a 32 ounce (8.92N) load cell (Pye Ether type UF1). The maximum theoretical motor torque with 14.5 turns per phase winding is 0.646Nm, as shown in section 6.6.7. Two attachment points for the load cell were provided on the torque arm at radii of 10cm and 20cm, thus allowing torques up to 0.89Nm and 1.78Nm respectively to be measured.

The two phase stator of the eddy current brake was energised from a variac with an 8μF capacitor (chosen to give equal phase currents) connected in series with the "capacitor" coil. The windings are rated at about 0.53 amp rms each but it was found that they could be operated at up to twice that current for short periods without serious overheating. An applied voltage of about 110 volts a.c.

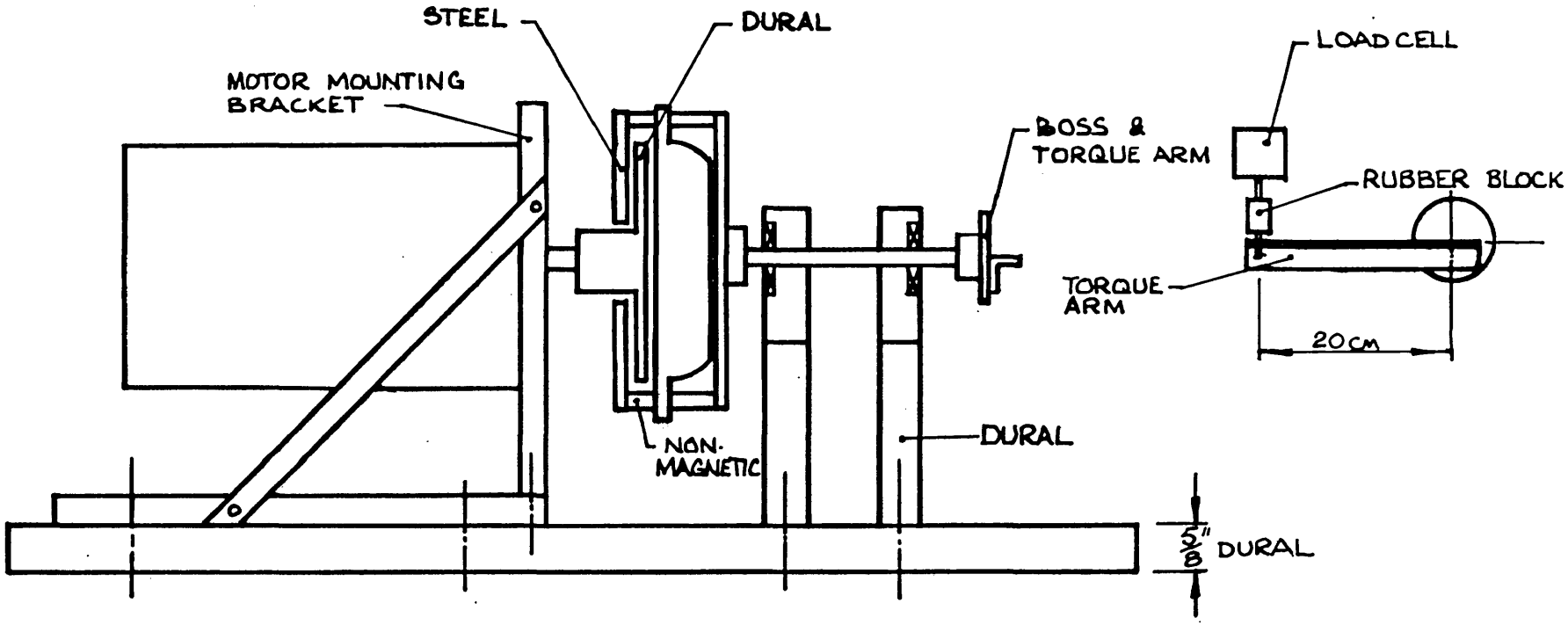


FIG. 6·46 MOTOR MOUNTING BRACKET & DYNAMOMETER ASSEMBLY LAYOUT

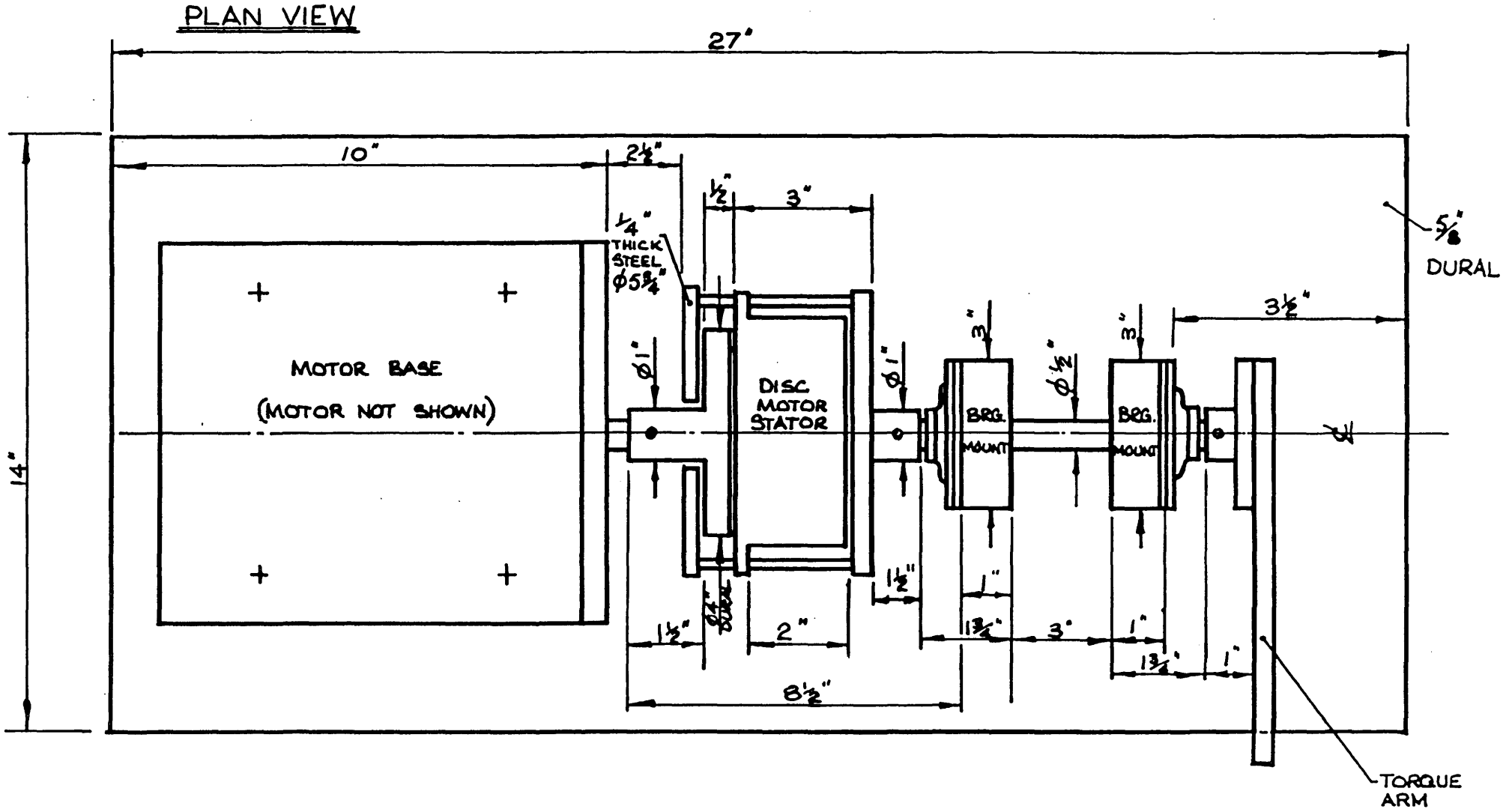


FIG. 6·47 MOTOR BASE PLATE & DYNAMOMETER ASSEMBLY LAYOUT

was required to achieve winding currents of 0.5 amp rms. A photograph of the motor test rig is shown in fig. 6.48.

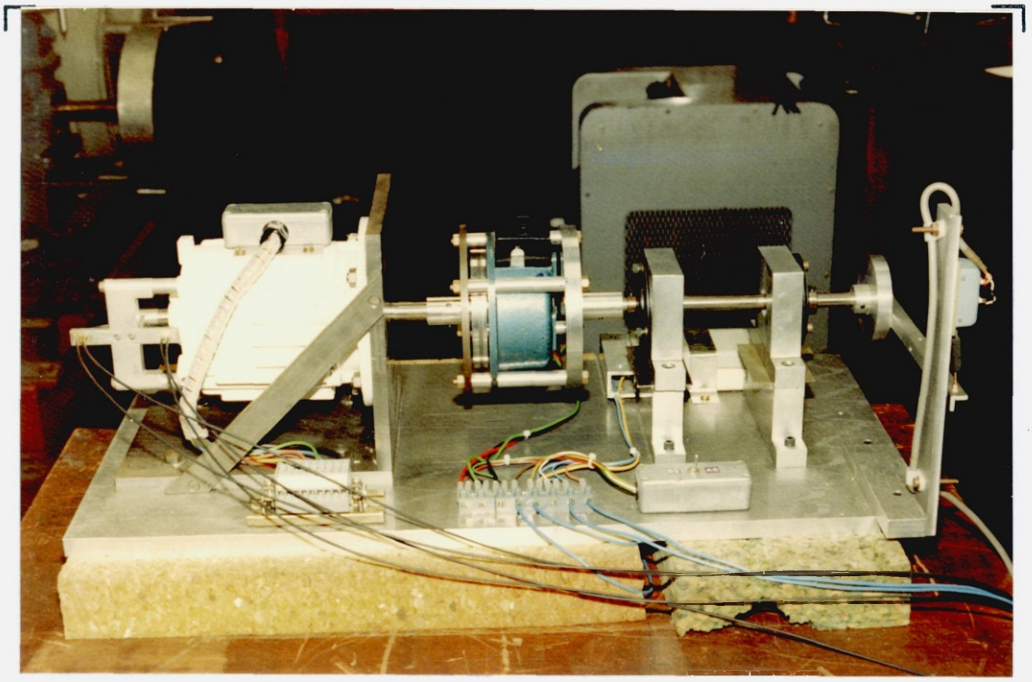


FIG.6.48. PHOTOGRAPH OF THE SQUARE WAVE MOTOR TEST RIG (SHOWING THE EDDY CURRENT BRAKE DYNAMOMETER)

6.9 Motor Parameter Checks

Prior to the main motor and inverter performance tests some simple measurements were made to determine the phase winding resistances and inductances, the cogging torque, and the rotor airgap flux.

6.9.1 Phase Winding Resistance and Inductance

The measurements were made with the standard volt-amp technique. The results are tabulated in fig. 6.49. The inductance values were obtained with a 1 amp 1kHz sinusoidal current passing through the winding under test. The phase resistance of 0.101Ω is slightly larger than the value of 0.083Ω predicted in section 6.7.2. The agreement is reasonable since the predicted value takes no account of the resistance of any joints or connecting leads. The measured phase inductance of 99μH is about 50% more than the value

of 67μH predicted in section 6.6.8, but this can easily be accounted for by the end winding leakage inductance which is difficult to calculate accurately.

PHASE NUMBER	WIRE COLOUR	RESISTANCE	INDUCTANCE
1	brown	0.101 Ω	98μH
2	red	0.101 Ω	99μH
3	orange	0.103 Ω	99μH
4	yellow	0.102 Ω	98μH
5	green	0.104 Ω	100μH
6	blue	0.102 Ω	99μH
7	violet	0.104 Ω	100μH

FIG. 6.49. TABLE OF RESISTANCE AND INDUCTANCE VALUES FOR THE PHASE WINDINGS OF THE SEVEN PHASE SQUARE WAVE SYNCHRONOUS MOTOR

6.9.2 Motor Cogging Torque

The cogging torque was measured by wrapping a piece of linen tape around the motor shaft and pulling on the tape with a spring balance. The force required to just overcome the cogging torque was noted. The test was repeated ten times and the averaged spring balance reading was 31.88N. The motor shaft radius is 8.485mm, and so the motor cogging torque is 0.271Nm.

6.9.3 Rotor Airgap Flux Density

The eddy current brake was found to be capable of driving the test motor around providing the rotor was flicked to overcome the cogging torque. With the rotor spinning at 1182 rpm, the generated voltages on each phase were measured by a moving coil a.c. volt meter (Avometer). It was found that each phase had an average back-emf E_{av} of 1.25 volts; i.e. E_{av} = 1.06 volts/1000 rpm. For a quasi-square voltage waveform, the average magnitude E_{av} is related to the peak

magnitude E_{ph} by the ratio of the generated voltage to zero voltage periods over a cycle. For the back-emf waveform considered here:

$$E_{ph} = \frac{7}{6} E_{av} \qquad 6.79$$

i.e. $E_{ph} = 1.236$ volts/1000 rpm

The airgap flux density is then calculated by using equation 6.66. Substitution of the appropriate values gave a value of 0.298T for the airgap flux density B_{ag}, which is well below the expected value of 0.4347T. This may be due to:

(a) the magnets not being fully magnetised when purchased;

(b) heating of the magnets during machining;

(c) overheating of the magnets during the curing of the carbon fibre;

(d) slotting effects not accounted for by Carters coefficient; (in fact Carters coefficient may not be stricly relevant in an application with permanent magnets).

The lower than planned airgap flux density does mean that the motor output power and torque per amp (already cut by the incorrect choice of the number of phase winding turns) are further reduced. Nevertheless it was decided to continue the project with the "downrated" motor.

As explained in Chapter 5, it was not possible to produce an inverter rated at ±80 volts per phase. (This was essentially because a suitable power supply was not available.) The actual inverter was rated at ±60 volts 7 amps per phase, and so had the motor been correctly designed it would not have been able to achieve a speed of 30000 rpm.

However, the incorrectly chosen number of phase turns and the lower than expected airgap flux density, are such that 30000 rpm can be achieved with less than 60 volts applied to the phase windings.

The use of a phase current with a magnitude of 7 amps is just over the conductor rating since the rms value of the current is 6.48 amps (from equation 6.70). However, this was not thought likely to lead to a winding failure. The total stator copper loss calculated by equation 6.78 for I_{ph} = 7 amps and R_{ph} = 0.101Ω is 29.7 watts, and this should not cause an excessive temperature rise. The maximum torque of the motor with peak phase currents of 7 amps is found to be 0.495Nm by equation 6.67, and so the maximum motor power rating at 30000 rpm is 1.556kW. This is a 48% derating on the power output that could have been obtained from the motor with B_{ag} = 0.4347T, N_{ph} = 21.5, and I_{ph} = 6.25 amps.

The motor has a back-emf constant K_b of 11.8mV per radian (1.236 volts per 1000 rpm). If the winding impedance is ignored, the motor requires a minimum applied phase voltage of 37.08 volts to achieve a speed of 30000 rpm.

The back-emf waveform was extremely good and an example showing the essentially square form is shown in fig. 6.50. The ripple in the waveform is caused by the open slots in the stator. The rise and fall times of the waveform are both about 5% of the cycle time; i.e. just less than the period to rotate one slot pitch (one slot pitch = 7% of cycle time). The fact that the rise and fall times are not instantaneous leads to a requirement for inverter protection components to prevent large current spikes. These components are described in Chapter 5.

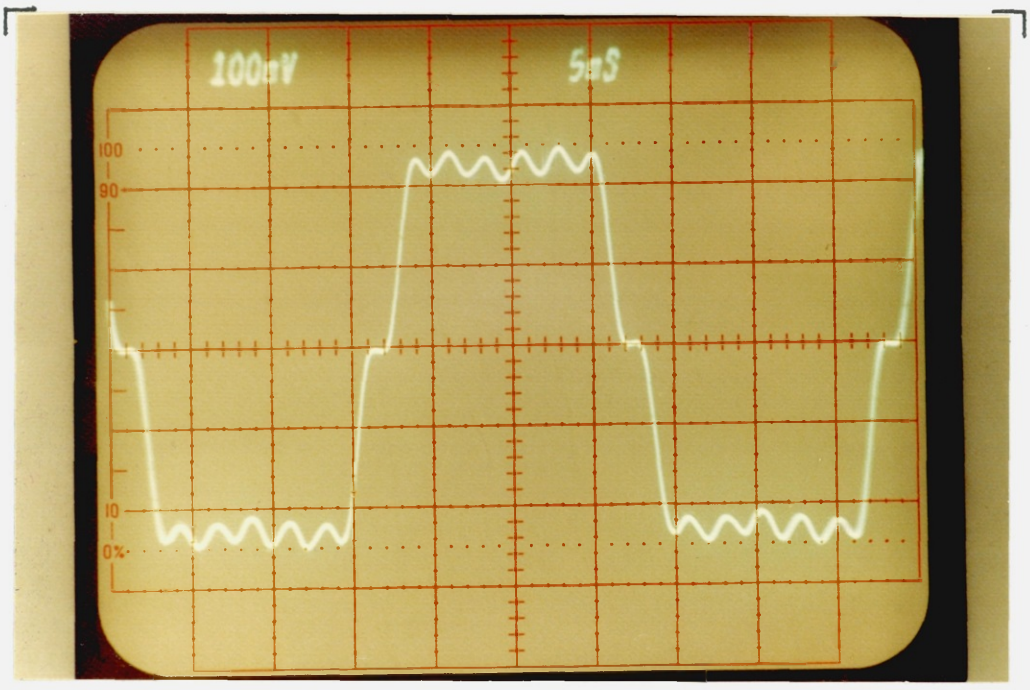


FIG.6.50. TYPICAL OPEN-CIRCUIT PHASE WINDING BACK-EMF WAVEFORM (MOTOR DRIVEN AT 1900 RPM) — 1·0V/cm — 5ms/cm

6.9.4 Summary of Motor Parameters

The assembled motor has the following parameters:

Maximum rotor speed:	30000 rpm
Number of turns per phase:	14.5
Measured phase resistance:	0.101 Ω
Measured phase inductance:	99μH
Back-emf constant:	11.8mV per rad/s
Calculated Copper loss at 7 amp phase current:	29.7 W
Calculated maximum torque at 7 amp phase current:	0.495Nm
Calculated power output at 30000 rpm:	1.556kW
Measured cogging torque:	0.271Nm

6.10 Rotor Position Sensor Unit

The 7 phase motor requires a rotor position sensor to enable it to be used in an autopiloted drive system. The various types of position sensor that can be used in such an application are discussed briefly in Chapter 3. Optical sensors are concluded to be the most suitable on the grounds of their high resolution and accuracy and the ease of implementing a position sensing system with them. It is further concluded that two position signals are necessary to enable the rotor position to be accurately and reliably known. These signals are the synchronising signal (SYNC') occurring once every revolution, and the segment signal (SEG') occurring at regular intervals during each rotation.

In order to autopilot the seven phase motor, the SYNC' and SEG' signals are used as interrupts by the microprocessor system described in Chapter 4 to control the MOSFET inverter described in Chapter 5. Therefore, the signals must be of a suitable form for use by the microprocessor, and they must not occur at a rate that is too fast for the microprocessor to handle. As there are several SEG' pulses between each SYNC' pulse it is more likely that the SEG' signal will place a limit on the rotational speed of the motor. The number of SEG' pulses per revolution can be chosen if the resolution required from the position sensing system is known. However, the optical position sensing system will be described in full before the choice of the SEG' pulse number is explained.

6.10.1 The Basic System

A block diagram of the position sensor system is shown in fig. 6.51. The system is basically comprised of:

(a) a perspex disc with the required SEG' and SYNC' pulses marked on its surface. The disc is mounted on one end of the motor shaft;

(b) a light source to provide light suitable for transmission through the SEG' and SYNC' patterns on the perspex disc;

(c) a pair of photo-sensitive amplifiers to receive the light passing through the patterns on the perspex disc. These amplifiers convert the incoming light pulses into electrical signals suitable for the microprocessor;

(d) optical fibres which transmit light from the light source to one side of the perspex disc, and carry the light received on the other side of the disc to the photo-sensitive amplifiers.

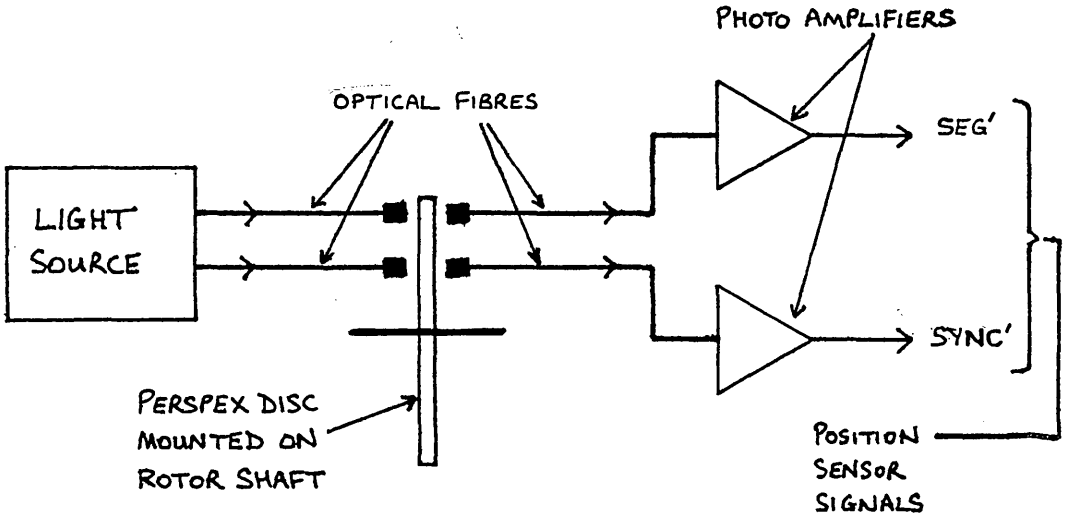


FIG.6.51. BLOCK DIAGRAM OF THE ROTOR POSITION SENSOR SYSTEM

Before describing the system in any further detail it is worthwhile explaining why optical fibres were used. The benefits of using optical fibres include:

(a) the fact that the position sensing electronics can be removed from the motor and housed in a separate place. This is useful if the motor has to operate in a remote and/or confined location,

since the optical sensors can be positioned in an easily accessible position, thus facilitating maintainance;

(b) the reduction of electrical interference caused by the rapidly switched currents in the motor windings. A great deal of spurious noise can be eliminated by keeping the position sensing electronics well away from the motor;

(c) an increase in position sensor reliability due to the fact that the sensors are isolated from the temperature and vibration of the motor;

(d) the fact that the position sensing electronics can be located next to the microprocessor system. Hence the power supply leads supplying the position sensor unit can be of minimal length;

(e) the electrical isolation of the sensor unit from the motor assembly. This is extremely useful in a system containing a microprocessor as it removes the possibility of the high voltages on the motor windings being connected via a short circuit to the microprocessor.

Against these points must be set:

1. the extra cost of both the fibre (actually very little) and the opto-coupling components (significant) in relation to conventional screened electrical leads. (Note that a zero interference electrical sensor system would not be particularly cheap);

2. the fact that there is a minimum radius that the fibre can be bent around. This slightly restricts the mechanical design of certain parts of the system.

6.10.2 Practical Details of the Position Sensor System

The constructional details of the system are summarised below. The system was run from the ±12 volt microprocessor supplies.

6.10.2.1 The Optical Fibre

The optical fibre was obtained from RS Components Ltd. It has a single-strand polymer optical core sheathed in black polyethylene. The flexible "cable" has a minimum bend radius of 15mm, a tensile strength of 5kg, and costs about 50 pence per metre. The optical core has a diameter of 1mm and an attenuation of not more than 1dB per metre. It can be cut with a scalpel and easily joined using readily available terminations.

In order to achieve good separation of the motor from the optical sensor electronics, it was decided that the pieces of fibre from the light source to the perspex disc and from the disc back to the photo amplifiers should not be less than 2.0m long. The pieces of fibre were actually made 2.20m long and so the total fibre length in both the SEG' and SYNC' light channels was 4.40m.

6.10.2.2 The Light Source

It was originally hoped to use either an infra red or visible red light emitting diode (LED) as the light source. The light source had to be of sufficient intensity to transmit light down 2.20 metres of optical fibre, across a reasonable airgap with a perspex disc in it, and finally along a further 2.20 metres of optical fibre to the photo amplifiers. Further loss of intensity had to be allowed for in the joints used for the optical fibre. It was found that the intensity of even large LED's was inadequate to overcome the losses inherent in the system; (the dominant loss being due to the airgaps at the perspex disc), and so a 12 volt 2.2 watt filiament bulb was used. The bulb was located in

an aluminium block and supplied light to both the SEG' and SYNC' optical fibre channels. The cores of the two fibres passed into the block via 1mm diameter holes which were aimed directly at the bulb's filiament. This maximised the light entering the fibres without the need to use lenses. The block was contained in a diecast box as shown in fig. 6.52. The optical fibres were brought out of the box via bulkhead connectors.

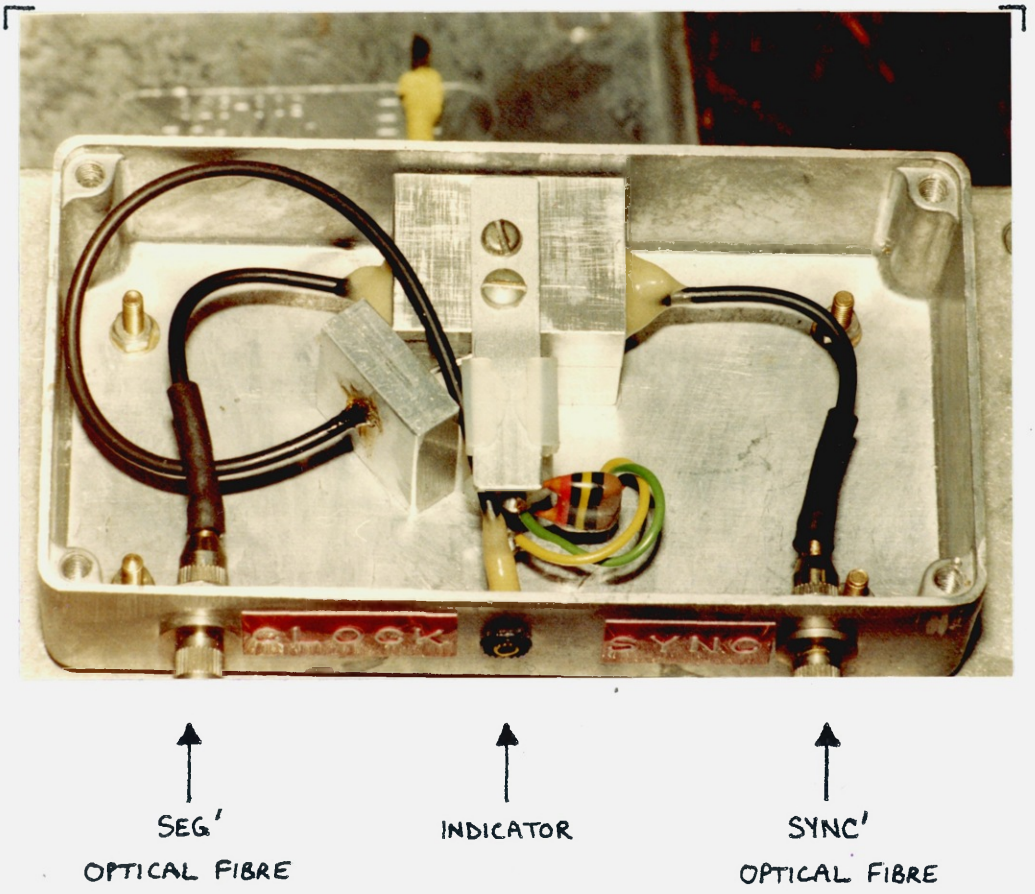


FIG.6.52. THE OPTICAL POSITION SENSOR LIGHT SOURCE

A piece of fibre was used to pick up stray light from the bulb. It was routed to the outside of the diecast box where it acted as an indicator to show if the bulb was working.

6.10.2.3 The Perspex Disc and SEG'/SYNC' Pattern

It was found by experiment that the chosen light source had sufficient intensity to produce a detectable signal with

an airgap of up to about 5mm inserted between the two lengths of optical fibre. The light received at the end of the fibre was actually increased when a sheet of perspex was placed in the airgap, because the perspex acts as a crude lens and reduces the light dispersal across the gap. It was decided to use 1.60mm (1/16") thick perspex for the disc and to limit its diameter to 50mm to reduce the risk of it shattering at 30000 rpm. A 3mm wide airgap was then chosen and this allowed reasonable clearance on both sides of the perspex disc. The optical fibres were supported on either side of the disc by passing them through holes drilled in an aluminium assembly of the form shown in fig. 6.53.

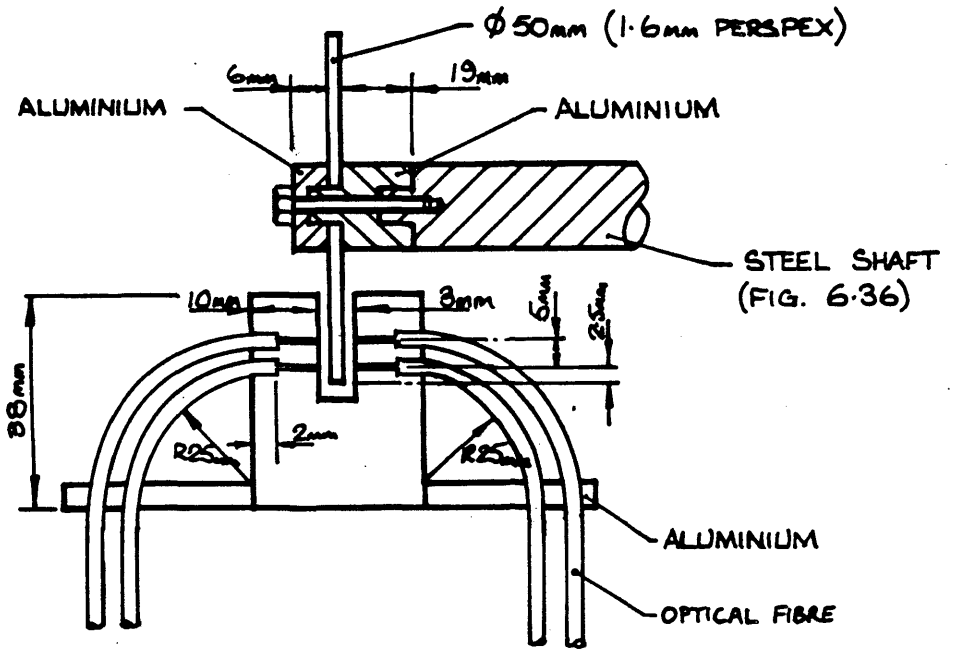


FIG. 6·53 THE ROTOR POSITION SENSOR FIBRE SUPPORT ASSEMBLY

The optical fibres were held in place on the aluminium supports by an adhesive suitable for use on the polyethylene sheath of the fibre. The arrangement for holding the disc onto the end of the rotor shaft is also shown in fig. 6.53.

The pattern of SEG' and SYNC' lines must be arranged as described in Chapter 3 to ensure that the SYNC' line is positioned for correct operation in both forward and reverse directions. The number of segment lines has to be a multiple of 7 in order to be suitable for the 7 phase motor system. It is not easy to geometrically sub-divide 360° into angles that are basically sub-multiples of 360/7°. It was found that a satisfactory result could not be produced by hand. Therefore, the graphics facilities available on the college CDC computer were used to produce a scaled up version of the required pattern on a flat-bed plotter. Any number of SEG' lines could be drawn around the periphery of the circle because the control software for the plotter produced the required angular increments. A selection of patterns with various numbers of SEG' lines were produced. The finished drawings were 12 inches (304.8mm) in diameter and were in black ink on white paper. The drawings were photographed onto 120 format transparencies with the image size adjusted to be 50mm in diameter. The transparencies were glued onto 50mm diameter perspex discs using cyanoacrylate adhesive and the surplus transparency material was trimmed off using a sharp knife. A view of the disc used in the system is shown in fig. 6.54.

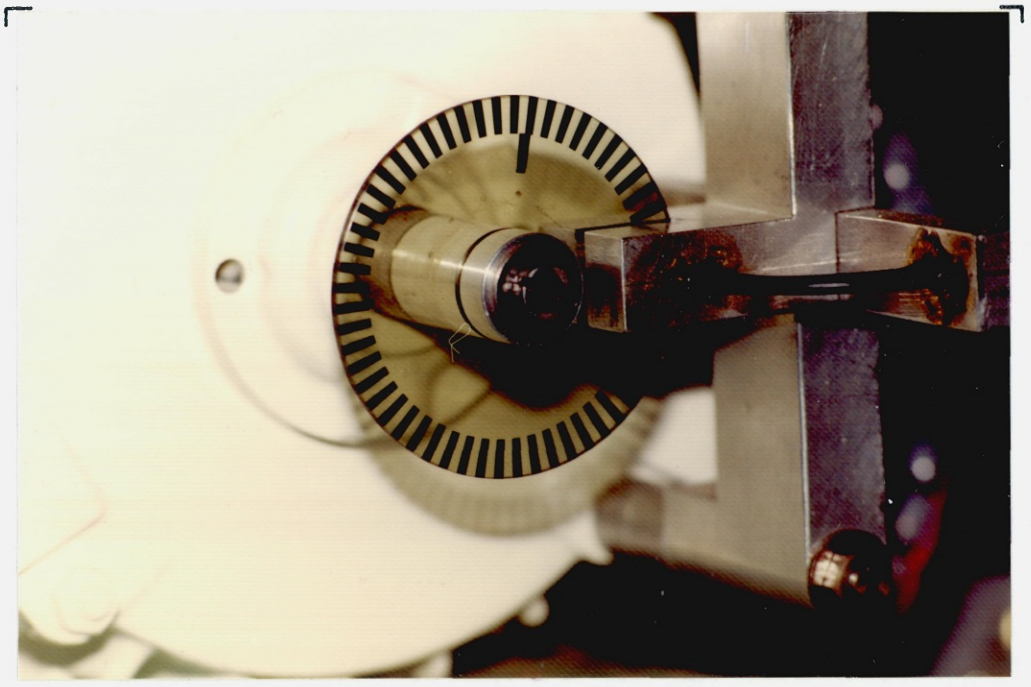


FIG. 6.54. PHOTOGRAPH OF THE OPTICAL POSITION SENSOR DISC

The disc has 56 segment lines. The aluminium support for the optical fibres can also be seen in fig. 6.54. The method used to choose the number of segment lines is explained in section 6.10.2.6.

6.10.2.4 The Photo Amplifiers

The intensity of light available at the end of the SEG' and SYNC' optical fibres was very small and so very sensitive photo amplifiers were needed. The basic arrangement of one of the photo amplifiers is shown in fig. 6.55.

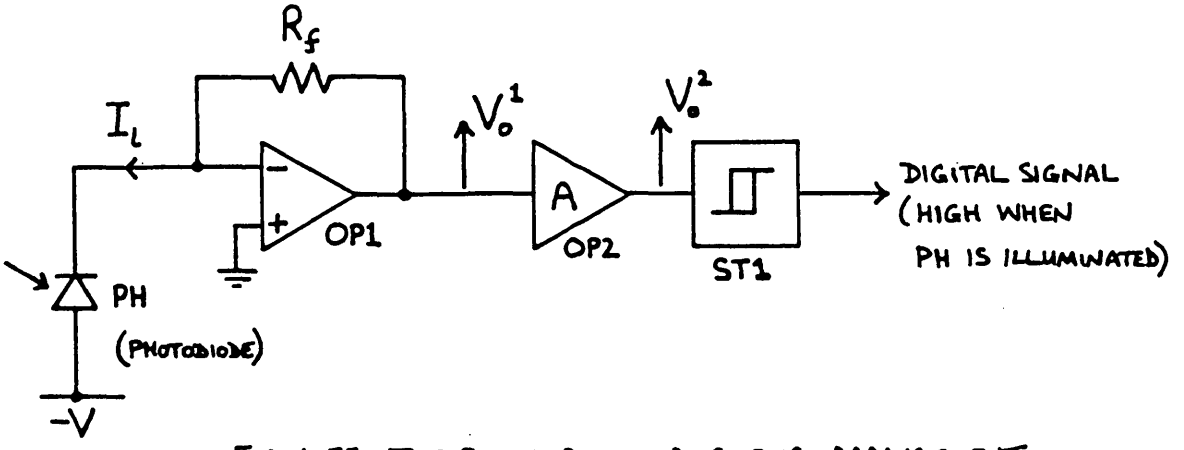
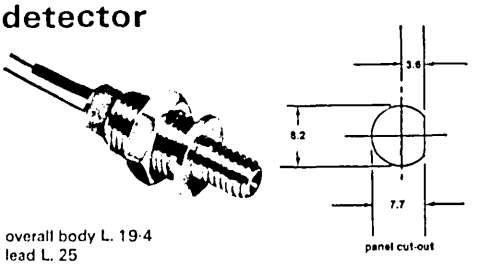


FIG. 6.55. THE BASIC PHOTO AMPLIFIER ARRANGEMENT

A reverse biassed photo diode PH is connected to the virtual earth of a high input impedance operational amplifier, OP1. The diode leakage current I_1 is proportional to the light falling onto it, and so the voltage output V_o^1 of OP1 is proportional to the incident light. The voltage is amplified by the operational amplifier OP2, and the resulting signal V_o^2 is converted into digital form by a schmitt trigger ST1. In addition to being very sensitive, the photo amplifier must have a frequency response sufficient to cope with the SEG' pulse rate.

The photo diodes used for the photo amplifiers were also obtained from RS Components. Each diode is housed in an optically aligned body and the optical fibre is connected onto the body by use of an end termination. The diode characteristics are summarised in fig. 6.56.

detector



overall body L. 19·4
lead L. 25
nut 10 A/F

A planar, silicon PIN photodiode housed and optically aligned in an identical body to the emitter 309-290. The diode features low junction capacitance, fast response and a high cut-off frequency. Red sleeved lead indicates anode and that the device is a detector.

technical specification

absolute max. ratings	
Reverse voltage, V_R	50 V
Operating temperature range	−55 °C to +100 °C
Power dissipation at 25 °C, Pd	250 mW

opto-electronic characteristics at 25 °C	
Wavelength of peak sensitivity	850 nm
Spectral sensitivity (λ=850 nm)	0·55 A/W
Rise time of photocurrent (R_L=50 Ω, V_R=20 V, λ=850 nm)	1ns typ., 5ns max.
Cut-off frequency (R_L=50 Ω V_R=20 V)	500 MHz
Dark current (V_R=20 V)	5 nA max.
Capacitance, V_R=20 V	3·5 pF
V_R=0 V	15 pF

FIG.6.56. THE TECHNICAL SPECIFICATION FOR THE PHOTO DIODES USED IN THE OPTICAL POSITION DETECTOR SYSTEM.

The very small diode leakage current can only be successfully detected by the use of an operational amplifier with a very high input impedance. The LF351N JFET amplifier was chosen since this has an input impedance of $10^{12}\Omega$. The LF351N has a bandwidth of 150kHz and a slew rate of 13v/μs. Therefore, it is able to deal with a rapid SEG' pulse rate.

It was found that in order to obtain a reasonable signal amplitude from OP1, it was necessary to use a feedback resistor R_f with a value of 47MΩ. The use of a very large feedback resistor on an operational amplifier to achieve a high gain is not without problems. The amplifier is very sensitive to stray electromagnetic noise (such as from the 50Hz mains) and the d.c. output level can vary depending on the capacitive influence of objects close to the amplifier. There is also a tendency for the amplifier to oscillate and there can be ringing on the output waveform. All of these effects were encountered but they were virtually eliminated by mounting PH and OP1 on a small printed circuit board and

placing the board in a diecast metal case. The output voltage V_o^1 was found to be about +4.0 volts when light was passing along the optical fibre and +1.0 volts when the light was blocked. To eliminate the +1.0 volt offset present with zero light, it was decided to a.c. couple the output from OP1 to the input of OP2. The final circuit diagram for the SEG' and SYNC' photo amplifiers is shown in fig. 6.57. The SEG' and SYNC' amplifiers are not identical because the mark/space ratio's of the two signals are so different. The two photo amplifiers will therefore be briefly discussed separately.

The SEG' light signal falling onto PH1 has a 50:50 mark/space ratio and so the output of IC1 varies from +1.0 volt to +4.0 volts as shown in fig. 6.58(a). The output of IC1 is a.c. coupled and level shifted by C1 and R5 so that the input to IC3(a) is centred about a +6.0 volts false ground G_1' as shown in fig. 6.58(b). The time constant was chosen as 1 second so that the pulses could be reliably coupled even at low motor speeds. The false ground G_1' is produced by the potential divider formed by R3, R4, C3 and C4. IC3(a) has a voltage gain of +22 defined by R9 and R10 and it is supplied from the +12 volt and 0 volt rails. Therefore, the input signal to IC3(a) causes the output to be at either 0 or +12 volts as shown in fig. 6.58(c). (In practice the levels are about +11.5 volts and 0.5 volts because of the voltage drops within the operational amplifier circuitry.) Finally, the output of IC3(a) is fed into the non-inverting schmitt trigger formed by IC4(b) and IC4(c). The output of IC4(c) is $\overline{\text{SEG'}}$ and it is shown in fig. 6.58(d). A CMOS CD4093 integrated circuit was used for the schmitt function in preference to a TTL schmitt circuit in order to eliminate the need for a +5 volt supply rail in the photo amplifier unit.

The SYNC' light signal falling onto PH2 has a mark/space ratio of about 75:1 as indicated in fig. 6.59(a). The light is only blocked for a very small proportion of each revolution. The output of IC2 is a.c. coupled and level

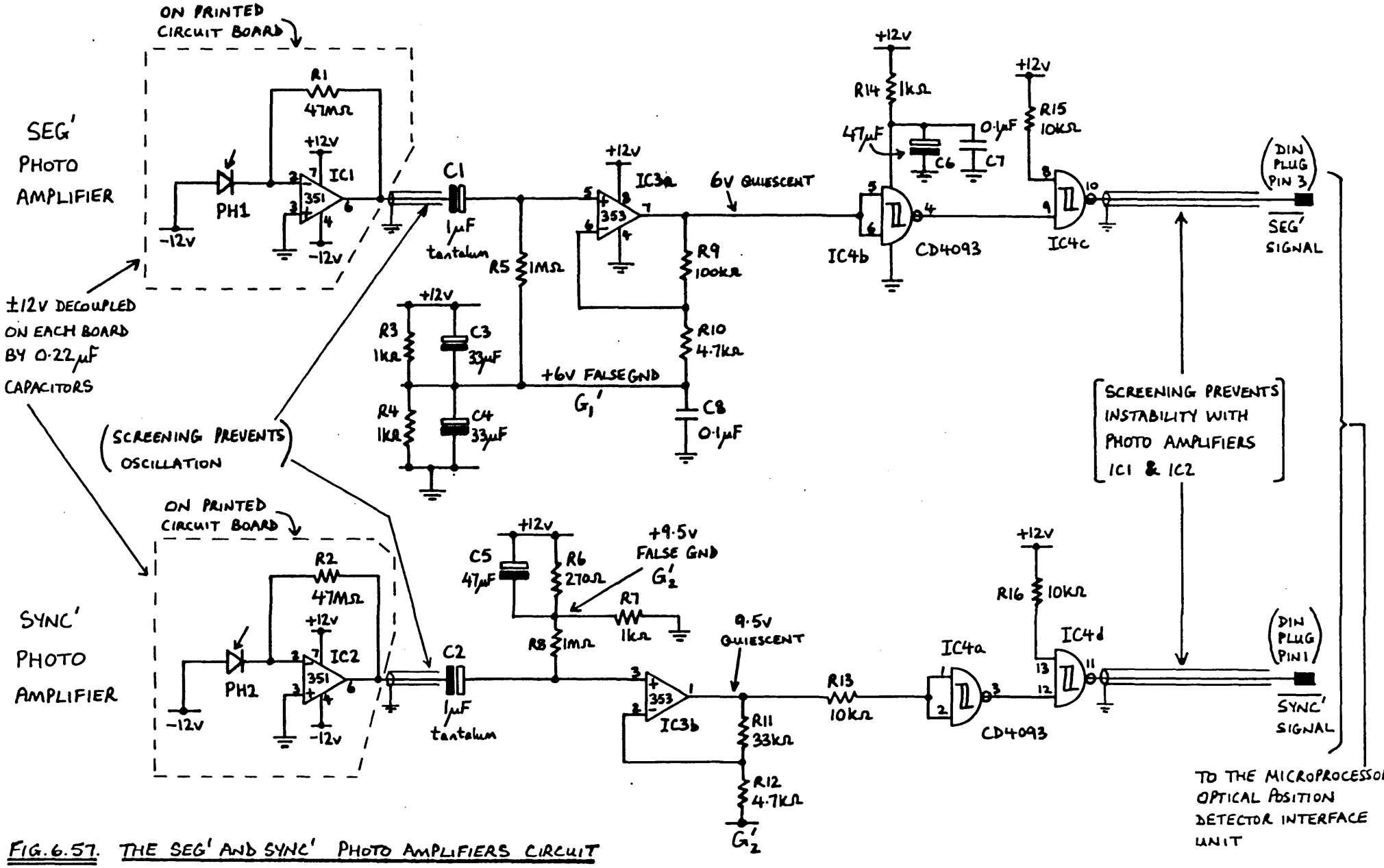


FIG. 6.57. THE SEG' AND SYNC' PHOTO AMPLIFIERS CIRCUIT

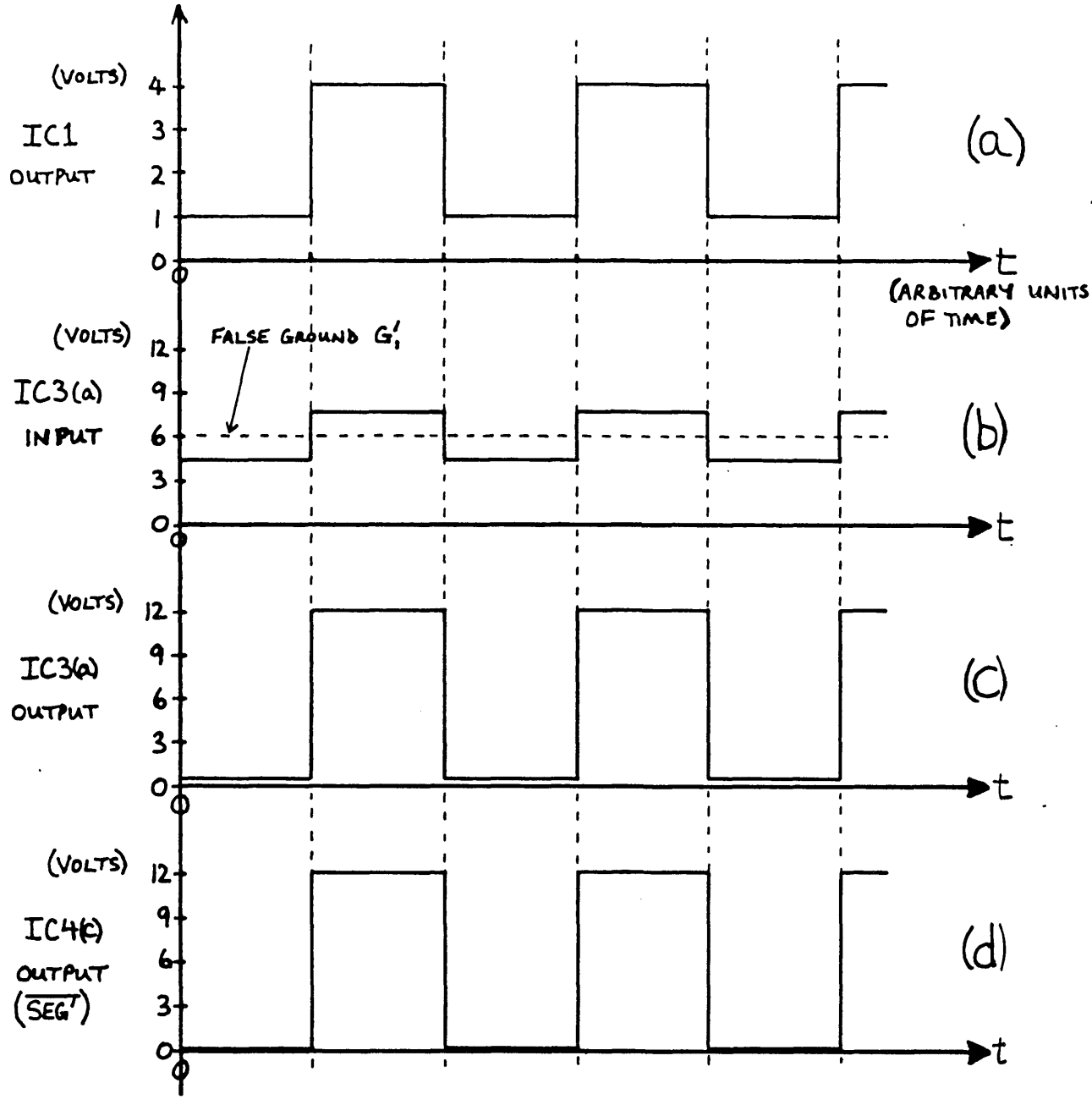


FIG. 6.58. VOLTAGE WAVEFORMS AT VARIOUS POINTS OF THE SEG' PHOTO AMPLIFIER SHOWN IN FIG. 6.57.

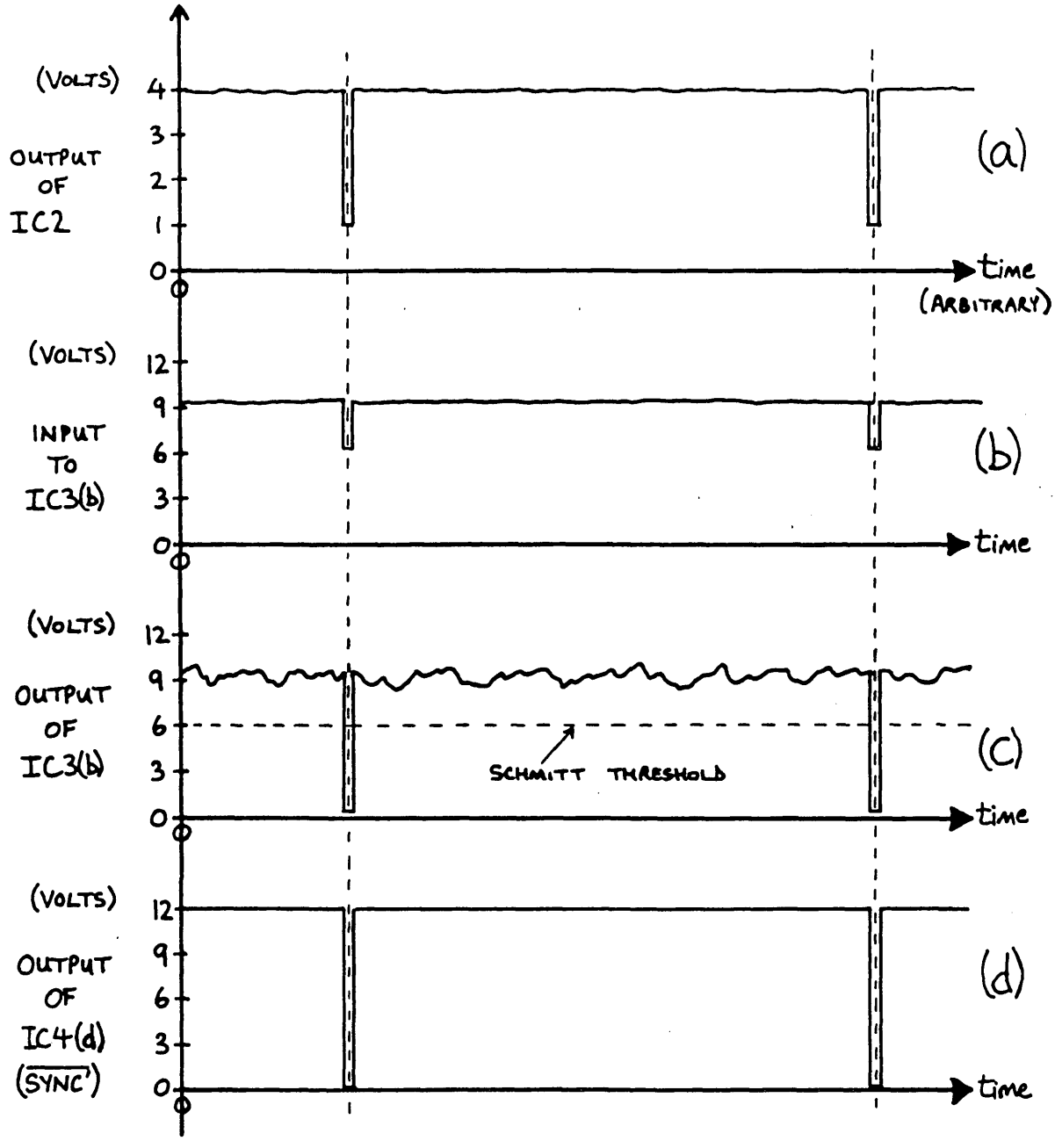


FIG. 6.59. VOLTAGE WAVEFORMS AT VARIOUS POINTS OF THE SYNC' PHOTO AMPLIFIER SHOWN IN FIG. 6.57.

shifted by C2 and R8 so that the input to IC3(b) is centred about a +9.5 volts false ground G_2'. The false ground is produced by the potential divider formed by R6, R7 and C5. For the majority of each cycle the output of IC3(b) is at a potential of about +9.5 volts as shown in fig. 6.59(c), but whenever the input drops by 3 volts the output goes down to about 0 volts. The noise shown in fig. 6.59(c) is the amplified effects of dirt and imperfections on the perspex disc. To minimise the chance of a spurious SYNC' pulse being generated, it was necessary to ensure that the noise waveform could not fall below the schmitt threshold of +6.0 volts. This was achieved by making the voltage gain of IC3(b) much smaller than that used for IC3(a). Feedback resistors R11 and R12 were chosen to give IC3(b) a voltage gain of 8. The output of IC3(b) is fed to the non-inverting schmitt formed by IC4(a) and IC4(d) and the resulting $\overline{\text{SYNC}}$' signal is shown in fig. 6.59(d).

The elimination of spurious oscillations in the photo amplifiers was a time consuming task but it was eventually achieved by the addition of several components and careful circuit layout:

(a) resistor R13 was added between IC3(b) and IC4(a). A value of 10kΩ was selected for R13 during the prototype tests;

(b) the +12 volt power supply to the CMOS schmitt gates was decoupled by a resistor and capacitor network (R14, C6, C7);

(c) screened leads were used for certain connections as indicated on the circuit diagram of fig. 6.57. This was a major factor in the removal of ringing on the edges of the $\overline{\text{SEG}}$' pulses;

(d) IC1 and IC2 along with their associated photo diodes were mounted on separate printed circuit boards which were located close to the bottom of

a diecast metal box. The remainder of the circuit (IC3 and IC4) was mounted on a circuit board located on one side of the diecast box. An aluminium screening "wall" separated the circuit board on the box wall from the boards on the bottom of the box;

(e) an absolute minimum of unnecessary conductor track was left on the printed circuit boards;

(f) the 47MΩ feedback resistors were mounted vertically rather than horizontally. This reduced the capacitance associated with them;

(g) a central star point was used for the 0 volts of the three circuit boards;

(h) all signal leads were carefully routed to minimise ringing. The screened leads from IC1 to IC3(a) and from IC2 to IC3(b) had to be kept well away from the schmitt trigger integrated circuit to prevent oscillations on the edges of the $\overline{\text{SEG'}}$ pulses.

The $\overline{\text{SEG'}}$ and $\overline{\text{SYNC'}}$ waveforms obtained from the finished photo amplifiers were clean and stable. The frequency response of the SEG' channel was tested by illuminating the photo diode PH1 by a LED driven from a square wave oscillator. The circuit was found to respond up to a frequency in excess of 30kHz, and so it was suitable for use at 30000 rpm with a perspex disc having 56 SEG' stripes on it. A photograph of the circuit boards mounted inside the diecast metal box is shown in fig. 6.60. The power supply and signal connections were connected to the unit via din plugs and sockets, and the optical fibres were brought into the box via bulkhead connectors.

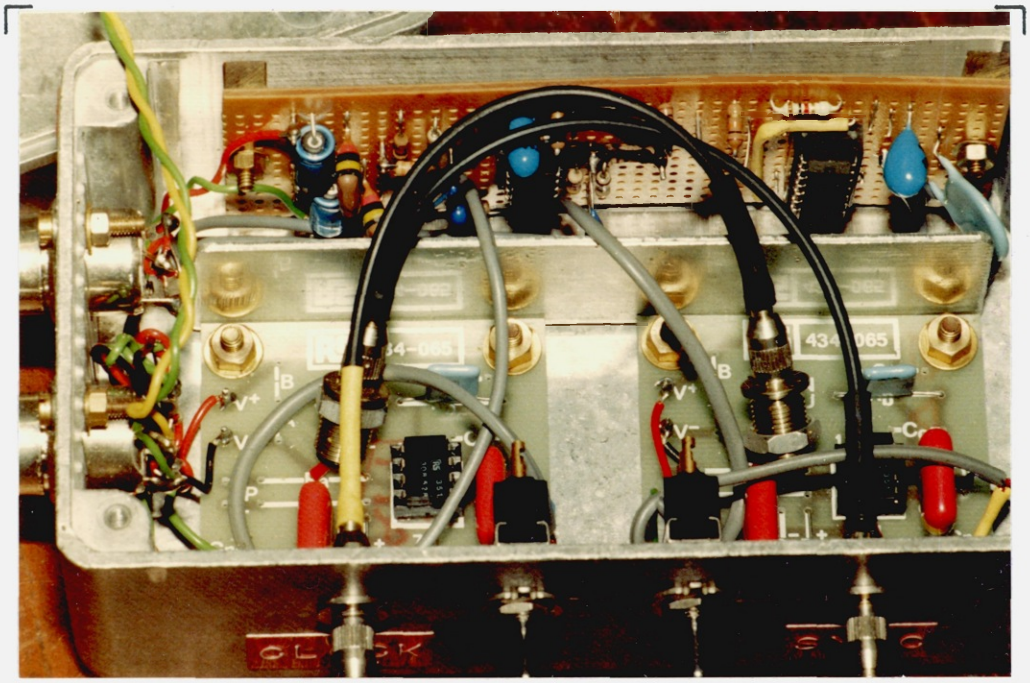


FIG. 6.60. PHOTOGRAPH OF THE OPTICAL POSITION DETECTOR PHOTO AMPLIFIER CIRCUITS.

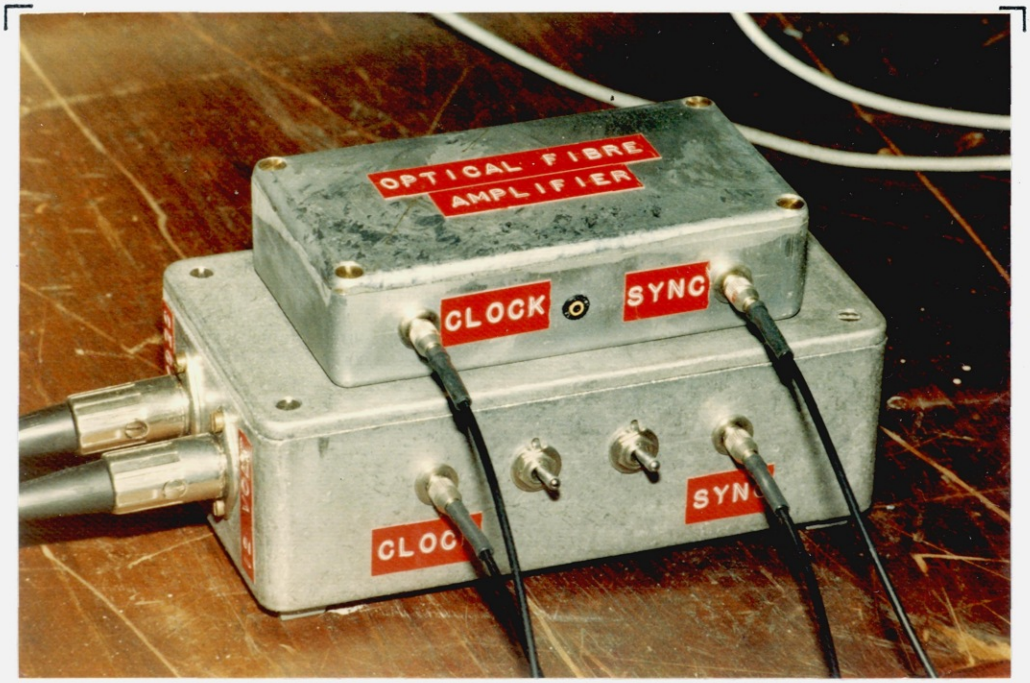


FIG. 6.61. PHOTOGRAPH OF THE COMPLETED PHOTO AMPLIFIER UNIT WITH THE LIGHT SOURCE BOX MOUNTED ON TOP.

6.10.2.5 Final Assembly of Photo Amplifier/Light Source Unit

The complete photo amplifier unit with the light source box mounted on top of the photo amplifier box is shown in the photograph of fig. 6.61. The output signals produced by the unit are the complements of those required by the microprocessor interrupt interface circuits and they are also not TTL logic levels. It was thought best to invert and level shift the signals on the microprocessor interface board because the noise immunity of +12 volt CMOS signals is much better than that of TTL signals. The larger noise immunity reduces the possibility of corruption of the signals on the leads between the photo amplifier unit and the microprocessor interface board.

6.10.2.6 The Number of SEG' Pulses per Revolution and Aspects of System Operation

In a square wave motor with minimal phase inductance there is no need for any displacement between the applied phase voltage V_{ph} and the back-emf E_{ph}. This is because the phase current I_{ph} is simply dependant on the phase resistance R_{ph} and the potential difference between V_{ph} and E_{ph}. The idealised waveforms are shown in fig. 6.62. However, in a motor with appreciable inductance as well as resistance in the phase windings, the phase current has finite rise and fall times as shown in fig. 6.63. The resulting reduction in the rms phase current leads to a fall in the average torque produced by the phase. The effect of the inductance can be minimised by advancing V_{ph} with respect to E_{ph} as shown in fig. 6.64. The large potential differences between V_{ph} and E_{ph} at the switch on and off points cause the current to rise and decay much more quickly, and the current waveform is much closer to the desired shape. As mentioned in Chapter 3, the amount of shift required between V_{ph} and E_{ph} obviously depends on the relative magnitudes of the phase inductance and resistance, and also on the frequency at which the phase winding is operating.

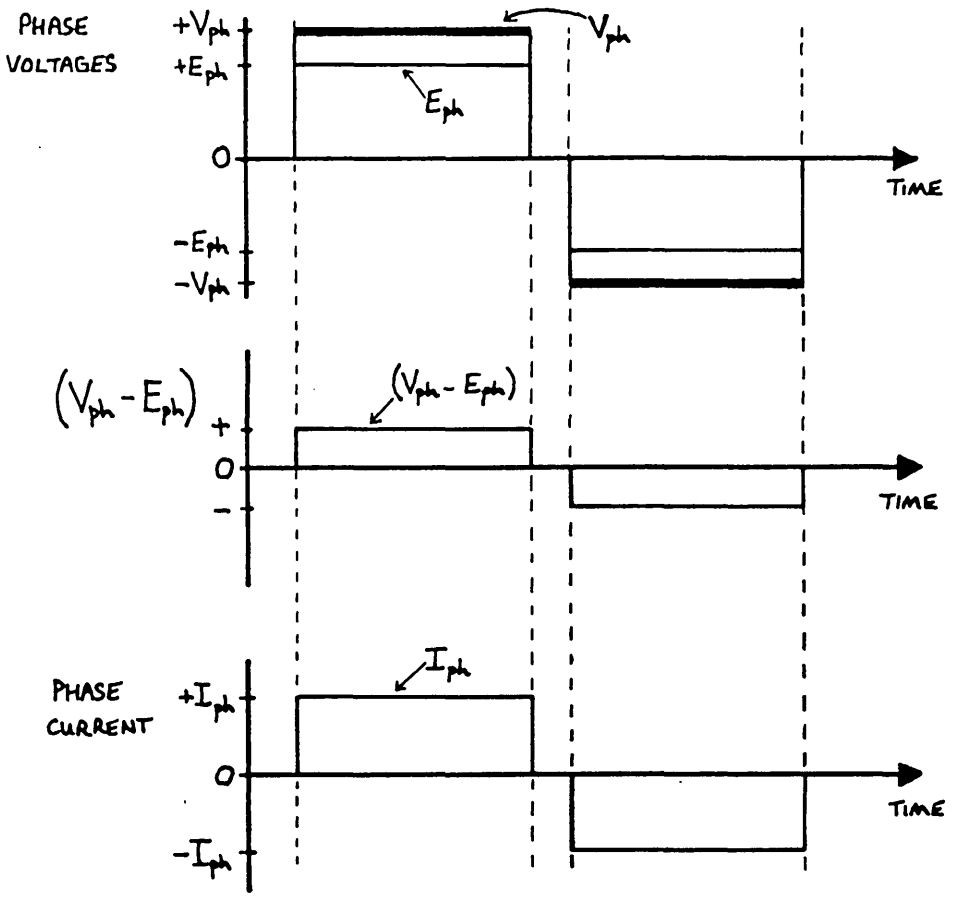


FIG. 6.62. THE IDEALISED PHASE WAVEFORMS FOR A SQUARE WAVE SYNCHRONOUS MOTOR WITH MINIMAL PHASE INDUCTANCE

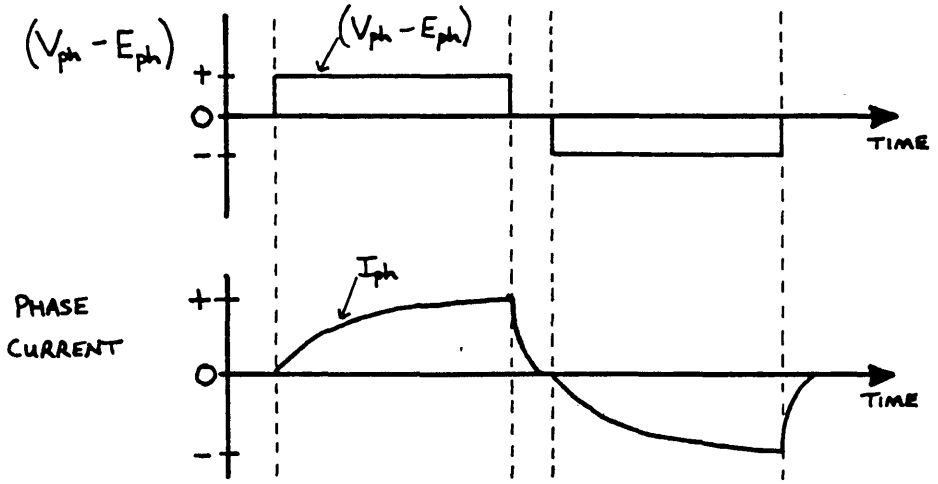


FIG. 6.63. TYPICAL PHASE CURRENT WAVEFORM SHAPE FOR A SQUARE WAVE MOTOR WHOSE PHASE WINDINGS POSSESS APPRECIABLE INDUCTANCE AS WELL AS RESISTANCE.

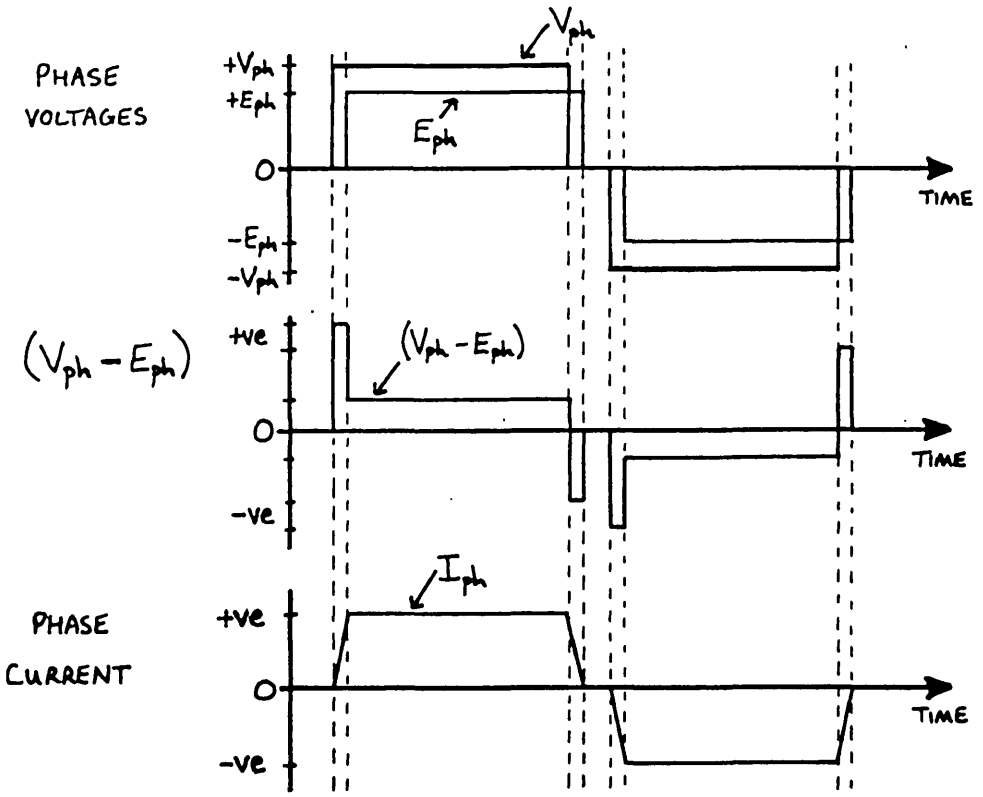


FIG. 6.64. THE METHOD OF ADVANCING V_{ph} WITH RESPECT TO E_{ph} IN ORDER TO MAINTAIN A REASONABLE "SQUARE" PHASE CURRENT WAVEFORM

(At higher frequencies there is a shorter time per cycle in which to force the current to rise and fall as desired.) Therefore, to ensure a reasonably square current waveform in a motor operating over a variable speed range, the advance angle between V_{ph} and E_{ph} must be adjustable.

The minimum step by which the advance angle can be adjusted depends on the resolution of the position sensor used for the motor. The position sensor resolution is fixed by the number of SEG' pulses per revolution:

(a) The absolute minimum number of SEG' pulses per revolution ($SEG'_{min(de)}$) can be calculated using equation 3.27. For a 2 pole 7 phase motor, $SEG'_{min(de)}$ is found to be 7. When the minimum number of SEG' pulses are used it is necessary to sense both edges of the pulses to obtain sufficient position information. The resolution of a "double-edge" sensing position sensor with the minimum number of SEG' pulses can be calculated by equation 3.31, and for a 7 pulse disc it is 25.71°. This resolution allows the applied phase

voltages to be shifted with respect to the generated back-emfs by multiples of 25.71°.

(b) The maximum number of SEG' pulses per revolution is fixed by the physical constraints of the system. In order to ensure that each SEG' pulse can be reliably detected in the optical fibre system, it is desirable that the change in light intensity should be as large as possible as the SEG' lines move between the transmitting and receiving optical fibres on either side of the perspex disc. Since the optical fibre core is 1mm in diameter, it is possible to totally block the light crossing the airgap by placing a 1mm wide SEG' line between the transmitting and receiving fibres. The amount of light reaching the receiving optical fibre can similarly be maximised by ensuring that the gap between adjacent SEG' lines is at least 1mm. In addition, making the gap width equal to the SEG' line width results in a 50:50 mark/space ratio for the SEG' signal. This mark/space ratio permits "double-edge" sensing and motor reversal to be simply achieved as explained in Chapter 3. The number of 1mm wide arcs that can be accomodated around the periphery of a 50mm diameter perspex disc is about 156; i.e. 78 SEG' lines and 78 spaces. However, the number of SEG' lines on the disc must be a multiple of $SEG'_{min(de)}$. Therefore, for the 7 phase motor, the nearest smaller number that is a multiple of 7 is 77. The resolution that can be obtained by triggering off both edges of a 77 SEG' pulse disc is 2.34° as given by equation 3.29. This resolution allows very fine control over the shift between V_{ph} and E_{ph} but the number of SEG' pulses per revolution is 154. Using this number of SEG' pulses results in an interrupt rate to the microprocessor of 77000 per second at 30000 rpm; i.e. one interrupt every 13μs. The TMS9900 microprocessor control system described in Chapter 4 cannot respond quickly enough to deal with such an interrupt rate since the minimum time required to enter and leave an interrupt service routine is 15μs.

It was therefore necessary to choose a number of SEG' lines somewhere between the minimum value of 7 and the im-

practical upper limit value of 77. It was realised that some shift between V_{ph} and E_{ph} would be necessary because the time constant of the 7 phase motor phase windings is 0.99ms. Hence a period of about 5ms (5 time constants) is required for a phase current to rise to its full value. At a speed of 30000 rpm the phase voltages have a fundamental frequency of 500 Hz and switch positive and negative for periods of 0.857ms per cycle. The phase current has insufficient time to completely build up in these times, and so it is necessary to advance V_{ph} with respect to E_{ph} by some amount.

Since no idea of the required resolution was available, it was thought best to arrange the system to have the best resolution that was practical. Thus the number of SEG' lines was chosen so that the interrupt rate to the microprocessor would not approach the minimum time required to execute each SEG interrupt service routine. In Chapter 4 it is shown that the chosen SEG interrupt service program takes a total of about 32μs to execute. Therefore, it was decided to choose 56 SEG' lines and to sense only <u>one</u> edge on each SEG' pulse. This results in a resolution of 6.43° (using equation 3.18), and a minimum time between interrupts of 35.7μs at 30000 rpm. This was judged to be within the capabilities of the microprocessor system.

The actual shift required between V_{ph} and E_{ph} can be estimated by making some simple approximations. If the phase winding impedance is small:

$$V_{ph} \simeq E_{ph} \quad \text{volts} \qquad 6.80$$

E_{ph} is directly proportional to the motor speed, ω_m (radians per second).

$$\text{i.e.} \qquad E_{ph} = K_b.\omega_m \quad \text{volts} \qquad 6.81$$

where K_b is the back-emf constant (volts per radian/sec). Combining equations 6.80 and 6.81 yields:

$$\omega_m = V_{ph}/K_b \quad \text{rads/sec} \qquad 6.82$$

The time t_m taken to rotate one revolution (2π radians) is given by:

$$t_m = 2\pi/\omega_m = 2\pi.K_b/V_{ph} \qquad 6.83$$

V_{ph} must rise t_r seconds before E_{ph} in order that I_{ph} has attained the desired value at the start of the on period. The time t_r can be defined as:

$$t_r = \alpha\, t_m \quad \text{seconds} \qquad 6.84$$

During the period t_r there is a potential difference of V_{ph} volts applied across the phase impedance. (Note that if t_r is sufficiently large, the potential difference can be $V_{ph} + E_{ph}$, but this is not considered here.) The rate of rise of current in the phase can be approximated by:

$$dI_{ph}/dt = V_{ph}/L_{ph} \quad \text{amps/second} \qquad 6.85$$

If it is assumed that I_{ph} is zero at the start of the period t_r, and it must have the desired value at the end of t_r, then:

$$I_{ph} = (dI_{ph}/dt).t_r \quad \text{amps} \qquad 6.86$$

Combining equations 6.86, 6.85, 6.84, and 6.83 yields:

$$\alpha = I_{ph}.L_{ph}/(2\pi.K_b) \qquad 6.87$$

Since α represents a fraction of the time taken for one rotation, it can also be expressed in terms of an angular displacement δ_{sq}:

$$\delta_{sq} = 360\,\alpha \quad \text{degrees} \qquad 6.88$$

$$\text{i.e. } \delta_{sq} = 180.I_{ph}.L_{ph}/(\pi.K_b) \quad \text{degrees} \qquad 6.89$$

A similiar expression can be obtained if the current fall period is considered instead of the current rise period. The angle δ_{sq} is equivalent to that load angle in a sinusoidal synchronous machine necessary to give a torque angle of 90^{o}. Equation 6.89 is only an approximation since it ignores mutual effects between phases and it does assume that both V_{ph} and E_{ph} have instantaneous risetimes. This is essentially true for V_{ph} but is not so for E_{ph}, as can be seen on the photograph in fig. 6.50. The slow rise and fall times can result in large current spikes which can damage the supply inverter, as explained in Chapter 5. The mismatch in voltage waveform risetimes is advantageous to some extent because it helps to force the phase current to rise to the required level and so could remove the need for a shift between V_{ph} and E_{ph}. However, the mismatch in the fall times is in the wrong sense since it causes the phase current to peak up and does not force it towards zero as desired. Therefore, in order to protect the inverter from excessive current spikes it was decided to add 1.25mH external inductors in series with the phase windings as described in Chapter 5. This precaution eliminates the effects of the momentary voltage waveform mismatches at the pulse edges. Current rise and fall times can then be minimised by shifting V_{ph} with respect to E_{ph} as discussed earlier in this section.

The predicted δ_{sq} for the 7 phase motor without the extra inductors was calculated using equation 6.89 for an estimated no load phase current of 1 amp and was found to be 0.5^{o}. If the external inductor is included, the total phase inductance is of the order of 1.35mH and δ_{sq} has a value of 6.6^{o}. This is virtually equal to the resolution available from the 56 SEG' pulse disc. It was therefore concluded that the optical position sensor unit would be suitable for use with the square wave motor.

6.11 Conclusions

The square wave synchronous motor described in this chapter is suitable for use in a voltage forced square wave motor drive system. It is unfortunate that errors during the design resulted in a lower predicted power output than is theoretically possible from the chosen frame size. However, the main objective throughout the work was to get some form of machine running to test the basic ideas. Some short cuts were necessary to save time, notably the rather inadequate dynomometer design, the lack of dynamic balancing of the rotor, the use of thick laminations, and the rather crude calculation of the airgap flux density. However, the back-emf waveform is very close to that required.

The optical position sensor is by no means compact but it is cheap, reasonably easy to make, and it performs very well without the need for any lenses in the optical paths. It is true that a fair amount of signal processing electronics is needed, due essentially to the rather low resultant level of received light at the photo diodes. With improvements in various optical components (specifically small lenses for the fibre-ends), it will become possible to simplify the amplification circuitry. The consequent reductions in cost and complexity will considerably improve the scheme's viability.

The performance results of the 7 phase motor are presented in Chapter 9.

CHAPTER 7

COMMUNICATION WITH A MICROPROCESSOR EXECUTING A REAL TIME PROGRAM

7.1 Introduction

The control of an inverter fed machine such as a synchronous motor by a microprocessor, requires a program that operates in real time. For example, in order to keep the motor synchronised with the driving inverter, it might be arranged that the microprocessor services a series of interrupts which give it position information about the rotor. Due to the fact that the rotor is continuously rotating, the microprocessor must act on an interrupt as soon as possible, in order that the position information denoted by the interrupt remains valid. The microprocessor hence cannot store up the position interrupts whilst it completes other tasks.

One function that is relatively important in a microprocessor controlled motor system is the communication of data between the human operator and the microprocessor. On the one hand this is one of the tasks that cannot be allowed to interfere with the maintenance of synchronism in the system, but on the other hand it is vital that STOP and similar commands can be entered and recognised by the microprocessor quickly.

Essentially there are two methods by which a

microprocessor can input or output data in a real time program.

(a) The microprocessor can continue performing its main duties until such time that it is interrupted by an interrupt signal from a data transfer peripheral. The microprocessor then assesses the priority of the interrupt and if nothing of higher priority requires attention, it performs the required data transfer.

(b) The microprocessor continuously samples the input ports at regular time intervals and outputs data, when required, during the time that it is not servicing the high priority interrupts. For example, in an interrupt controlled program the main program could consist of nothing more than a wait loop in which the microprocessor is trapped until interrupted. This wait loop can be expanded to incorporate the necessary input/output sequences which would be executed as required whenever the microprocessor was not servicing interrupts. The microprocessor could determine if data transfer was required by sampling flags set by the peripherals. Alternatively the microprocessor could always read in the data, but this can lead to spurious data entry if the data is being altered as the microprocessor reads it.

There are no particular rules as to when or where (a) or (b) should be employed and indeed a combination of them can be used if necessary. However, in general, method (a) is appropriate in a system where data transfer occurs relatively infrequently and method (b) is suited to systems where the data to be transferred is continuously or frequently changing. In a motor control system it would be sensible to use continuous sampling on a data

input from a tachometer, but to only input the required control speed whenever the appropriate interrupt is generated.

Whichever method of data transfer is used, a suitable interface is required for communication between the human operator and the microprocessor used in a motor control system. The job of maintaining a synchronous motor in synchronism at high rotor speeds can keep the microprocessor busy for a large portion of the available time and it is important that the chosen communication system can reliably enter and extract data in the remaining small periods of time available. There are several methods available to achieve real time communication with a microprocessor. The three methods that are worthy of discussion are:

(a) communication via an asynchronous communications port to a Visual Display Unit (VDU) or similar terminal;

(b) communication via special purpose single function peripherals; for example a thumbwheel switch providing a binary code. The switch can be arranged to appear as a "read only" byte of memory in the microprocessor memory map;

(c) communication via a special purpose multifunction keypad which can be hand-held for user convenience.

These alternatives are considered in the following section: 7.2.

7.2 Possible Data Terminal Strategies

7.2.1 VDU or Similar Keyboard/Printer Unit

The usual means of modifying or monitoring program data in computer systems is via a VDU or similar keyboard unit. A VDU is an attractive method of data entry for the following reasons:

(a) it can display transmitted and received data (a relatively large amount if needed) on a common screen;

(b) it is a very common piece of equipment found almost everywhere;

(c) it can be a very powerful peripheral when used with complex control software.

VDU's have been used successfully in motor control systems (7.1). Data is transferred in serial form to and from the VDU at a preset transmission rate, usually somewhere between 300 and 9600 bits per second (baud). (The serial transmission of data is widely used since it requires fewer interconnections than parallel transmission and it permits the use of a standard connecting cable for machines with different size data buses.) An Asynchronous Communications Controller (ACC) provides the necessary interface between the microprocessor and the serial data line.

The data to be sent to and from a VDU is commonly arranged in ASCII code (7.2). Each alphanumeric character and the various control characters on a VDU keyboard have a unique seven bit code word. The character word also has a start bit, a parity bit and either one, one and a half, or two stop bits, depending on the system application. In the 9900 microcomputer system,

the ACC is programmed for two stop bits and can operate at a bit rate of 9600 baud. Therefore, whenever a key is pressed on the VDU keyboard, an eleven bit data word is generated, made up of the appropriate seven bit ASCII code and the start, parity and stop bits. The eleven bit word takes 1.146ms to transmit at 9600 baud and it is received by the ACC and held in parallel form ready for the microprocessor to read it. The ACC signals to the microprocessor to tell it that a new character is waiting to be read from the receive buffer.

Due to the fact that the data is transmitted in serial form, it is possible that electrical noise pick up on the line might cause a particular bit in a character stream to be corrupted. The parity bit is a simple check for errors but it is not foolproof. A corrupted character could be detected using software, by comparing it against a table of permitted ASCII codes, but this would not detect corruption if a permitted character had been changed into another permitted character. It is therefore very useful to echo the character that the microprocessor receives back to the VDU display. In this way the operator can immediately tell if the microprocessor has received what was typed. Therefore the sequence of events to input one character into the 9900 microprocessor might be as follows.

(i) Key pressed on keyboard and serial transmission to ACC starts.

(ii) When the character is completely received in the ACC receive buffer, an interrupt is generated to tell the microprocessor that a character is ready in the ACC.

(iii) When available time and the interrupt priorities permit, the microprocessor executes the relevant interrupt service routine which transfers

the character into a register or the system memory. The microprocessor then places the received character into the transmit buffer of the ACC and initiates a character transmission, the end of which can be signalled by another interrupt. (The microprocessor distinguishes between transmit and receive interrupts by sampling ACC status lines.)

(iv) The VDU receives the character in its receive buffer and displays it on the screen.

The general purpose nature of a VDU keyboard means that special purpose keys such as START, STOP, or FORWARD and REVERSE, which might be useful in a motor control system, are not available to the user unless modifications to the keys are made. To avoid modifications, one must therefore use mnemonic names in which combination of the standard character keys make up easily remembered commands. It is important that a command such as STOP should be easily entered, since it might be necessary to stop the motor system quickly. Therefore commands such as 'S' for STOP and 'GO' for START might be used along with 'FOR' and 'REV' for FORWARD and REVERSE respectively. The single character STOP command is good since the microprocessor can initiate a STOP as soon as it has received the character and identified it as an 'S'. In the case of the 'GO' command, the microprocessor would first receive the character 'G'. The decoding software could be programmed to then expect a character 'O'. If the next received character was not as expected, an error message could be transmitted to the VDU screen.

The entry of data during motor operation is slightly more involved, since information about the numerical value of the data and its intended use is required. For example, if the speed that the motor was running at was to be changed from its current value to say 7000rpm, an entry such as RPM7000 followed by a carriage return

might be made. Similarly, a current limit might be adjusted to say 20amps with a command such as ILIM20.

For all of these data entries, each character is received and echoed one at a time, so that if the operator was a slow typist the data entry could span several seconds. It should be noted that the speed example given involves eight ASCII characters, including a carriage return to terminate the entry. If the echoing procedure is included, this results in a total of 176 bits of data being transmitted to and from the microprocessor at a baud rate of 9600, taking a time of 18.3ms. No allowance has been made for the time to service each character's interrupt service routines. Transfer of the character from the ACC to the microprocessor is not quick in the 9900 microprocessor system. A serial link is used which involves the STCR instruction lasting about 16μs, making the interrupt service routine longer than one might expect. This could be a problem when position interrupts occur at 40μs intervals. Whilst the actual time used is small compared with the speed of typing the data in, it does mean that there is less time available for other functions such as speed control, and the large number of bits increase the chances of error. If the speed data was entered on a 16 bit parallel data link, a single interrupt only would be needed to attract the microprocessors attention and read in the data. Consequently the microprocessor would be involved with data transfer for a much shorter time. Even when the characters have been successfully entered into the microprocessor there may be further delay after the 'CR' character is recognised, whilst the ASCII data is converted into binary code for use in numerical algorithms. Therefore, a VDU is not likely to be the fastest method of entering data into a microprocessor. It is likely to demand a considerable amount of attention from the microprocessor during data transfer, and at high motor speeds this could elongate the procedure beyond acceptable limits.

One possible method of using a VDU in situations where insufficient time is available for serial data transfer, is to use an intermediate buffer unit which converts the data from serial to parallel (and vice versa) and forms the appropriate binary data from the ASCII characters, at the same time interpreting what the data is intended for (speed, volts, etcetera). The unit then generates an appropriate interrupt and loads the data into an input port latch of the microprocessor. However, it is probably better to build a special purpose peripheral or use a VDU with a parallel data link (which are not so common or flexible in use) rather than pursue this course of action.

A further reason why the use of a VDU is unattractive in motor drive systems, is the high cost of a VDU. It is not economic to use a VDU costing at least £500 for simply entering and examining data. Once the motor control program has been developed and stored on EPROM, the VDU is under used for simple control and data entry. The VDU is also large and relatively fragile and not suited to industrial environments, although some miniature hand held VDU's are now available at prices between £245 and £495.

After considering the points discussed in this section, it was felt that the VDU was not the best input/output peripheral to use and consideration was therefore given to the alternatives outlined in the following two sections 7.2.2 and 7.2.3.

7.2.2 Special Purpose Unifunctional Data Peripherals

Often only very simple data needs to be entered or extracted from a microprocessor and this section considers the appropriate peripherals that can be used.

A basic microprocessor system consists of a microprocessor, some memory in which the program and other

data is stored, and various input/output devices which enable the system to be interfaced to the world around it. These basic system components are all interconnected by an address bus, a data bus and a control bus. The control and address buses are manipulated by the microprocessor and they determine which units in the system are allowed to use the data bus at any time, and simple data peripherals must therefore be controlled.

The simplest form of data entry peripheral is simply a switch connected as shown in fig. 7.1.

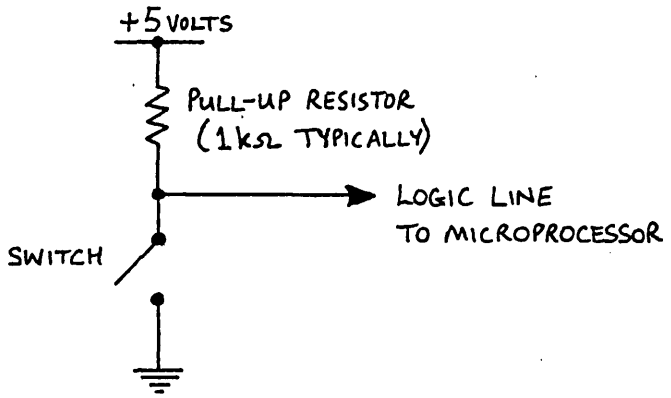


FIG. 7.1. SIMPLE SWITCH INPUT PERIPHERAL

When the switch is closed, a logic 0 signal is placed on the signal line. With the switch open, a logic 1 signal is placed on the signal line. Obviously not a great deal of information can be conveyed by a single signal line, but a parallel binary word can be formed by using several switches, each controlling an input line.

The simplest form of data output peripheral is a single signal line on which a binary voltage signal of say 0 volts or +5 volts can be placed by the microprocessor. This signal can be used to illuminate a lamp, switch a relay or do any simple task requiring a true/false type of output. By having several lines in parallel, binary numbers can be output, which may be used

to drive light emitting diodes, or even perhaps a digital to analogue converter so that an analogue signal can be obtained.

The method by which the input lines can be interfaced with the microprocessor data bus system depends very much on the particular microprocessor used. Popular methods that are currently in use are as follows.

(a) Some microprocessors have a few dedicated input lines on which data can be placed by a peripheral and sampled by the microprocessor. Similarly there are some microprocessors that have program controllable output lines. For example, the RCA 1802 microprocessor has four testable external flag lines which can be used for the input of binary data, and a single output line which can be set and reset.

(b) Most microprocessor systems offer a programmable parallel interface circuit, usually called a peripheral interface adapter (PIA) or a programmable systems interface (PSI). This type of circuit has several signal lines which may be programmed as inputs or outputs. Individual programming of the lines is sometimes possible. The circuit usually occupies a particular point in the microprocessor memory map and all control instructions to the interface circuit and the data to be transferred, pass through that point in memory. A typical PIA is the Motorola MC6820 which contains two input/output blocks, each capable of controlling an independent 8 bit peripheral. In addition four control/interrupt lines may be programmed to suit the user's application.

More than one PIA can usually be connected to a system and they do offer an extremely cheap and easy

method of interfacing external signals to a microprocessor.

(c) In certain applications the software program is simplified if the data peripherals are configured as locations in memory. To input data from the peripheral it is then a simple task to read the data at the appropriate memory location. To ensure that the correct peripheral is enabled when the microprocessor communicates with that location in memory, it is necessary to decode the control bus and address bus signals. For data output, the decoded signal can be used to latch data from the data bus into an output peripheral. For data input, the decoded signal is used to enable the peripheral so that it can drive its signals onto the data bus. At all other times the output peripheral must not interfere with the data bus. The output and input arrangements are shown schematically in figures 7.2(a) and 7.2(b) respectively.

This form of interfacing is called memory mapped input/output. It is simpler to use than a PIA since no software initialisation is needed for the peripheral, but the cost of the necessary decoding hardware can be excessive if many input/output ports are required.

The method used to interface signals to the microprocessor does affect the complexity of the input or output peripheral that can be used. If four parallel lines are available as an input, a thumbwheel switch with ten or sixteen positions can be used with its outputs either coded in binary coded decimal or binary. Sixteen parallel lines would allow a four digit hexadecimal number to be entered from four thumbwheel switches. If four parallel output lines are available they can be decoded to drive a hexadecimal display. Seven lines can

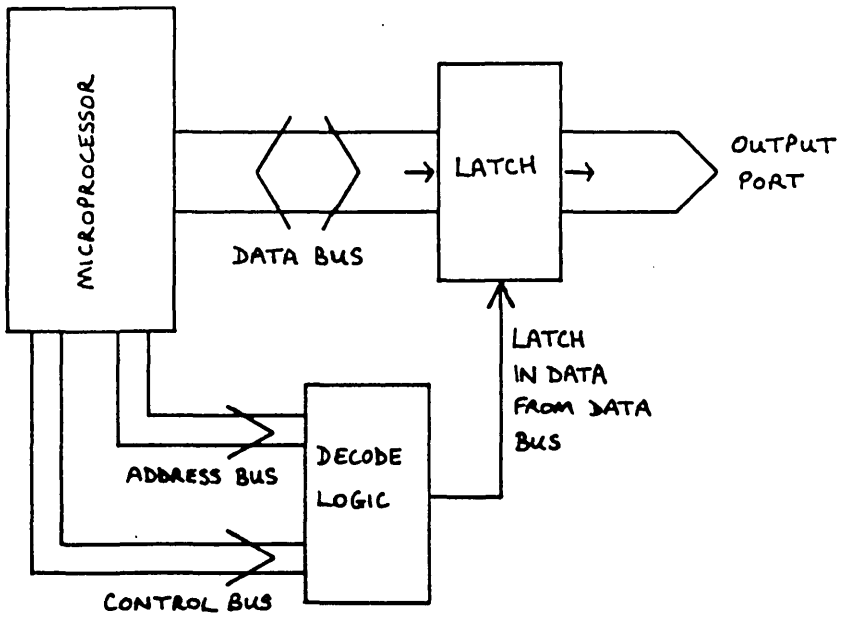


FIG. 7.2a. TYPICAL MEMORY MAPPED OUTPUT PERIPHERAL ARRANGEMENT

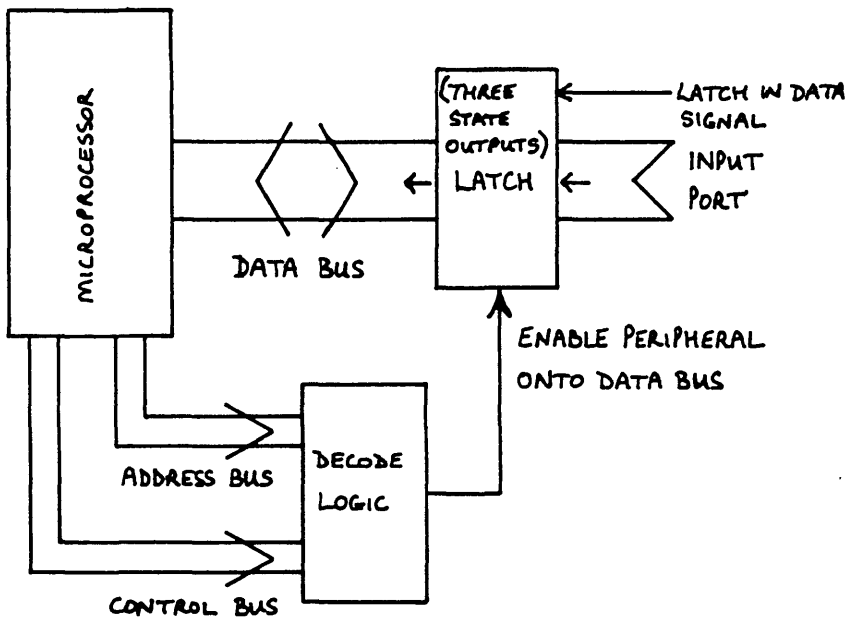


FIG. 7.2b. TYPICAL MEMORY MAPPED INPUT PERIPHERAL ARRANGEMENT

be used almost directly to drive a standard seven segment display. Eight lines can drive a digital to analogue converter to provide an analogue output voltage. The possibilities are enormous and a suitable peripheral can be constructed for most simple requirements. The main disadvantage, however, is that the peripheral generally can only perform one function, which might result in the need for more peripherals, and very often the data format is for the convenience of the microprocessor, not the human operator. (For example the entry of data in binary form rather than decimal.) It is possible to enter data directly in BCD form but this often carries the penalty of needing more input lines. As an example, 65000_{10} requires twenty parallel lines to enter it in parallel in BCD form, but only sixteen lines when entered in binary form. It is of course possible to enter the digits one at a time, in which case only four lines are needed in either case, but this is once again less convenient for the user.

Special purpose unifunction peripherals are relatively cheap but are inflexible and can require a high degree of user awareness.

7.2.3 Multipurpose Keypad

A Multipurpose Keypad is a solution to the data interfacing problem, which combines the cheapness and robustness of a simple peripheral with the ease of use associated with a keyboard VDU.

A keypad is a small, possibly hand held, unit which has on it a limited simple set of key switches for data and command entry and may also have a simple display for status information. The limited number of switches reduce the possibility of an incorrect entry such as can occur on a full sized VDU keyboard when adjacent characters are accidentally pressed. Ideally the display will

show the information being keyed in and possibly could display data recalled from the microprocessor memory. In some cases the function of the keys can be reprogrammed, either internally or by the microprocessor software. In addition some basic data checks can be performed by the keypad before the data is transmitted to the microprocessor. For example, in a 16 bit microprocessor system the maximum numerical value that can be dealt with is 65535_{10}. If a number greater than this is entered in the keypad, it can indicate an error immediately. Various control buttons can be interlocked so that signals can only be modified in safe situations. It is the "semi-intelligence" of a keypad and the easily recognised function buttons that make it a flexible and attractive tool in motor control systems.

There are several ways in which data can be transmitted between the keypad and the microprocessor system. The data can either be transferred in parallel or serial form. Obviously if a cable is used to interconnect the two units, it is more attractive to use a serial data link to minimise the size of the connecting cable, but this does require some decoding logic at the microprocessor end to format the data into a suitable state for entry into the microprocessor. If a parallel connection is used, the data can be entered into a PIA or a memory mapped port directly, with perhaps interrupts used to indicate the presence of new input data. The use of a wire link does, however, limit the full flexibility of a keypad. If an infra-red or ultrasonic link is used, the operator can walk around the motor system whilst carrying the keypad controller with him and there are no long trailing leads to worry about. Portable remote control keypads have become very common recently with domestic television receivers in the U.K.

If it is desired to display microprocessor data on the keypad there is no problem when using a wire link but

a second ultrasonic/infra-red channel is needed for a remote control keypad. The need to minimise power consumption in a remote keypad can be helped by the use of liquid crystal displays (LCD). Alternatively, a separate display unit can be used to display data requested from the microprocessor. The display unit should be in an easily visible part of the area in which the motor system is located.

The keypad offers great flexibility at low cost and it was therefore decided to use a keypad in the motor control system described in this thesis.

7.3 Possible Keypad Circuit Strategies

The choice of techniques available to implement a keypad peripheral is quite wide and their use has been reported in the technical press (7.3). Three alternatives were considered at the time it was decided to use a keypad. The alternatives are discussed in the following sub-sections.

7.3.1 Hardwired Logic Circuitry

The first task of the keypad is to scan a set of keys and detect which key has been pressed. Several integrated circuits are available which perform this task and they output a binary coded word containing the information about which key is depressed at any time. The circuits usually debounce the switches and provide two key roll-over. The second task of the keypad is to arrange the data into a format suitable for the microprocessor and if necessary, combine several keystrokes to form binary numbers or special commands. This type of logic problem can involve combinational and sequential logic and the widely available TTL and CMOS logic families can be used to implement the required logic functions.

The main disadvantage of using hardwired logic is that it is less easy to change the functions of the keypad once it has been constructed. However, the absence of any software in the system reduces the likelihood of any spurious bugs once the circuit has been successfully developed, and the ease of using CMOS and TTL makes it an attractive path to follow.

7.3.2 Keypad Controlled by the Main System Microprocessor

The task of scanning a set of keys to determine which key is depressed can be performed by a microprocessor, providing the keys are arranged in a matrix form as shown in fig. 7.3 where a twenty switch keyboard is shown.

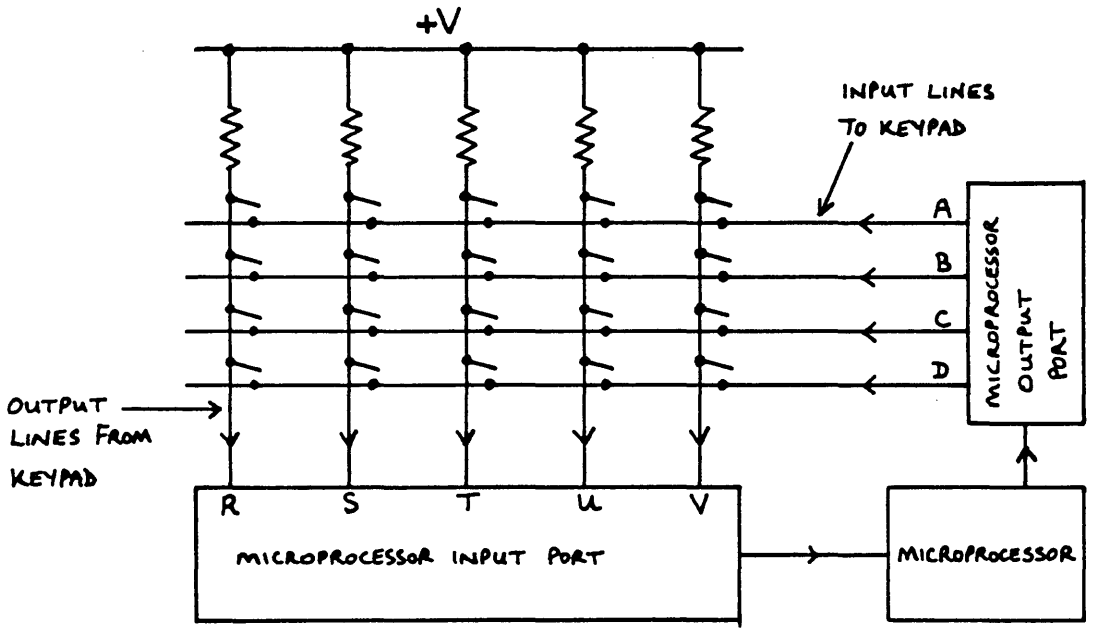


FIG. 7.3. TWENTY SWITCH KEYBOARD SCANNED BY A MICROPROCESSOR

The microprocessor scans the keyboard by taking lines A,B,C and D to logic 0 in succession. If a switch is closed, one of the lines R,S,T,U or V is taken low at a particular time, depending on which input line (A,B,C

or D) is connected to it by the switch. The microprocessor can therefore determine which switch is closed and if more than one switch is closed it can decide which switch should have priority. In addition it can debounce the switches by checking if a closed switch remains closed for a given period of time. This system is very simple since the microprocessor software performs all of the logic but the need to scan the keypad continuously does take up a significant portion of the microprocessor's time. Further microprocessor time must be spent decoding the meaning of a combination of successive keystrokes.

Therefore this method is unlikely to be suitable for a system in which a high speed real time process is being controlled by the same microprocessor. Also it is not possible to avoid hardware if the display of data is required. Very often it is crucial to know what has, or is being, entered into a system.

7.3.3 Keypad Controlled by a Dedicated Microprocessor

To enable the keypad switches to be continuously scanned and yet gain the benefit of software performing all of the complex logic operations, it is possible to use a dedicated microprocessor to monitor the keys and decide what action has been demanded. Virtually any of the currently available 8 bit microprocessors would be suitable. Microprocessors such as the Motorola MC6801 which possess self contained ROM and RAM memory plus some input/output lines, are ideal as they reduce the component count. However, for a fully remote keypad the best candidate at present is probably the RCA 1402 CMOS microprocessor, which draws very little power and is ideal for battery operation. The ability to change the control program and hence change the function of the keypad without any hardware modifications makes this particular method very attractive.

7.4 The Necessary and Sufficient Functions Required to Enable a Keypad to Control the Motor System

Considerable time was spent deciding exactly what features were required on the keypad. Those finally selected are listed below.

(i) A STOP/START key with light emitting diodes (LED's) to show the selected status. The key should only be active if the microprocessor has initialised the system.

(ii) A FORWARD/REVERSE key with associated display LED's. The forward/reverse status should be interlocked with the STOP/START key so that the direction selected can only be changed when the motor is stopped.

(iii) Sufficient keys to enable one of six possible variables to be selected and LED's to indicate which variable is selected. Possible variables are voltage, demanded speed, current limit or load angle.

(iv) To permit the modification of a selected variable, a set of numeric keys (0-9) to enable the entry of decimal integers in the range 0 to 65535_{10}. The entered number should be displayed on a seven segment display on the keypad. Numbers outside the permitted range should be shown as errors.

(v) An ENTER key to enable the user to indicate to the microprocessor that new data is available.

(vi) A DATA RECALL key (later called 'PRINT' on the keypad unit) which places the keypad in a mode that allows data to be accessed from predefined memory locations in the microprocessor memory. The recall mode should be indicated by a LED. Display of

the data should be brought about by selecting the data recall mode, then entering a single digit number 0-9 on the numeric keys, followed by the ENTER key. If two numbers are entered, the second should overwrite the first. The memory addresses corresponding to the data recall numbers 0-9 are held in the microprocessor program memory and can be modified if desired.

(vii) An extension of the data recall mode should allow the last entered value of any one of the six variables to be recalled for inspection. To achieve this, the data recall mode should be selected. A CONTROL key should then be pressed at the same time as the required function key. The CONTROL key should remain pressed whilst the ENTER key is then selected.

(viii) To avoid the need for a CLEAR key, incorrect data should be correctable simply by selecting the required function again.

(ix) To show that a particular piece of data has been sent to the microprocessor, the display should clear when the ENTER key is pressed.

7.5 Keypad Circuit Design

The limited amount of available development time meant that a hardware logic solution was chosen for the keypad. A microprocessor based keypad would have taken too long to build and debug. It was reasonably simple to implement the specifications listed in section 7.4 using standard logic. The keypad was designed to work into the memory mapped microprocessor input/output interface board described in Chapter 4.

7.5.1 Practical Considerations Relevant to the Design and Construction of the Unit

The requirement that the keypad should be hand held places a restriction on the size of the keys and display that can be accomodated on the front panel of the keypad box. The limited room inside the box also means that only the logic that really has to be adjacent to the keypad and display can be placed in the box. The remaining logic must be housed in an auxiliary keypad logic unit next to the microprocessor system. An infra-red data link would have been very pleasing, but since the technology is reasonably standard and the need for remote control was not critical, it was decided to save time by using a wire link. Similarly the data transmission was made parallel rather than serial to simplify the logic required. Finally, it was initially hoped that recalled data from the microprocessor memory could be displayed on the keypad display. Unfortunately, the limited space for logic did not permit this and so the recalled data is displayed on a display housed in the auxiliary keypad logic unit case.

7.5.2 Choice of Components

It was decided at the outset of the design to use as much CMOS logic as possible to minimise system power consumption, but wherever a suitable CMOS integrated circuit was not available TTL was used. A sixteen key multiplexed switch unit was chosen for the main keyboard, with three extra micro-switches for the STOP/START, FORWARD/REVERSE and CONTROL keys. The sixteen keys are: 0-9; mode keys a,b and c and a shift lock key to select an "upper case" A,B,C giving six variables; a PRINT key to select data recall; and an ENTER key. The shift lock remains enabled until it is cancelled by pressing it again.

A 5 digit Red 7 segment LED display was to be used. It was hoped to assemble the keypad in a 190 x 60mm A.B.S. box.

It was decided to construct the auxiliary logic circuits on single size Eurocards and mount them in a Eurocard rack. The power supply required for the keypad logic would fit in the Eurocard rack. All the circuits were wirewrapped.

7.5.3 Signal Lines Required Between the Two Keypad Units and the Microprocessor Interface Board

When in one of the variable modes, the presence of new data for the microprocessor is indicated by an interrupt on the data available* interrupt line, the interrupt being generated when the ENTER key is pressed. The new data is carried on a 16 bit parallel bus and the interrupt signal is used to latch the data onto the microprocessor interface board. The code that identifies which variable the data is intended for is carried on a separate 4 bit parallel bus and the data available interrupt also latches this onto the interface board. This bus also carries the data recall code number when the keypad is in the print mode. The presence of a data recall number is indicated by a data recall† interrupt line, the interrupt again being generated when the ENTER key is pressed.

The STOP/START information is carried on the stop/start interrupt line and the FORWARD/REVERSE data is carried on a flag line, the status of which is tested prior to a motor start.

Finally, a control line (SYSTEM INITIALISE LINE) comes from the interface board and determines whether the STOP/START key is active.

In practice a 25 way signal cable plus a twin core power lead was needed between the keypad and its auxiliary unit, and two 25 way signal cables connected the auxiliary unit to the microprocessor interface board. One lead carries data from the keypad and the other carries display data to the auxiliary display unit.

* CALLED THE "DATA" INTERRUPT IN CHAPTER 4
† CALLED THE "DISPLAY" INTERRUPT IN CHAPTER 4
} SEE SECTION 4.6.

7.5.4 Circuit Description and Diagrams

The basic elements of the system are:

(i) a keypad decoder to scan the keys and indicate when a new keystroke is made;

(ii) a keypad display unit to indicate what is entered on the keyboard;

(iii) a command decoder to monitor the keystrokes and decide if the entered character is a decimal number 0-9 or a mode selector (A,B,C, shift lock, print or enter);

(iv) a BCD converter circuit to convert the entered decimal numbers into 16 bit binary. This circuit also supplies the necessary clock signals to the rest of the system.

(v) A display unit to display the recalled memory data in decimal and hexadecimal form;

(vi) a regulated +5 volt power supply.

The main system block diagram is shown in fig. 7.4 with the data recall display block diagram shown in fig. 7.5.

The basic operation of the keypad is as follows.

Referring to fig 7.4, when a key is pressed on the sixteen switch keypad, it is debounced and a character code and new character available signal are generated. After a delay to allow for propagation and settling times in the command decoder, the character code is examined to see if it is a decimal number (0-9) or a mode selector character (A,B,C, shift lock, print or enter). If the

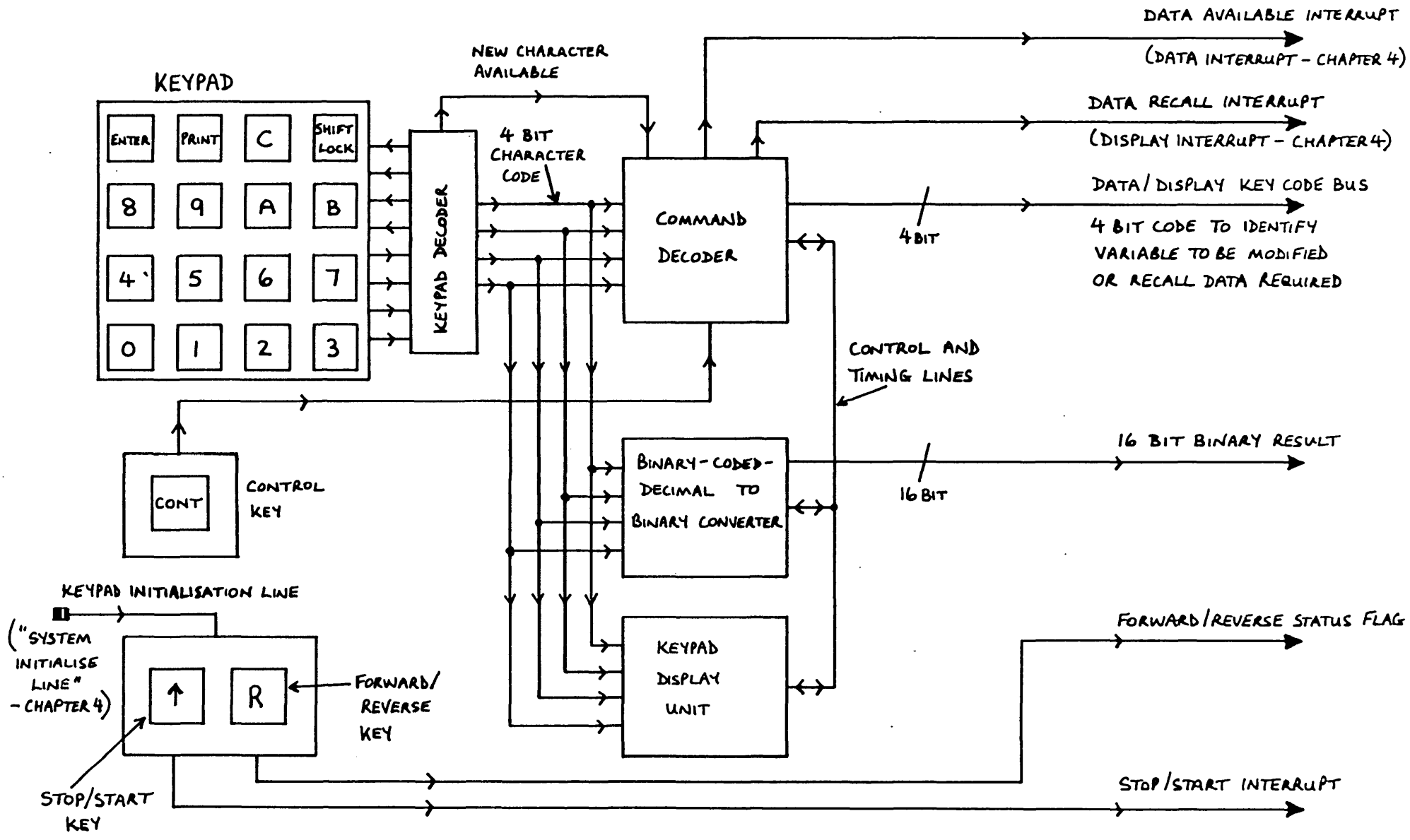


FIG. 7.4. BLOCK DIAGRAM SHOWING THE MAIN UNITS MAKING UP THE KEYPAD SYSTEM

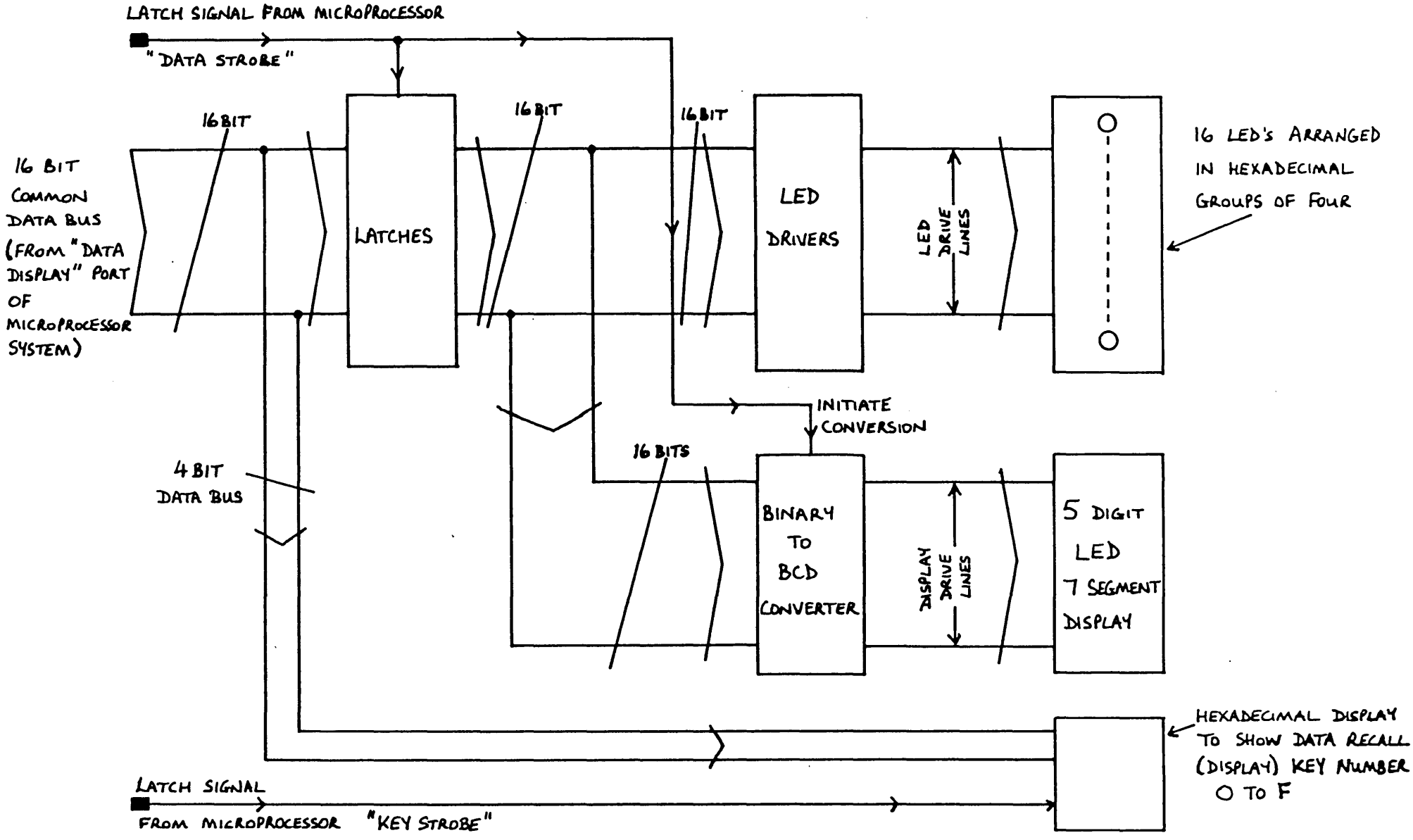


FIG. 7.5. BLOCK DIAGRAM OF THE DATA RECALL DISPLAY UNIT.

character is a mode selector, the command decoder stores this information and clears the BCD-to-binary converter and the keypad display. In addition, if the mode selected is ENTER, a data available interrupt or a data recall interrupt is generated, depending on the mode previously selected before ENTER. If the character is a decimal number, it is fed into the BCD-to-binary converter and also into the keypad display unit. As successive numbers are entered into the BCD converter, they are given a weighting ten times that of the previous character and are added into the previous result, by using a recirculating shift register technique with full adders. This allows a binary result to be available after each keystroke, and is a great advantage over converters which have to wait until the last digit has been entered, in order that conversion can take place. Successive numbers are also entered into the keypad display and displayed numbers are shifted to the left, as in standard calculators, by a shift register. The BCD converter is arranged so that it will only accept five BCD characters. Furthermore, the converter conveniently generates signals which indicate that the entered number has exceeded the capacity of a 16 bit binary register. If the entered number is larger than 65535, the display is made to flash.

When the desired number has been keyed into the display, a data available interrupt is generated by pressing the ENTER key. The microprocessor responds to the interrupt by reading in the 16 bit binary number and also the 4 bit Data/Display key code. This code is used by the microprocessor to decide which one of the six variables the binary number is intended for in the microprocessor program. If for any reason a mistake is made when keying in the number, a re-entry can be made by reselecting the desired mode followed by the correct numbers. This technique saves having to use a clear key.

If the keypad is in data recall mode (selected by

the PRINT key), the BCD converter is not involved in the operations. Having selected PRINT, a decimal number can be entered which is routed directly through the command decoder onto the 4 bit Data/Display key code bus. If two numbers are entered, the second one overwrites the first, both on the keycode bus and in the keypad display. When the ENTER key is pressed, a data recall interrupt is generated. The microprocessor responds to the interrupt by reading in the Data/Display key code. It uses the code to select the data required from the memory and outputs it to the display unit.

The stop/start and forward/reverse logic is completely separate from the data input and data recall circuitry. The STOP/START, FORWARD/REVERSE and CONTROL switches are debounced by capacitors and schmitt triggers. When in the stop state, the forward/reverse status can be changed. As only one line is used for stop/start information, the microprocessor senses both the rising and falling edges as interrupts and reads the logic level on the line to determine if the interrupt just received is a stop or start request. A keypad initialisation line is controlled by the microprocessor. This holds the stop circuitry in the stopped state unless initialised. Since active low logic is used with TTL logic gates, the keypad is placed in a stop mode if the initialisation line is broken for any reason, as the input floats to a logic 1. The system status is indicated on a set of traffic lights above the STOP/START key. When NOT initialised a red LED is on; initialised but stopped is indicated by an amber LED; GO is indicated by a green LED.

The data recall display unit is very simple. Referring to fig. 7.5, a 16 bit binary number is latched off the input data bus by a signal* generated by the microprocessor. The binary number is displayed directly on LED's grouped in fours to enable its hexadecimal value to be easily read. The binary number is converted to BCD by the

* "DATA STROBE"

standard count up/count down method. The number is latched into a 16 bit binary down counter and at the same time a 5 decade BCD counter is cleared. The down counter is then clocked to zero and the same clock pulses are used to increment the BCD counter. Thus when zero is reached, the BCD counter contains the correctly converted decimal number.

The operation of the various circuit blocks are explained in more detail below.

The circuit diagram for the keypad decoder and display is shown in fig. 7.6. The circuit is actually assembled on two small boards (A & B) mounted in the keypad case, and the circuitry is split between board "A" and "B" as indicated by figs 7.6a and 7.6b respectively. Whenever a key is pressed the keypad encoder (IC15) produces the appropriate code on its data lines and a character ready pulse. The command decoder board examines the generated code. If it is a mode character, the keypad display is cleared by parallel loading each bit of the 20 bit display shift register, comprised of IC6, 7 and 8, with logic 1. The outputs of the shift registers are decoded by the decade display drivers IC9 to IC13 inclusive, and with every decoder input at logic 1 the display is blanked. If the generated code is a number between 0 and 9, the command decoder arranges for it to be latched into the least significant 4 bits of the display shift register (IC8). If the number is the first one after a clear operation, it is simply latched in. However, for subsequent numbers in the data entry mode, the shift register first shifts the previously entered number towards the most significant end of the register by 4 bits (i.e. the bits move leftwards on the diagram). After each shift and latch operation, the resulting shift register contents are latched into the display decoders (IC9 to IC13). Latching the data into the displays at this point, prevents display flicker or spurious characters being seen, which might occur if the dis-

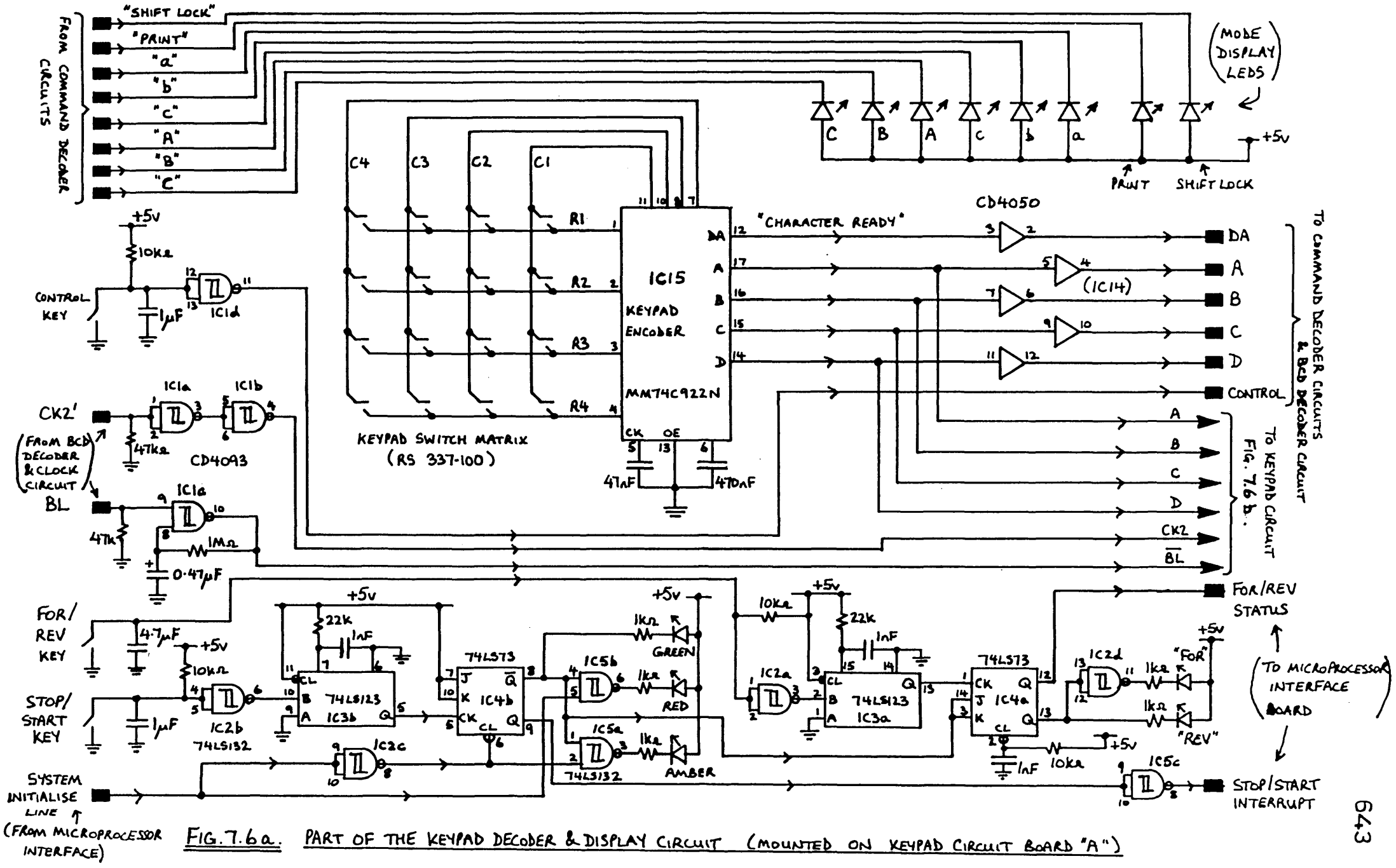


FIG. 7.6a. PART OF THE KEYPAD DECODER & DISPLAY CIRCUIT (MOUNTED ON KEYPAD CIRCUIT BOARD "A")

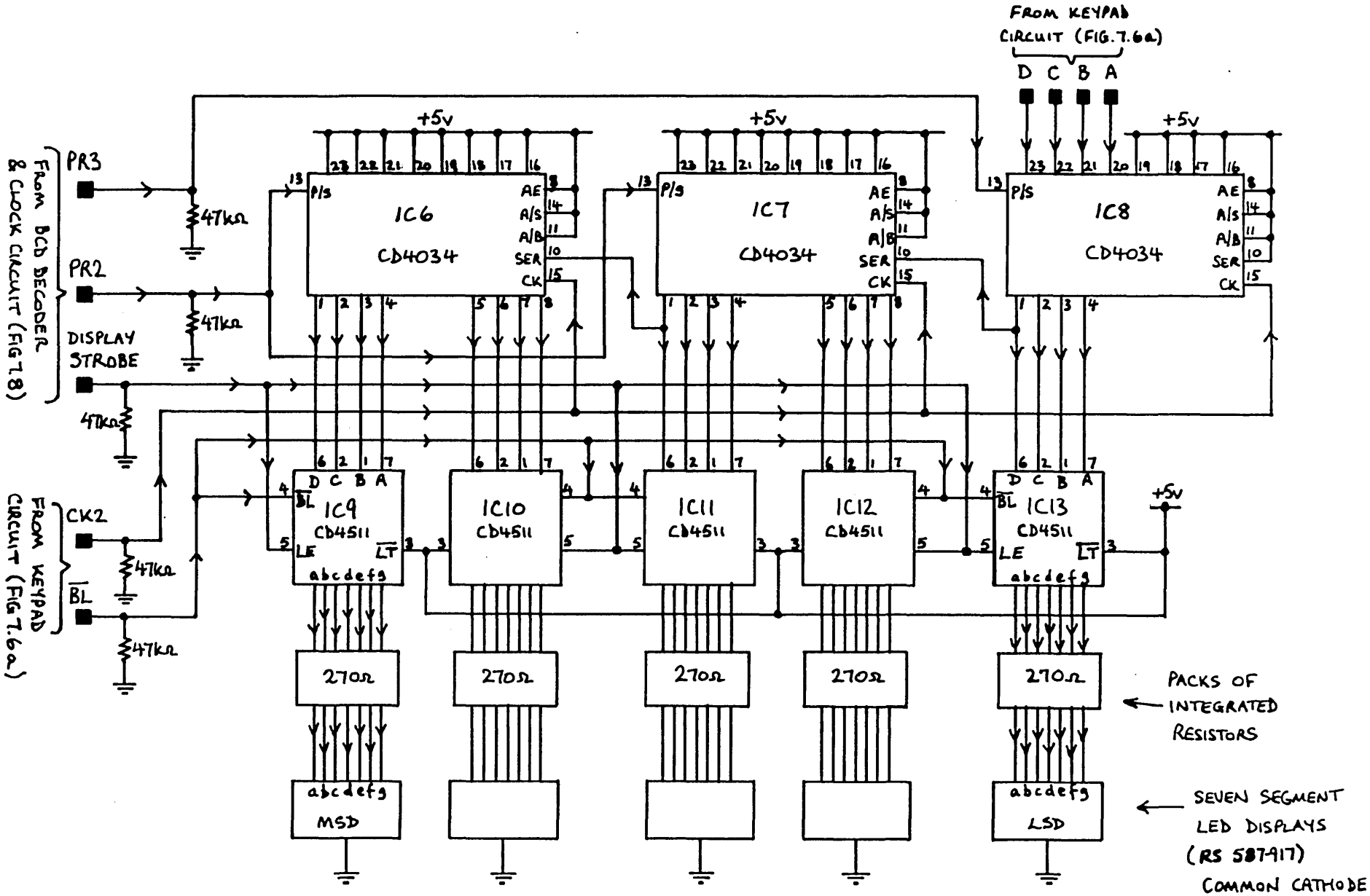


FIG. 7.6b. PART OF THE KEYPAD DECODER & DISPLAY CIRCUIT (MOUNTED ON KEYPAD CIRCUIT BOARD "B")

plays continuously displayed the shift register data as it was shifted.

The operation of the stop/start circuitry (IC2, IC3, IC4, etcetera) is very simple. The switch signals are debounced by a combination of RC filters, schmitt triggers (IC2a, IC2b, IC2c) and monostables (IC3a and b). The resulting pulses change the states of the flip-flops (IC4a and b) which store the stop/start and forward/reverse status.

The circuit diagram for the command decoder is shown in fig. 7.7. The circuit was built on two eurocard boards, A and B, with the circuitry split between board A and B as indicated by figs 7.7a and 7.7b respectively.

The essential elements of the command decoder are the demultiplexer (IC3) which decodes the keypad character code; the mode stores (IC8, IC9, IC5a and b) which latch the last selected mode and arrange for its display; the encoder (IC12) which produces a binary code to indicate which of the six data variable modes is selected, and the multiplexer (IC13) and latch (IC14) which place the data variable identity code onto the 4 bit key code data bus.

When in the print mode the numerical character present at the inputs of the demultiplexer (IC3) is routed directly onto the 4 bit key code data bus via the multiplexer IC13 and the latch IC14.

The various monostables are required to delay control pulses by appropriate amounts so that the propagation delays through the integrated circuits can be allowed for.

The circuit diagram for the BCD decoder and clock circuit is shown in fig. 7.8. The action of the BCD-to-binary converter is more easily understood by studying the simplified form shown in fig. 7.9. The integrated circuit numbers listed in the following description refer to fig.

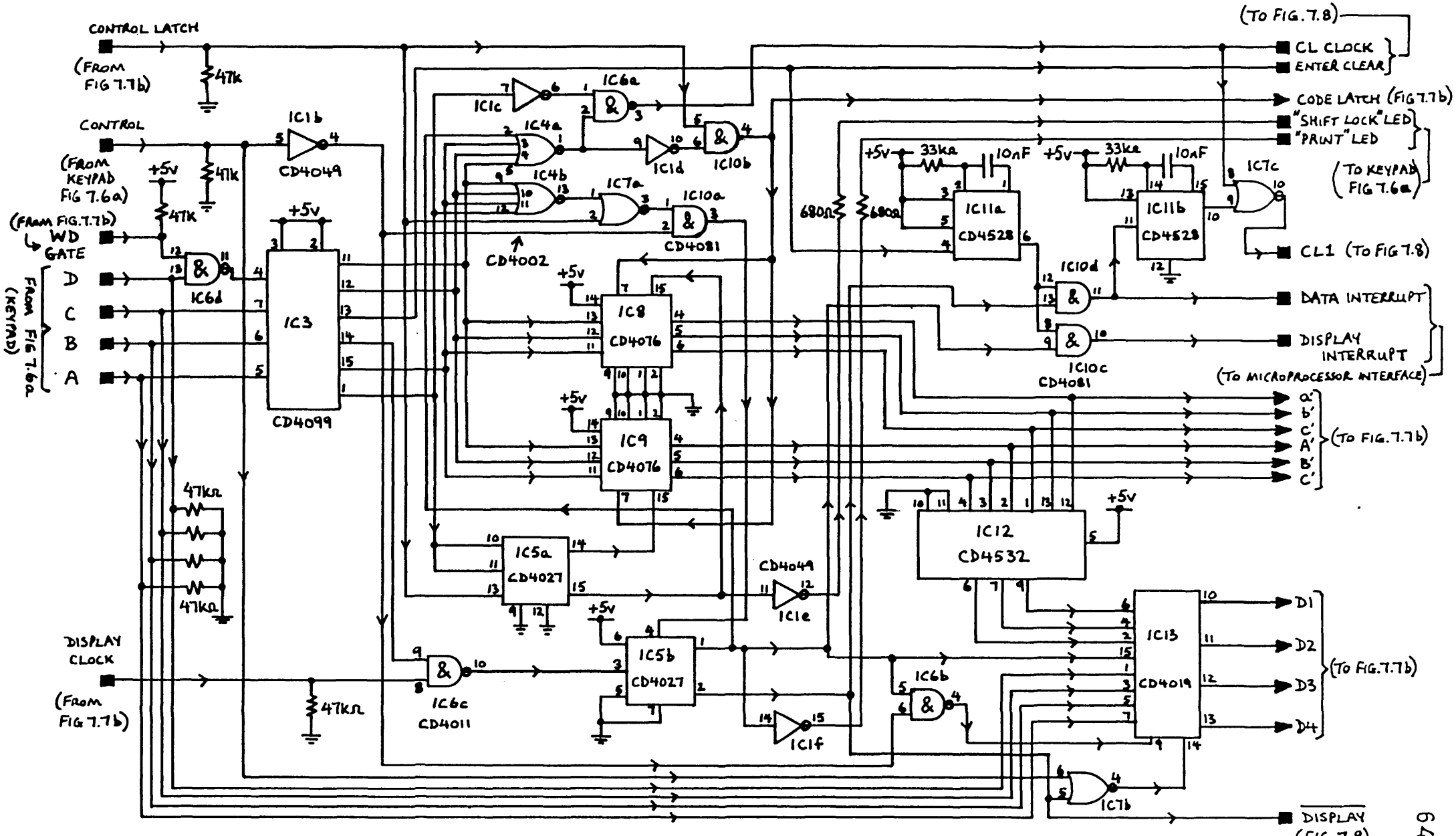


FIG.7.7a. PART OF THE COMMAND DECODER CIRCUIT (MOUNTED ON COMMAND DECODER BOARD "A")

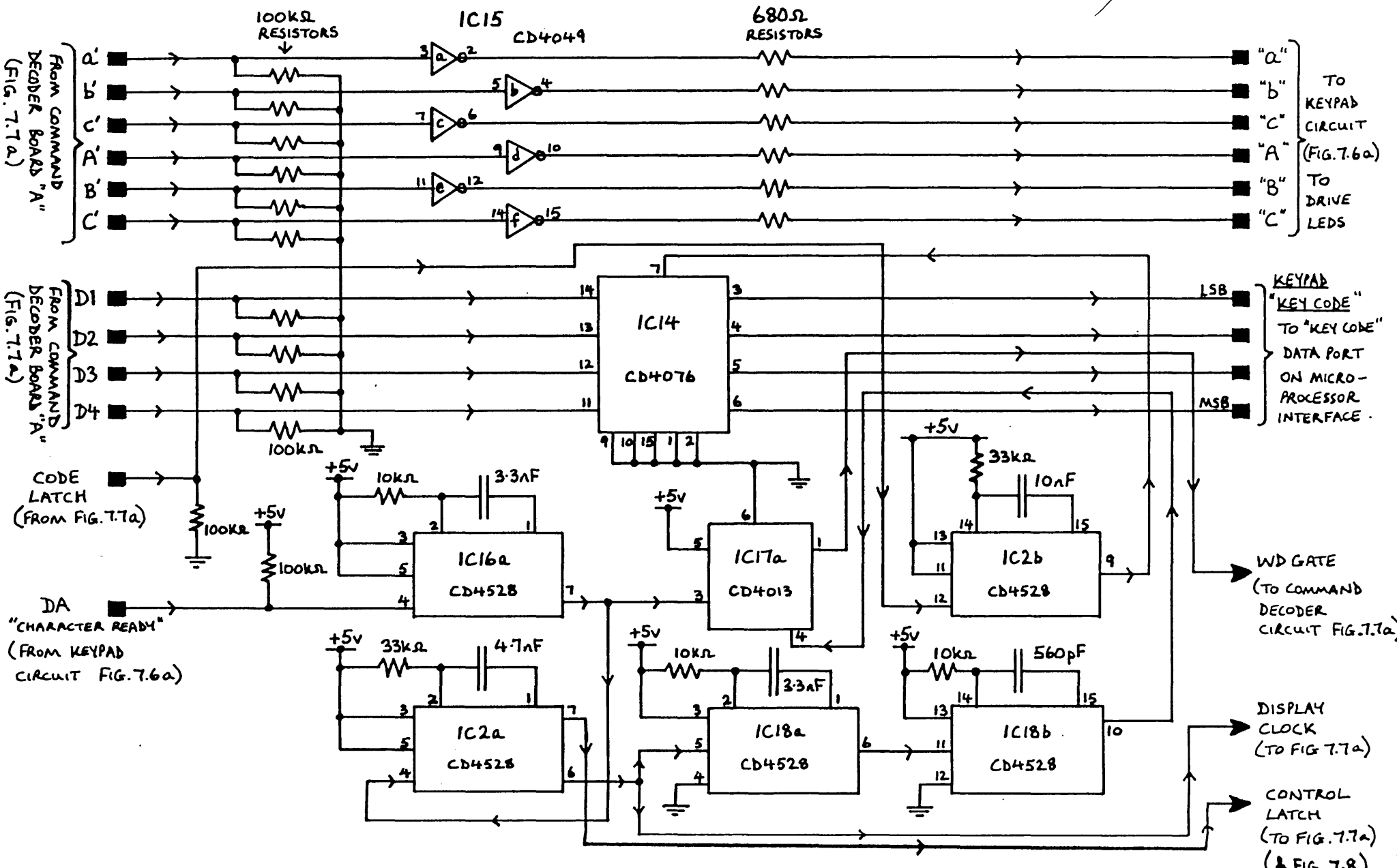


FIG. 7.7b PART OF THE COMMAND DECODER CIRCUIT (MOUNTED ON COMMAND DECODER BOARD "B")

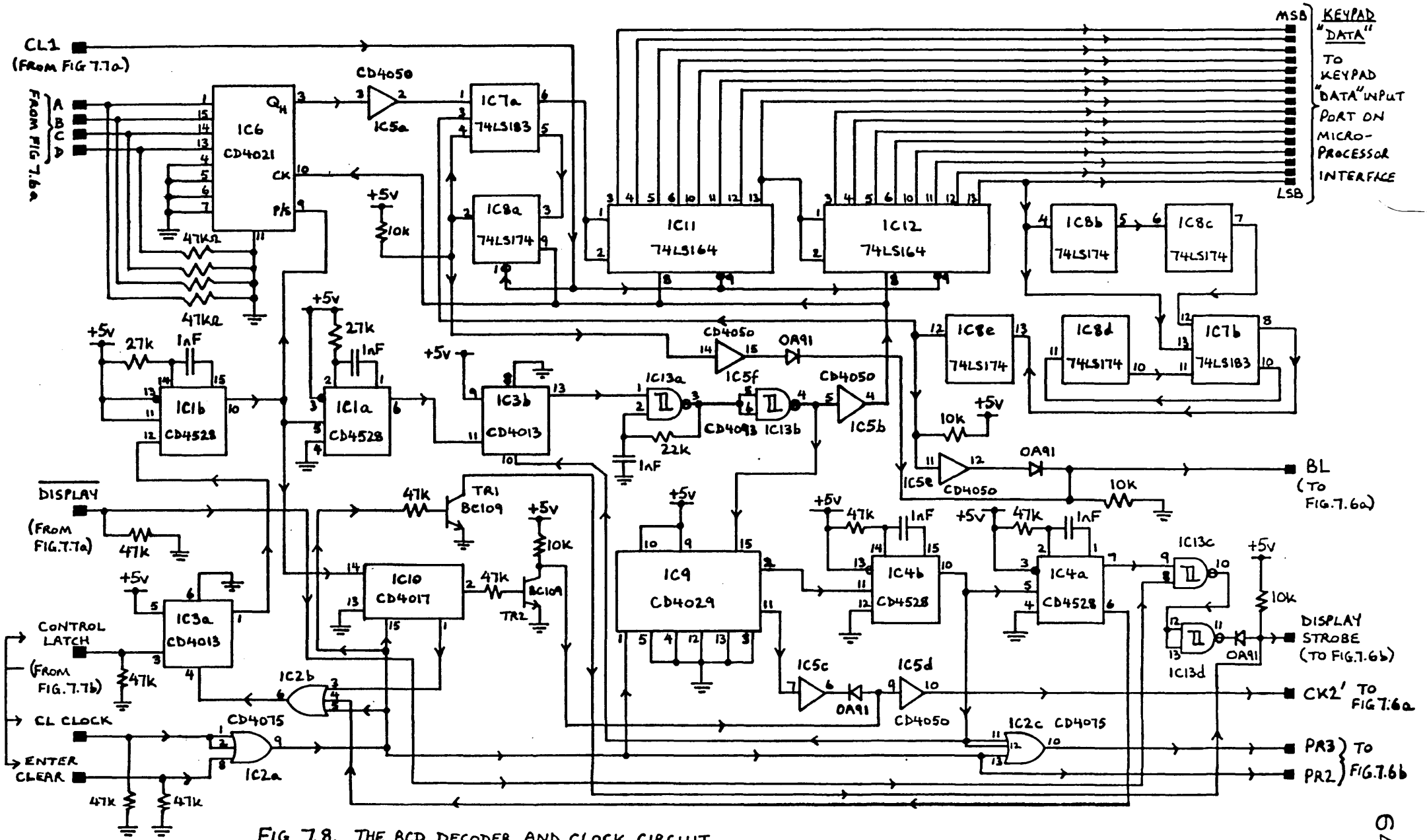
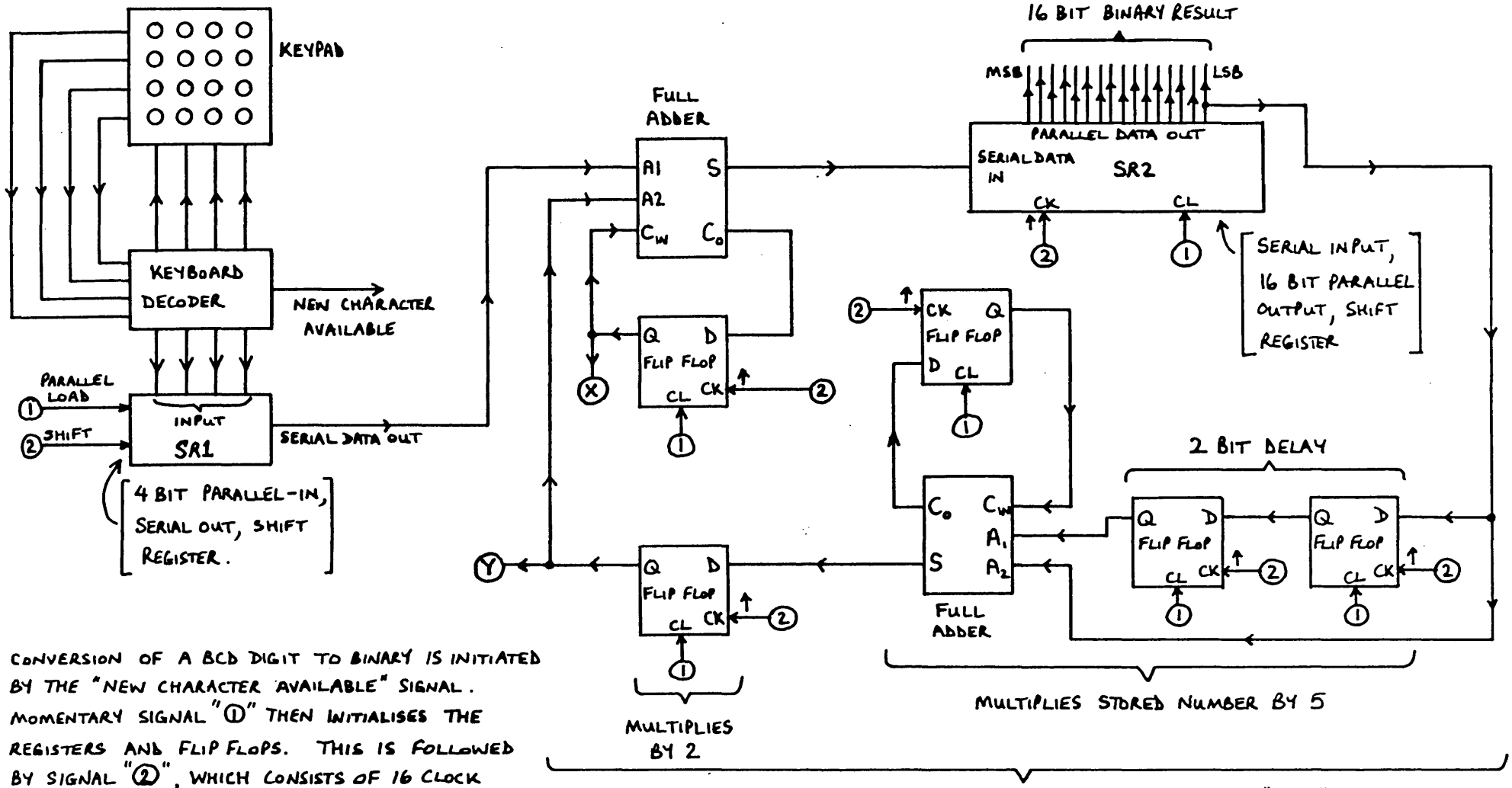


FIG. 7.8. THE BCD DECODER AND CLOCK CIRCUIT



CONVERSION OF A BCD DIGIT TO BINARY IS INITIATED BY THE "NEW CHARACTER AVAILABLE" SIGNAL. MOMENTARY SIGNAL "①" THEN INITIALISES THE REGISTERS AND FLIP FLOPS. THIS IS FOLLOWED BY SIGNAL "②", WHICH CONSISTS OF 16 CLOCK PULSES. N.B. THE 4 BIT SHIFT REGISTER "SR1" IS CLOCKED 16 TIMES EVEN THOUGH IT CONTAINS ONLY 4 BITS OF DATA. IT SIMPLY OUTPUTS LOGIC '0' FOR THE FINAL 12 CLOCK PULSES OF A CONVERSION CYCLE.

IF A 5 DIGIT BCD NUMBER GREATER THAN 65535 IS ENTERED, THEN AT THE END OF THE FIFTH DIGIT CONVERSION CYCLE AN OVERFLOW CONDITION IS INDICATED BY SIGNALS "Ⓧ" AND/OR "Ⓨ" BEING ACTIVE.

FIG. 7.9 SIMPLIFIED DIAGRAM OF THE BCD TO BINARY CONVERTER CIRCUIT

7.8. The converter is based on an old design dating from a time when full adders were relatively expensive and hence were used sparingly. This design only requires two full adders (IC7a and b). Assuming that the 16 bit result shift register (IC11 and IC12) has been cleared by the selection of a mode previously, the sequence of events is as follows. The 4 bit character generated by the keypad is examined by the command decoder. If it is found to be a number, it is latched into the 4 bit shift register (IC6). The 100 kHz oscillator formed by a schmitt trigger (IC13a) in association with the binary counter IC9 , produces four clock pulses to shift the keypad display over by one digit and sixteen pulses to clock the data in IC6 down to the last significant bit end of the result shift register (IC12).

If a subsequent numerical character is entered, it is latched into IC6 and similarly shifted into the result register, but the previously entered digit is added back onto it, having been multiplied by a factor of ten by the delay flip flops (IC8a to e) and the full adders (IC7a and b). IC10 counts how many characters have been entered and locks out more than five. It also stops the keypad display shifting the first entered number to the left. This is achieved by the transistor*-diode gate which blocks the four clock pulses. If the converted number exceeds 65535, the overflow output BL is active and causes the keypad display to flash. The display strobe output ensures that the keypad display data is latched into the displays after being shifted. This output is permanently active in the print mode since no shifting occurs in the keypad display.

As stated earlier, this type of BCD-to-binary converter supplies the correct binary output after each entered character. The time required to convert each character is 0.16ms at the 100 kHz clock rate chosen.

The interconnections between the keypad circuit, the command decoder and the BCD converter are shown in fig. 7.10.

[* TR2]

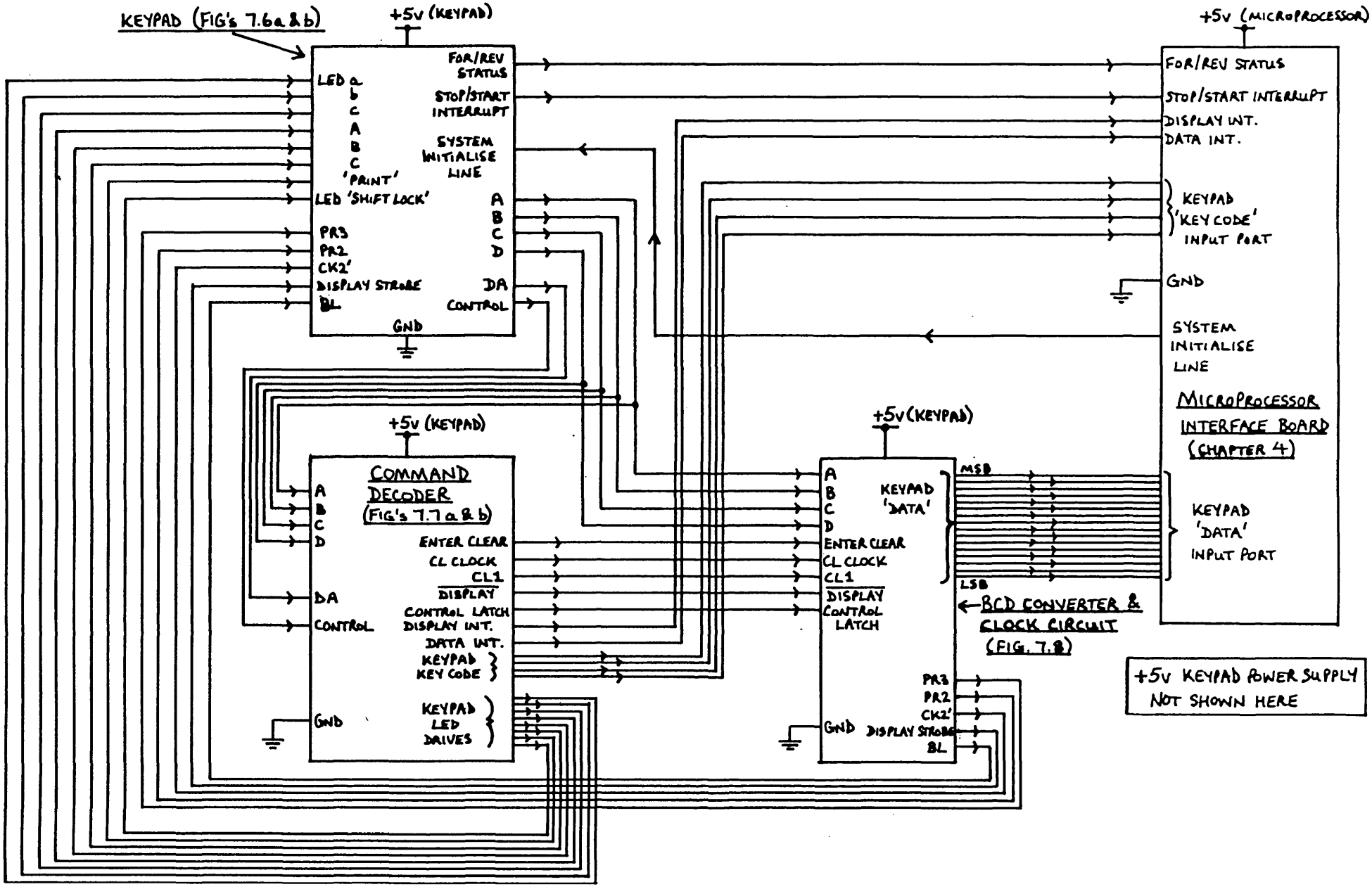


FIG. 7.10. THE INTERCONNECTIONS BETWEEN THE KEYPAD, THE COMMAND DECODER, AND THE BCD CONVERTER CIRCUITS (ALSO SHOWING THE MICROPROCESSOR INTERFACE BOARD)

The circuit diagram of the recalled data display circuit is shown in fig. 7.11. As with the command decoder, this circuit had to be built on two eurocards as indicated on the diagram. The 16 bit data word to be displayed is latched into the display registers (IC1, IC2, IC3, IC4) by the control signal "data strobe". This input toggles a flip flop (IC5a) which records the request for a binary to BCD conversion. The $\overline{Q}$ outputs of the memory registers drive the LED's that form the hexadecimal format display. The Q outputs are fed to the 16 bit binary down counter (IC7, IC8, IC9, IC10). After a delay generated by a monostable (IC11a), the down counter is parallel loaded with the data from the display memory and the five decade BCD up counter (IC15 and IC16) is cleared. The 250 kHz clock circuit (IC13) is then enabled and clocks the down counter and the BCD counter simultaneously. This continues until the down counter reaches zero. The BCD counter then contains the converted number. The BCD counter consists of a 4 digit integrated circuit BCD counter (a ZN1040)* plus a decade counter (74LS90) which deals with the most significant digit. The ZN1040 drives the LED displays directly, but the 74LS90 requires a seven segment drive circuit (IC17). A 74LS247 was chosen as it generates the same format sixes and nines as the ZN1040 display drivers.

The data key word is latched from the data bus by the TIL311 (IC6). This integrated circuit includes a latch, a decoder and a hexadecimal display.

The small modifications needed in all the circuits to ensure correct operation, have necessitated the use of some transistors and diodes. This was because insufficient space was available on the circuit boards to allow extra logic integrated circuits to be fitted. The voltage supply rails in figs 7.6a, 7.6b, 7.7a, 7.7b, 7.8, and 7.11 were well decoupled.

The circuit diagram of the keypad regulated +5 volt power supply is shown in fig. 7.12. The circuit is a standard design employing an integrated regulator circuit. The

* Details of the ZN1040 are given in RS data sheet 2848 (7.4)

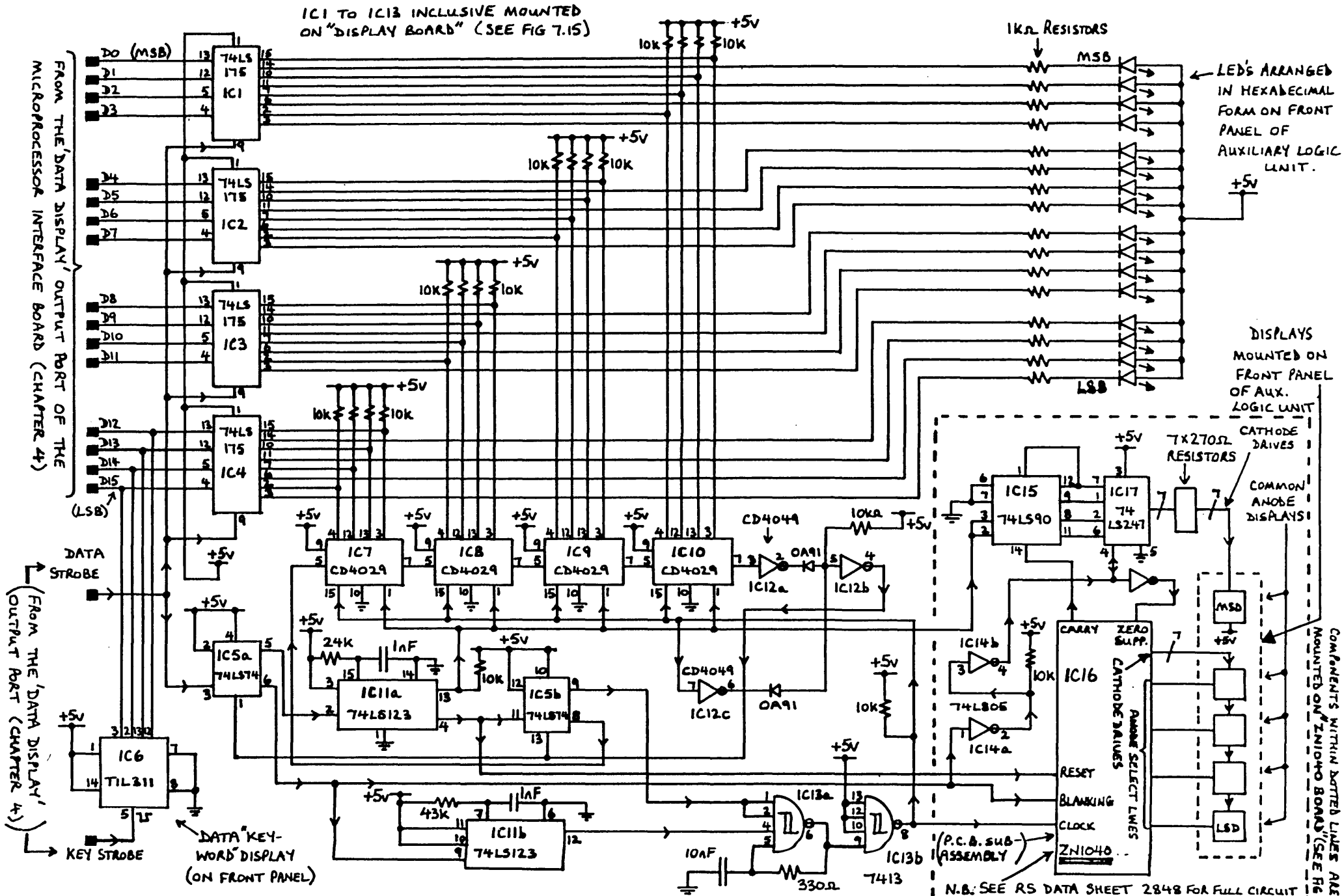


FIG. 7.11. THE RECALLED DATA DISPLAY CIRCUIT

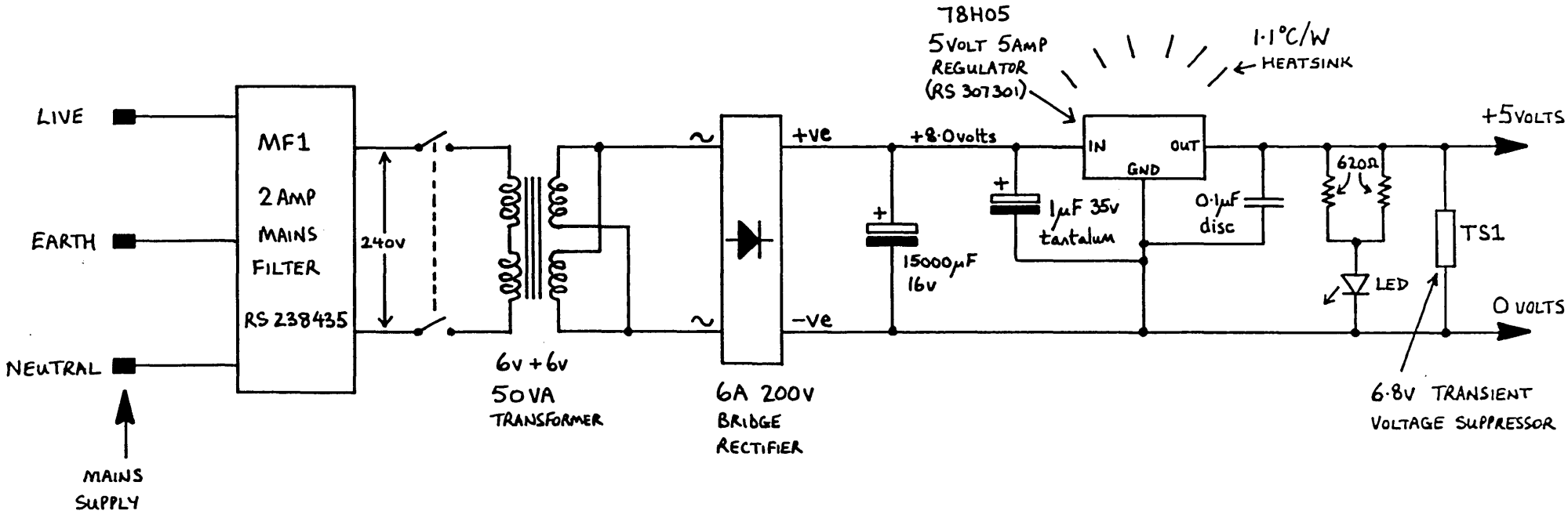


FIG. 7.12. THE +5VOLT 5AMP REGULATED SUPPLY USED TO POWER THE KEYPAD CIRCUITS (FIGS 7.6, 7.7, 7.8) AND THE DATA RECALL DISPLAY CIRCUIT (FIG. 7.11) (POWER SUPPLY HOUSED IN THE KEYPAD AUXILIARY LOGIC UNIT)

supply is capable of supplying 5 amps. The mains filter (MF1) and transient suppressor (TS1) were incorporated to reduce the chance of transient mains interference appearing on the data ready*, data recall† and stop/start interrupt lines. The power supply is mounted in the auxiliary logic unit.

Photographs of the keypad, the keypad circuit boards, the auxiliary logic unit boards and the auxiliary logic unit are shown in figures 7.13, 7.14, 7.15 and 7.16 respectively.

[*"DATA" INTERRUPT, †"DISPLAY" INTERRUPT } CHAPTER 4]

7.6 Software Required to Service the Keypad Unit

The basic software logic required to use the keypad is described below in flow chart form. The assembly code realisation of the flow charts is not given in this chapter. It can be seen in the motor control program listing given in Appendix 4A . Further details are given in Chapter 4.

The microprocessor only reads data from the keypad when it is interrupted. Three interrupts are generated by the keypad. They are the stop/start interrupt, the data available interrupt and the data recall interrupt. The maximum times taken to execute their interrupt service routines are 86μs, 78μs, and 89μs respectively. The flow charts for the stop/start, data available and data recall interrupt service routines are shown in figures 7.17, 7.18 and 7.19 respectively.

7.7 Operational Problems

When initially using the keypad, two minor problems were encountered.

(a) On entering the number 65535_{10} a data available

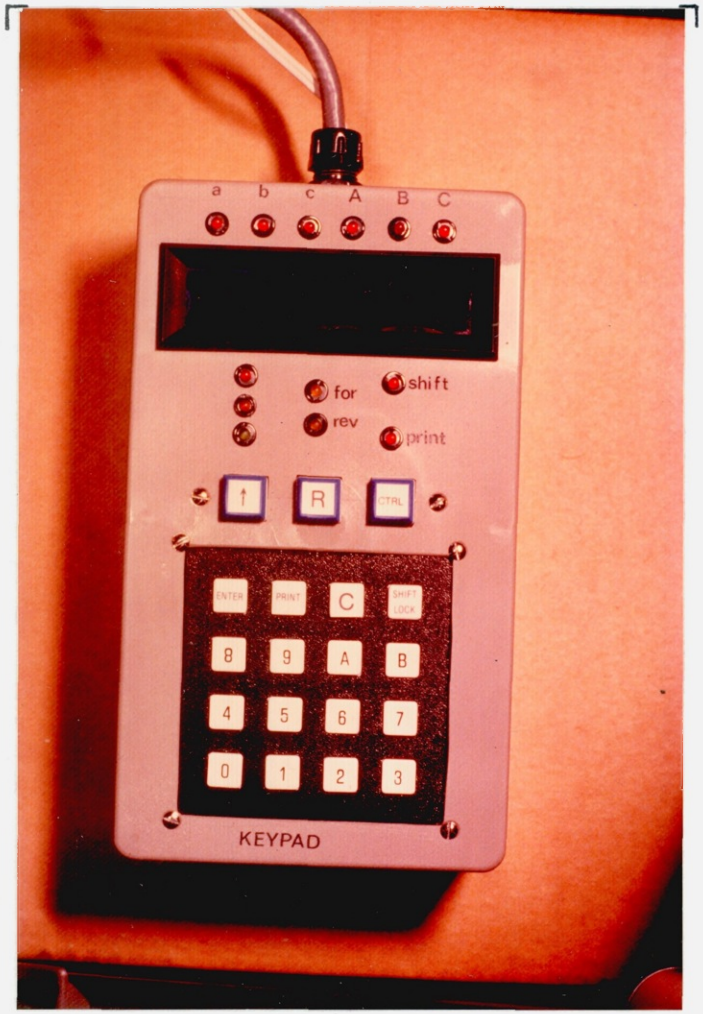


FIG. 7.13. PHOTOGRAPH OF THE KEYPAD UNIT

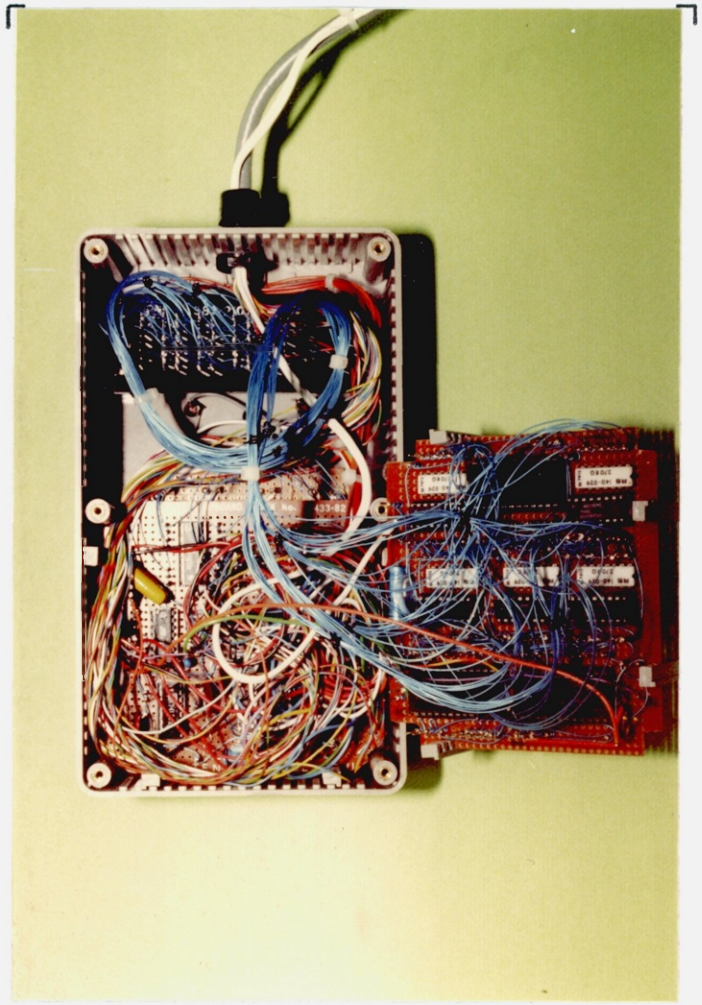


FIG. 7.14. PHOTOGRAPH OF THE KEYPAD CIRCUIT BOARDS

FIG. 7.15. PHOTOGRAPH OF THE AUXILIARY LOGIC UNIT KEYPAD-RELATED CIRCUIT BOARDS. THE DISPLAY BOARD AND THE ZN1040 BOARD FORM THE RECALLED DATA DISPLAY UNIT CIRCUIT.

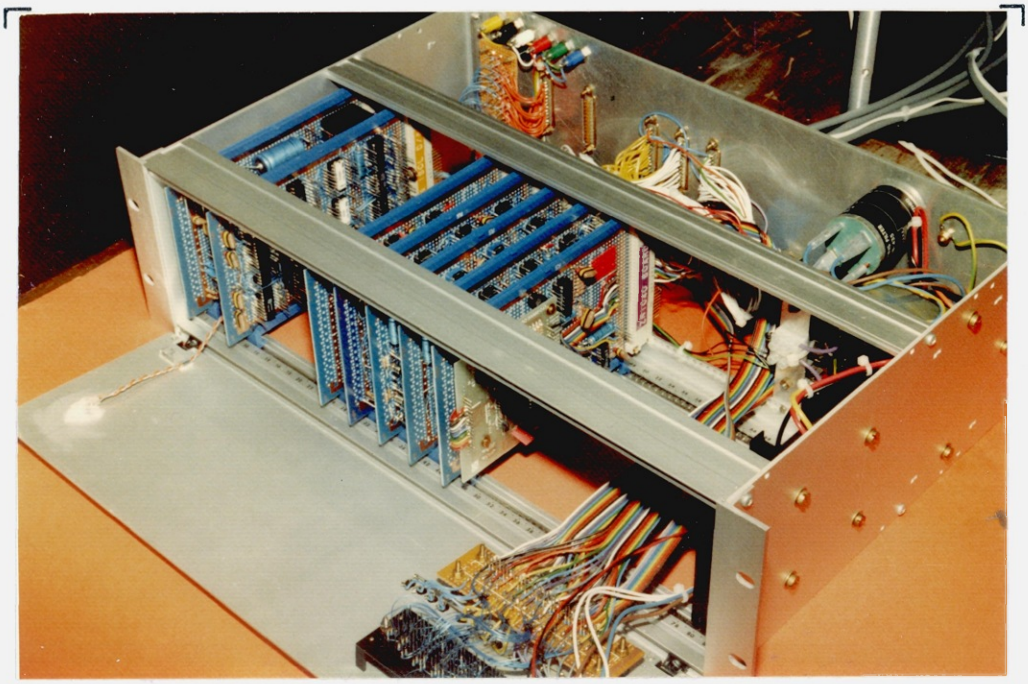


FIG. 7.16. PHOTOGRAPH OF THE KEYPAD AUXILIARY LOGIC UNIT. THE CENTRALLY LOCATED BOARDS ARE THOSE SHOWN IN FIG. 7.15. THE TWO BOARDS ON THE EXTREME LEFT ARE THE CIRCUITS RESPONSIBLE FOR CONTROLLING THE 60 VOLT PROGRAMMABLE POWER SUPPLIES (SEE CHAPTER 5)

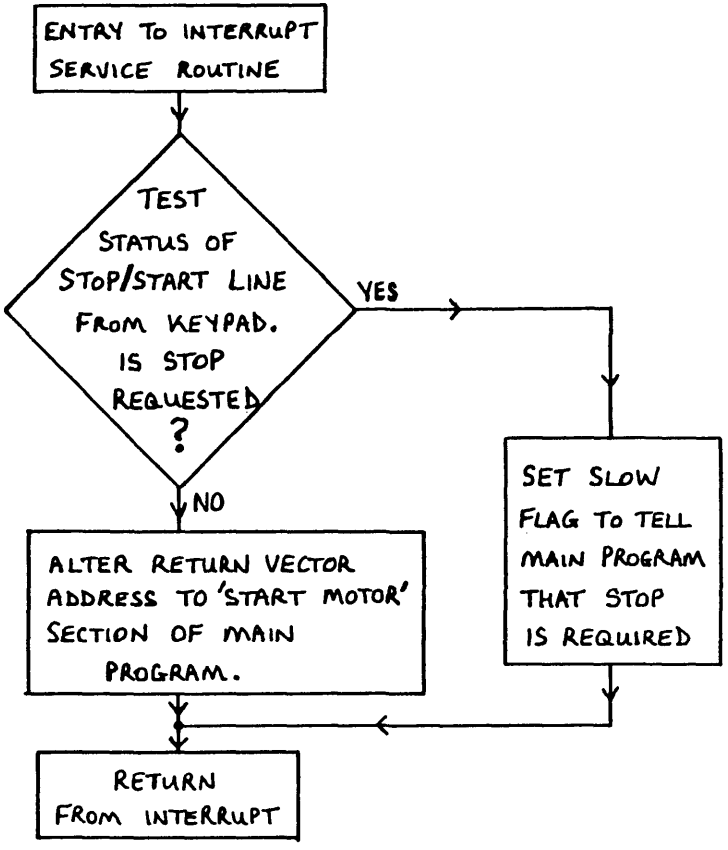


FIG. 7.17. BASIC FLOWCHART FOR THE KEYPAD-REQUESTED STOP/START INTERRUPT SERVICE ROUTINE.

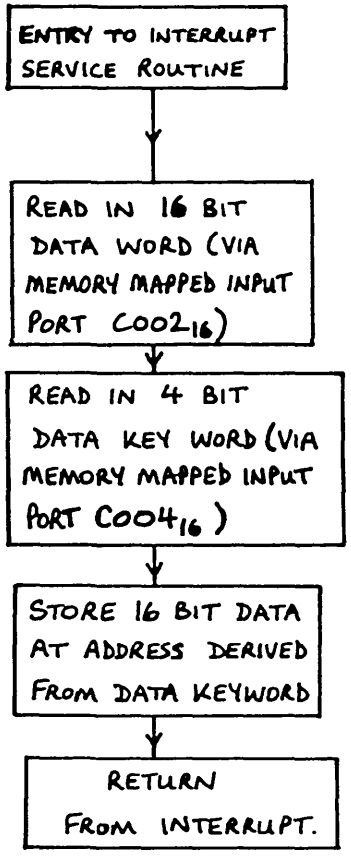


FIG. 7.18. BASIC FLOWCHART FOR THE KEYPAD-REQUESTED DATA AVAILABLE ("DATA") INTERRUPT SERVICE ROUTINE

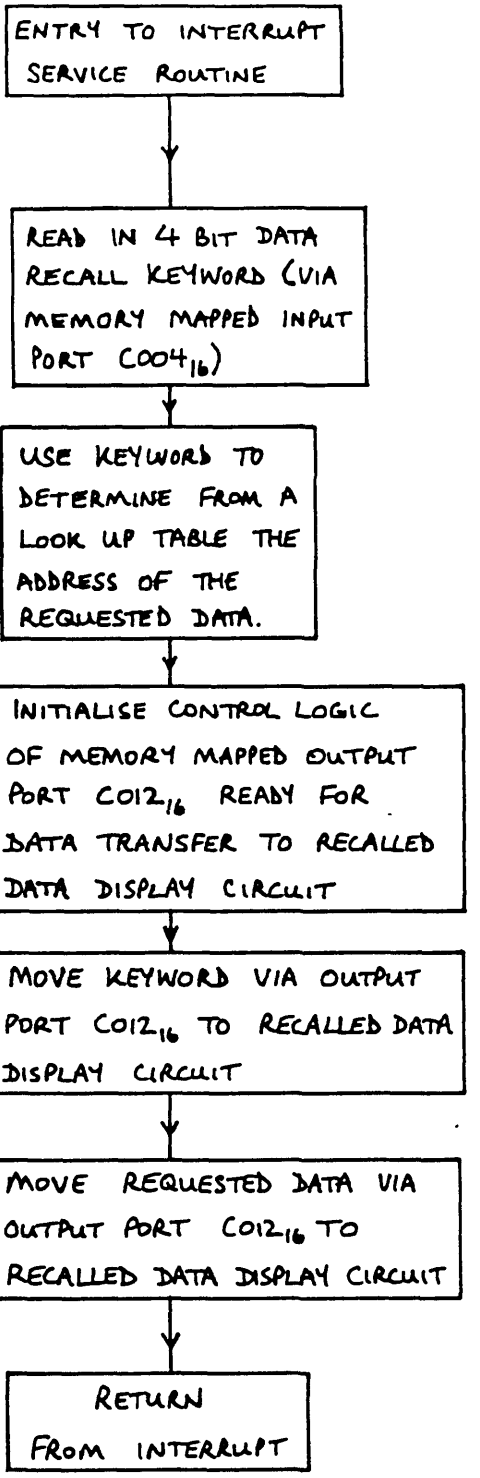


FIG. 7.19. BASIC FLOWCHART FOR THE KEYPAD-REQUESTED DATA RECALL ("DISPLAY") INTERRUPT SERVICE ROUTINE

interrupt was generated as the last character was pressed.

(b) On entering the numbers 11111_{10}, 22222_{10}, or 44444_{10}, data available and data recall interrupts were generated as the last characters were pressed.

It was eventually discovered that pick-up between the various lines in the cables connecting the auxiliary keypad unit to the microprocessor interface board was responsible for both problems. The inter-line capacitance was coupling fast edges on the 16 bit binary bus into the interrupt lines, as data on the 16 bit data bus rapidly changed state during a BCD-to-binary conversion. The induced glitches were sufficient to trip the interrupt latches. The addition of RC filters and schmitt triggers on the microprocessor interface board filtered out the glitches and solved the problem, although it would probably have been better to have designed the system with the interrupt lines in a separate screened lead.

7.8 Conclusions

The keypad unit described in this chapter has performed reliably for a period exceeding six months. It has been found to be convenient in use and the limited number of keys minimise the number of entry errors. The fact that the microprocessor only has to spend very short times servicing the keypad interrupts is a great advantage in a real time motor control system, and during the tests it was always found to be able to enter data and commands, even at the highest motor speeds when the microprocessor is under the greatest load. It is concluded that a keypad is a sensible choice of data peripheral for use in this and other microprocessor controlled motor systems.

CHAPTER 8

DESIGN AND DEVELOPMENT OF A HIGH RESOLUTION DIGITAL TACHOMETER

8.1 Introduction

Very often in industrial processes it is necessary to be able to measure the speed of a rotating shaft to a reasonable degree of accuracy. Traditionally the speed measurement has been made with either electro-mechanical tachometers employing eddy current effects to move a pointer over a scale, or by tachogenerators that produce a d.c. voltage proportional to the speed of rotation.

An increasingly important application for tachometers nowadays, is in speed regulated motor drive systems. This chapter examines the requirements for a tachometer used in such a system and discusses the possible circuit techniques available. A description of a digital tachometer suitable for use in a microprocessor controlled motor drive system is then given.

8.2 Requirements for a Tachometer Used in a Motor Drive System

The specification that a tachometer should meet in order to be classed as a good Speed sensing peripheral can be summarised as follows.

(i) It should be able to operate over a wide speed

range without, if possible, the need to switch ranges.

(ii) It should be as accurate as possible for a given cost and circuit complexity.

(iii) It should have good resolution so that accurate fine speed control is possible. Note that a tachometer can have a poor resolution and yet be highly accurate. For example, a tachometer with a resolution of only 100 rpm may well be completely accurate whenever the input speed is a multiple of 100 rpm, but for any other input speed it is inaccurate to a varying extent.

(iv) It should be fast acting in order that it can follow rapidly changing shaft speeds. The only way to achieve tight speed control in feedback systems is to have a tachometer that responds quickly to any change of shaft speed.

(v) It should provide output signals that are directly compatible with the system in which it is to be used. For example, if a human operator is to monitor the speed, the reading should be given directly in relevant units and on a clear, easily read display. If the tachometer is used in an analogue feedback system then its output should be an analogue signal. Similarly, it is better to have a direct digital output if a digital controller is used.

(vi) It should be stable over normal working temperatures.

(vii) The direction of rotation should not affect the accuracy of the reading.

8.3 Available Techniques for the Implementation of a Tachometer

There are several techniques that can be used for the electrical measurement of speed. They may be split basically into three groups encompassing analogue methods, digital methods and hybrid methods, in which either analogue and digital circuitry is combined or microprocessor software plus some circuitry is used. The three groups are examined in the following sections. The temperature stability of the various alternatives is not considered since in commercial tachometers, suitable compensation is usually included to counteract it. The choice of tachometer depends entirely on the system it is to be used on, and in many cases a very simple tachometer can be used. It is true that with an increasing number of digital systems in use today, digital tachometers are becoming more popular.

8.3.1 Analogue Tachometer Circuits

An analogue tachometer circuit converts the rotational speed of a shaft into a d.c. voltage or current, whose amplitude is proportional to the speed. Three methods of achieving this are the tachogenerator, the frequency to voltage (F/V) converter, and the phase locked loop.

8.3.1.1 Tachogenerators

A tachogenerator is a small precision made d.c. generator which provides an output voltage proportional to speed. It must be mechanically coupled to the shaft whose speed it is monitoring and consequently because of friction and windage, it does absorb some power from the shaft. The commutator and brushes are a source of unreliability and there is some ripple on the output voltage. The inertia of the generator can be a problem in small systems. However, tachogenerators are popular because they are relatively easy to understand and repair, they are relatively linear,

and they provide signals suitable for many applications at present. In addition the output voltage can vary very quickly, unlike some electronic circuit tachometers.

8.3.1.2 Charge Pump Frequency to Voltage Converters

Shaft rotation transducers are available that provide a fixed number of digital pulses every revolution of the shaft. The rate at which the pulses are generated is proportional to the speed of the shaft. The pulse rate can be converted into an analogue voltage or current by using a frequency to voltage converter. These circuits employ a charge pump technique. Every time a pulse is received, a capacitor is given a small burst of charge. The capacitor is continuously being discharged at a fixed rate and so the average voltage across the capacitor depends on the rate at which pulses of charge are fed into it. (i.e. the voltage is proportional to the speed.) There are several integrated circuits available which with the addition of a few passive components enable such a tachometer to be built. Examples include the National Semiconductor LM2907N8, and the Teledyne 9400.

However, although these circuits are easy to build, they do suffer from some non-linearity over a wide speed range and there is ripple on the output voltage which gets progressively worse as the input frequency falls. The speed range over which the circuit can work is also usually fairly limited, a ten to one ratio being typical. The smoothing circuits required to smooth the output voltage, to make it suitable for feedback use, give the tachometer a rather slow time constant and so the best application for this type of tachometer is perhaps simply driving a moving coil meter movement which averages the speed reading automatically.

8.3.1.3 Phase-Locked Loops

A phase-locked loop is a circuit in which the phase of the input frequency is compared with the output frequency of a voltage to frequency converter (V/F). The phase difference is used to generate an error voltage which actually controls the voltage to frequency converter. Hence, when the system is locked, it can track the input frequency and the error voltage varies proportionally with the input frequency. V/F converters can be produced with very linear voltage to frequency characteristics and the phase-locked loops can track an input frequency over a 20 to 1 frequency range. Integrated circuits are available containing the necessary circuit blocks to implement phase-locked loops. An example is the Signetics NE565A. Phase-locked loops do suffer from the problem of having initially to lock to the input frequency, and unfortunately the ability to lock over a wide range of input frequencies has to be traded off against the frequency range over which it will track. The phase-locked loop is nevertheless a useful alternative to the charge pump frequency to voltage converter.

8.3.2 Digital Tachometer Circuits

A digital tachometer circuit converts the rotational speed of a shaft into a digital code, usually binary or binary coded decimal, the value of which is proportional to the input speed. A digital tachometer requires a continuous series of pulses, usually obtained from a suitable speed transducer on the rotating shaft. The way in which these pulses are processed to give a digital speed signal determines the type of digital tachometer circuit. Digital tachometers are usually controlled by a quartz crystal oscillator which results in very stable, low drift performance. A feature common to all the digital tachometers discussed in this section is the fact that they can be implemented by hardware logic such as TTL and CMOS, or

equally well by software on a microprocessor.

8.3.2.1 Pulse Counting over a Fixed Timebase

This type of circuit uses a digital counter to count the input pulses coming into the tachometer. The period over which the counter is allowed to count is controlled by a timebase, which can be crystal frequency locked for accuracy. The control logic clears the counter and resets the timebase. The counter input is then enabled for a period determined by the timebase. The resulting number is then latched out into a storage register for display and other uses, and the tachometer commences another calculation. The counters used can be either binary or BCD, depending on the form of signal required.

The speed is obtained from the digital code as follows. If the speed transducer produces P pulses per revolution and the shaft is rotating at N rpm, the number of pulses per second, V, is given by:

$$V = \frac{N.P}{60.0} \qquad 8.1$$

If the timebase allows the counter to operate for a time T seconds, the resulting count, C, is given by:

$$C = V.T = \frac{N.P.T}{60.0} \qquad 8.2$$

By suitable choice of P and T it is possible to scale the count C so that its value is the speed in revolutions per minute. For example, if the disc produces 60 pulses per revolution and the timebase is 1 second, then C is calibrated in rpm.

The major problem with this system is the long count time required in order to achieve good accuracy and high resolution. The speed of a shaft can oscillate very rapidly, but with a long count time only the average speed is

calculated. To achieve a faster response rate it is necessary to reduce the count time, but this then requires an increase in the pulse rate from the shaft transducer if accuracy and resolution are to be maintained. This method can work over very wide speed ranges, limited only by the number of bits possessed by the counters, and the resolution obviously depends directly on the number of bits used. It gives linear results but obviously there is some discretization error due to the digital nature of the output signal. This method is widely used in practical systems (8.1), and is suited to microprocessors which offer programmable timers as part of the system.

8.3.2.2 Period Timing Between Two Pulses

In this technique a control circuit clears a counter. When enabled, the counter is clocked by a crystal oscillator. At a certain distinctive point during one of the input pulses from the speed transducer, the counter is enabled and starts to count up, clocked by the oscillator. At the identical point on the next input pulse, the counter is disabled. The count that it then holds is proportional to the time between two adjacent speed pulses. The easiest repeatable points that can be detected in a digital speed signal are either the rising or falling edges. Thus the counter could be controlled by successive rising edges.

The speed is calculated from the digital code held in the counter as follows. If the period between two rising edges on the speed signal is T seconds, and the counter is clocked by an oscillator of frequency f Hertz, then the resulting count C is given by:

$$C = f.T \qquad 8.3$$

Now if there are P pulses per revolution of the shaft, and the speed of rotation is N rpm, then the time T in seconds is given by:

$$T = \frac{60.0}{N.P} \qquad 8.4$$

$$\therefore \quad C = \frac{60f}{N.P} \qquad 8.5$$

It can be seen that the count C is inversely proportional to the speed N. This raises an immediate problem since it is not easy to form the reciprocal of a digital number.

The resolution and accuracy of this system depend on the clock frequency f, the number of bits in the counter, and the number of pulses per revolution N. Its accuracy is further limited by the need to calculate the reciprocal. It is also true to say that speed control algorithms can be modified to use the count directly rather than needing to form the reciprocal, so that to control a motor speed the algorithm would compare a measured time with a demanded time. It is only when a readout in rpm is required for a human operator, that the reciprocal must be formed. However, period timing is used in systems (8.2) and with counter-timer circuits available on microprocessor systems, it is reasonably easy to implement in software.

8.3.2.3 Digital Frequency Comparators

A digital frequency comparator is a circuit which can compare two input frequencies and provide information about their relative magnitudes. For example for typical frequencies of A and B, logical outputs for A<B, A>B and A=B might be provided.

The A input could be supplied from a speed transducer on a motor shaft and the B input with a known reference frequency. The circuit would then indicate A=B whenever the motor speed was such that the speed transducer's frequency equalled the known frequency.

Such comparators can be used in speed control systems

(8.3). When the A<B output is active, power is applied to the motor to accelerate it and when the A>B output is active, the power is cut off. The motor hunts about the desired speed and with suitable electrical and mechanical damping, the system can be made stable.

Typical digital frequency comparators are shown in schematic form in figures 8.1(a) and 8.1(b).

The circuit shown in fig. 8.1(a) is suitable for presettable binary down counters that produce a signal when they reach their minimum count. Such a counter is the 4 bit CMOS CD4029. The CD4029 produces a low level at its CARRY OUT output when it reaches zero and can be cascaded to make counters with 4n bits.

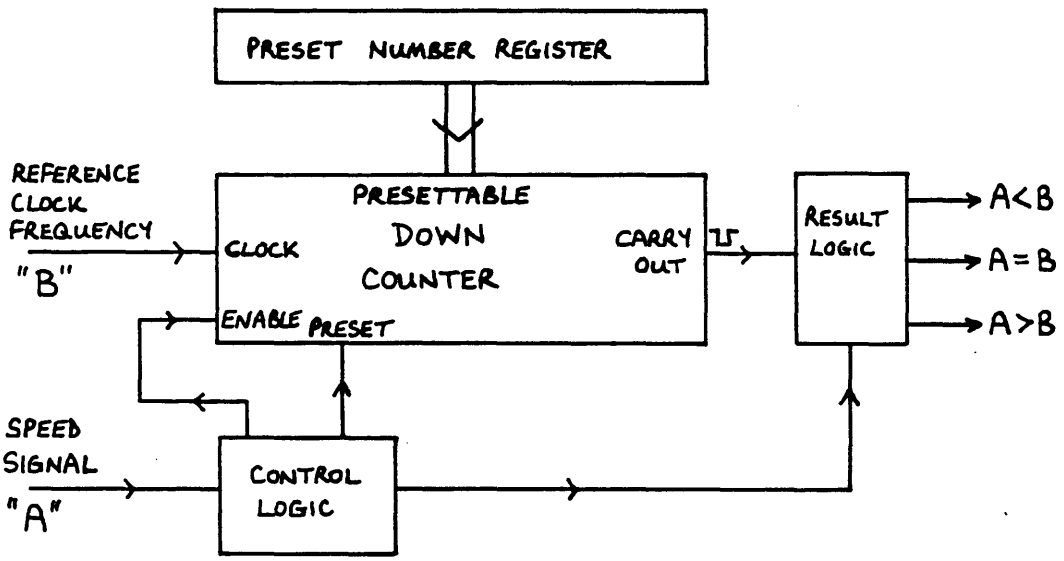


FIG. 8.1a. SCHEMATIC FORM OF A DIGITAL FREQUENCY COMPARATOR BASED ON A PRESETTABLE BINARY DOWN COUNTER.

The sequence of events over one period of the speed signal A is as follows.

The rising edge of the speed signal A is used to preset a number into the counter. The reference clock frequency is then enabled to clock down the counter towards

zero. If the preset number is chosen correctly for the speed it is wished to detect, then the counter reaches zero exactly in the time between two rising edges of the speed signal, and the carry out signal in combination with the control logic produces an A=B output. If, however, the speed is high, then the counter will not reach zero in the given time and a carry out signal will not occur. In the absence of a carry out signal the control logic can generate an A>B signal. Alternatively if the speed is low, the counter reaches zero and underflows. The carry out therefore goes low momentarily. Therefore at the next rising edge of the speed signal, the control logic sees that a carry out has occurred but is no longer active and so generates an A<B output.

The control circuit performs a comparison between each consecutive pair of rising edges on the speed signal.

The circuit shown in fig. 8.1(b) is suitable for counters which do not provide suitable signals to indicate the end of the count cycle. A magnitude detector is used to determine whether the count is greater than, less than, or equal to the comparison number at the end of the speed signal period.

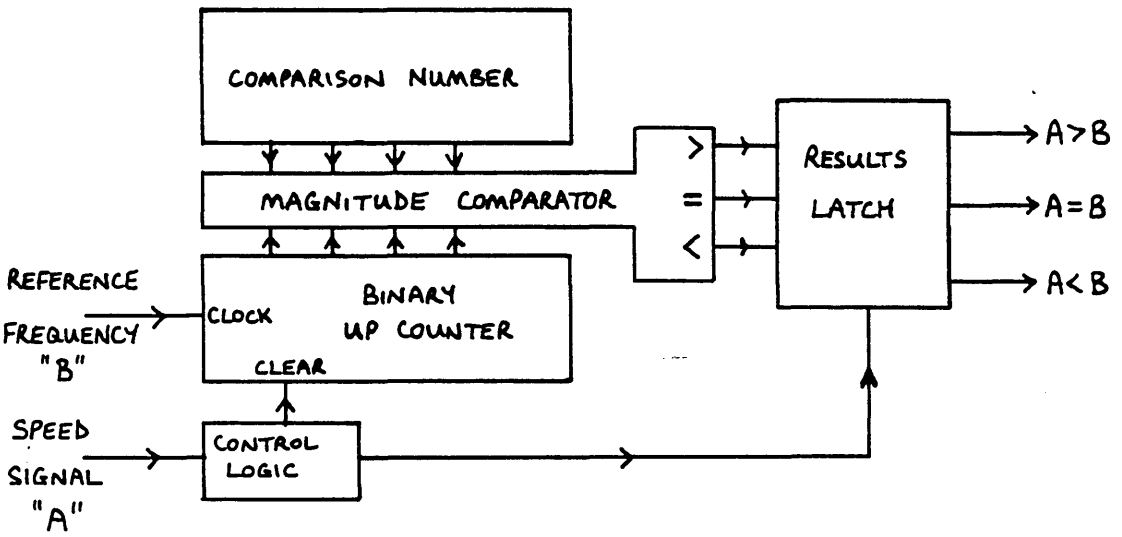


FIG. 8.1b. SCHEMATIC FORM OF A DIGITAL FREQUENCY COMPARATOR BASED ON A MAGNITUDE COMPARATOR.

Fig. 8.1(b) shows a binary UP counter such as the TTL 7493. The rising edge of the speed signal A is used to clear the binary counter. The counter is then clocked by the reference frequency. The magnitude comparator compares the count with the comparison number and produces the required magnitude signals at the end of the speed signal period.

This method is slightly more flexible than the one shown in fig. 8.1(a), since the comparison number can be changed as the counter is counting. This may be an advantage in applications where the system attempts to home in on the correct speed.

To detect speeds over a wide range it is necessary to vary the comparison frequency, which in practice means changing the preset number in the systems shown in figures 8.1(a) and 8.1(b). However, a tachometer that requires an operator to select various numbers until a suitable match is found is both slow acting and expensive to run. The digital frequency comparator can be adapted into a fast response, highly accurate digital tachometer by using the A>B, A=B and A<B signals to force the preset number towards the required value for the A=B condition. The system starts in a random state and iterates towards the correct speed. In effect, the system calculates the speed and then uses the new calculated value of speed in its comparison with the actual speed, so that it can further improve the calculated speed value. This is unlike the pulse counting or period timing digital tachometers which start from scratch in each measurement interval. A frequency comparator tachometer can therefore concentrate more on the small variations in speed rather than with the average speed. It can do this because the speed of a shaft cannot change instantaneously, and since it can only change by a relatively small amount in the measurement interval, it is better to measure the change from the expected speed.

A tachometer system that employs such a method was developed in Sweden by Professor V. Török of the Royal Institute of Technology, Stockholm, and reported in a paper (8.4). A frequency to digital converter forms the basis of the system. The manner of operation is most easily described by summarising part of the paper.

A stream of pulses from a shaft speed sensor form the input to the tachometer. For each pulse there is an expected moment at which it should occur, with the time interval between the previous pulse and the expected pulse being inversely proportional to a stored "measurement" number corresponding to the last (estimated) value of the speed. When the new speed pulse arrives, the speed estimate is adjusted by an amount depending on the difference between the observed and predicted length of the interval, thus improving the measurement iteratively. Fig. 8.2 shows that the difference is measured by subdividing the expected pulse distance $T_{p,e}$ into a number of parts, N. This is achieved by generating an internal pulse train f_r having a frequency N times higher than the expected value of the speed pulse frequency $f_{p,e}$.

The internal pulse train f_r is generated by a digital rate multiplier (fig. 8.3), the clock input of which is driven by a known precise clock frequency f_c and the rate inputs of which are connected to a register containing a number Z. The internal frequency f_r is thus proportional to this measurement number Z in accordance with the equation:

$$f_r = \frac{Z}{Z_o} f_c \qquad 8.6$$

where f_c = the clock frequency and Z_o = the divisor of the digital rate multiplier.

The measurement number Z is adjusted by the difference between the expected number of internal pulses per

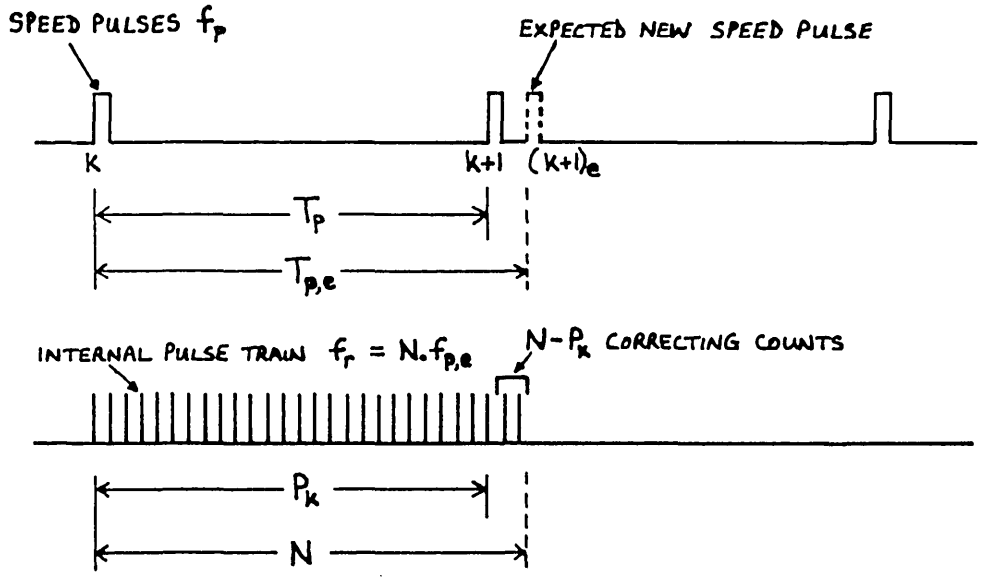


FIG. 8.2. ILLUSTRATION OF HOW THE SPEED ESTIMATE IS CORRECTED BY MEANS OF AN INTERNALLY GENERATED PULSE TRAIN FROM WHICH A CORRECTION COUNT IS DERIVED.

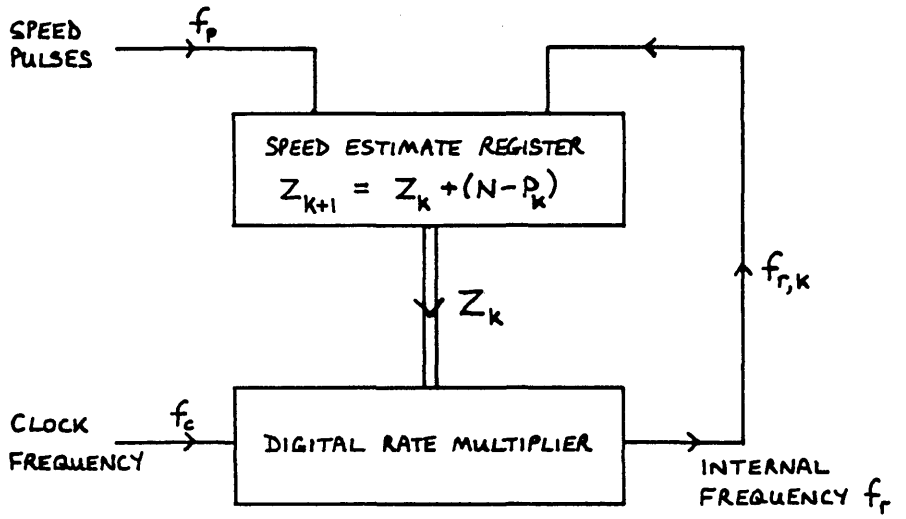


FIG. 8.3 SCHEMATIC DIAGRAM SHOWING THE USE OF A DIGITAL RATE MULTIPLIER TO GENERATE THE INTERNAL FREQUENCY f_r.

interval, N, and the actual number, P_k, so that the new measurement number will be:

$$Z_{k+1} = Z_k + (N-P_k) \qquad 8.7$$

where Z_k is the measurement number for the k^{th} measured interval (i.e. the old measurement number); and Z_{k+1} is the measurement number for the $(k+1)^{th}$ measured interval (i.e. the new measurement number).

The exact value of the measurement number Z_p for a given speed fulfils the condition N-P=0 (from equation 8.7). With the system in this state:

$$f_r = N. f_p \qquad 8.8$$

which when substituted into equation 8.6 gives:

$$\frac{Z_p}{Z_o} = N. \frac{f_p}{f_c} \qquad 8.9$$

The actual number of internal pulses in a measuring interval is given by:

$$P_k = \frac{Z_k}{Z_o} . f_c . T_{pk} \qquad 8.10$$

where $$T_{p,k} = 1/f_{p,k} \qquad 8.11$$

is the time interval between two speed signal pulses.

Substitution of equations 8.9 and 8.10 into equation 8.7 to eliminate N and P_k gives:

$$Z_{k+1} = Z_k + (Z_{p,k} - Z_k) \frac{f_c . T_{p,k}}{Z_o} \qquad 8.12$$

If the speed-pulse frequency and hence Z_p remain constant during the $(k+1)^{th}$ interval, the deviation in the measurement number can be evaluated by subtracting the predicted value Z_{k+1} given by the equation 8.12 from the actual value Z_p:

$$Z_p - Z_{k+1} = Z_p - \left[Z_k + (Z_{p,k} - Z_k) \frac{f_c \cdot T_{p,k}}{Z_o} \right]$$

Since Z_p is constant over the measurement interval, $Z_{p,k} = Z_p$, and $T_{p,k} = T_p$ giving:

$$Z_p - Z_{k+1} = (Z_p - Z_k) \left(1 - \frac{f_c \cdot T_p}{Z_o}\right) \qquad 8.13$$

If the speed pulse frequency remains constant for x intervals the result is:

$$Z_p - Z_{k+x} = (Z_p - Z_k)\, a^x \qquad 8.14$$

$$\text{where } a = \left(1 - \frac{f_c \cdot T_p}{Z_o}\right)$$

i.e. the measurement error vanishes exponentially, as shown in fig. 8.4.

The paper goes on to derive an equivalent time constant of the measurement system:

$$T_m = \frac{Z_o}{f_c} \text{ seconds.} \qquad 8.15$$

i.e. the time constant is unchanged within the entire range of measurement and is equal to the cycle time of the digital rate multiplier.

The invariable time constant is extremely desirable in control applications.

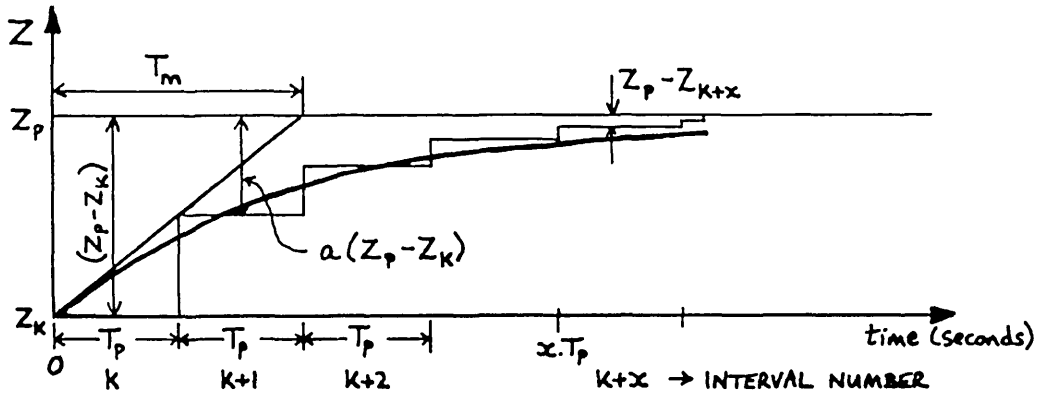


FIG. 8.4. ILLUSTRATION OF HOW THE SPEED MEASUREMENT ERROR REDUCES EXPONENTIALLY.

Further analysis shows that the measurement accuracy is constant through the entire range of measurement and the resolution Z_o is independent of the number of teeth on the speed transducer pulse wheel. The resolution from equation 8.15 is $Z_o = T_m.f_c$. Therefore, for a desired time constant T_m, the resolution is only dependent on the clock frequency f_c, and the measuring procedure is equivalent to a pulse counting procedure applied to the internal frequency f_r. This is N times more advantageous than pulse counting applied directly to the speed pulses. In practice N has a value between 10 and 1000.

The step response of the system is shown to be monotonic providing $T_p \leqslant T_m$ and this defines a lower limit speed for the system.

The analysis given in the paper ignores the influence of quantization in the algorithm (i.e. P and Z are integers). Furthermore, the internal pulse train becomes somewhat irregular when generated by a digital rate multiplier. However, it is stated that these factors result in a superimposed digital noise of only ± 1 unit in the number Z, and this is of no significance.

The paper reports that the method has been used with great success in industrial speed control applications.

This particular form of digital frequency comparator has many performance characteristics that are highly desirable but which are not all achievable with the pulse counting or period timing methods. The slightly more complicated circuitry is therefore easily justified and since it can be implemented in either hardware or software, it is no more difficult to build than a pulse counting or period timing circuit.

8.3.3 Hybrid Tachometer Circuits

Occasionally a tachometer is available on a motor drive system but the output signal that it provides is not of the correct form for the control electronics. For example, a system might be controlled by a digital computer using a digital speed control algorithm but the only speed signal might be an analogue voltage provided by a tachogenerator. The systems could be interfaced by using a sample and hold circuit followed by an analogue to digital converter to convert the speed signal into a suitable digital form.

The addition of the converter circuitry to the basic tachometer forms a hybrid device. Such hybrid devices are often very convenient especially when both analogue and digital signals are required in a system.

However, to eliminate conversion errors, it is preferable to use tachometers that provide the required form of output initially and so hybrid techniques will not be discussed further here.

8.4 The Chosen Tachometer System

The motor system described in this thesis is based around a microprocessor controller. The tachometer is therefore a peripheral of the microprocessor and the most

efficient method of data transfer between the two units is in digital form. Having considered the available digital tachometer methods, it was decided that the Török-Valis form of digital frequency comparator offered the greatest flexibility and best performance.

The following subsections detail the development of a practical, self-contained tachometer, which can be used on its own or as a peripheral with the microprocessor controller.

8.4.1 Basic Design Decisions

The first choice to be made was between a hardware or software realization of the chosen method. At the time that the tachometer was designed, it was not known to what extent the system microprocessor would be occupied by its basic task of maintaining the synchronous motor in synchronism. There was, therefore, a risk that if the tachometer was implemented on the microprocessor, there might be timing problems when the microprocessor was attempting to keep the motor synchronised at high speed. It was important to ensure that the main task of the microprocessor was not impeded by other jobs which could be done elsewhere. Therefore, it was decided not to implement the tachometer algorithm on the system microprocessor, although facilities were incorporated in the interface hardware to enable a software tachometer solution to be tried if desired. Due to lack of time this was not made use of and so it will not be pursued here. Some details of the software implementation are given in section 8.4.9.

Having decided that the tachometer should be an independent peripheral, the choice was then between using an independent microprocessor to implement a software solution or one of the popular logic families to implement a hardware version. At this stage of the project, far more experience was held in hardware design and so the software

solution was rejected. The hardware circuit was based on the one suggested in the paper by Török and Valis.

"..... the F/D converter comprises three main parts:

- a digital rate multiplier, where the internal frequency $f_{r,k}$ during each transmitter pulse interval is generated as a product of a known clock frequency f_c and the measurement number Z_k.

- a memory for the measurement number Z_k.

- an arithmetic unit where the new measurement number Z_{k+1} is calculated in accordance with equations 8.7 and 8.10

$$Z_{k+1} = Z_k + (N - \frac{Z_k}{Z_o} f_c \cdot T_{p,k}) \text{ "}$$

A simplified block diagram is shown in fig. 8.5.

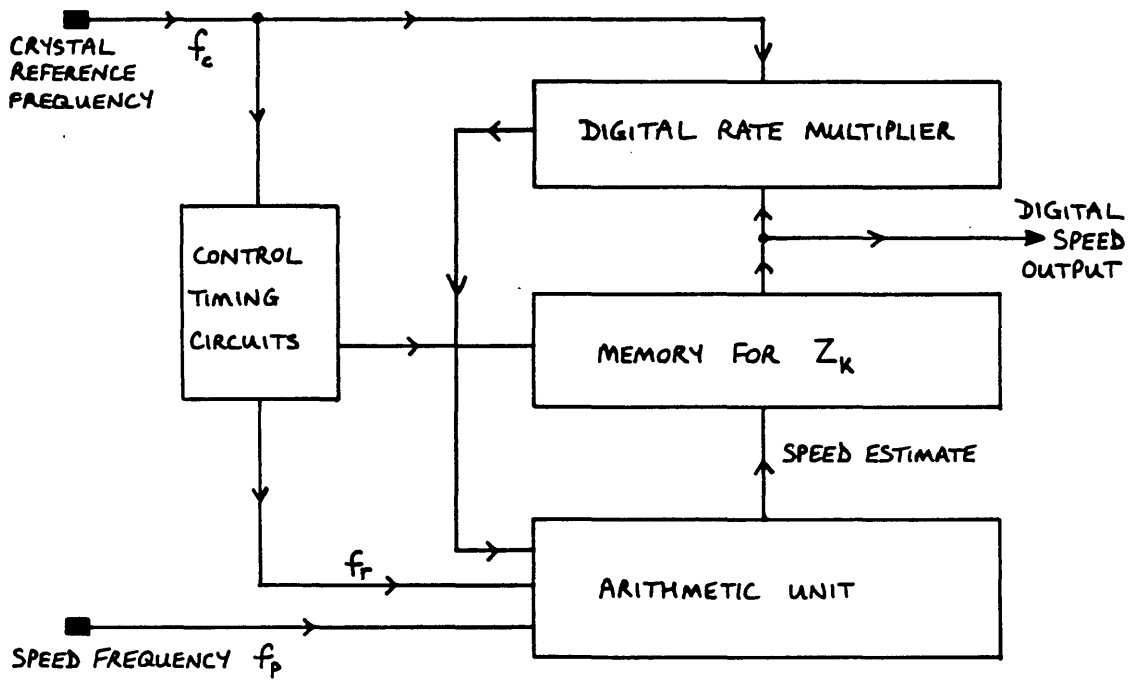


FIG. 8.5. SIMPLIFIED BLOCK DIAGRAM OF THE HARDWARE BASED DIGITAL TACHOMETER.

A very important choice that had to be made in the design of the tachometer was whether to arrange the circuits to provide binary coded outputs or binary-coded-decimal (BCD) outputs. It was required that the tachometer should provide a suitable digital signal for the microprocessor and also the signal should be suitable for display on a seven segment read-out in decimal format. The easiest solution would be to work with BCD signals, since these could be directly used by readily available LED displays and could also be input by the microprocessor. However the 9900 microprocessor has a 16 bit data bus and so only four BCD digits can be input in parallel at any time, placing a limit of 9999_{10} on the speed reading. This speed was less than the expected operating speed of the motor system. It would be possible to read in a larger BCD number in two parts, but the two operations required were thought undesirable in a real time microprocessor controller. Far more information can be conveyed on a 16 bit bus if pure binary code is used, the maximum number possible being 65535_{10}. This does, however, have the disadvantage that a binary to BCD converter is required to enable the speed to be displayed on the tachometer unit. Both binary and decimal rate multipliers, (BRM and DRM respectively), are available in the standard logic families and so a binary or BCD Török-Valis tachometer can equally well be implemented. The choice depends entirely on the application, and in the case where only a visual speed reading is required, the DRM approach minimises the amount of display hardware needed.

It was decided to use BRM's in the tachometer circuit so that speeds of up to 65535 rpm could be transmitted to the microprocessor in 16 bit binary code. To make full use of the available data bus, the tachometer was designed to use a 16 bit measurement number Z_k. In addition, it was decided that the clock frequency f_c and the speed frequency f_p should be chosen so that the measurement number Z_k was calibrated directly in rpm.

The number of speed pulses per revolution and the number of internal pulses per speed pulse directly affect the clock frequency f_c required by the system. The formula for the required clock frequency f_c is derived as follows. If the speed transducer is rotating at S revolutions per minute and produces R pulses per revolution, the number of speed pulses per second V is given by:

$$V = \left(\frac{S.R}{60} \right) \text{ Hz} \qquad 8.16$$

If the number of internal pulses to be inserted between each speed pulse is N, the frequency of the internal pulses, I, is given by:

$$I = N.V = \left(\frac{N.S.R}{60} \right) \text{ Hz} \qquad 8.17$$

The internal pulse frequency is produced by the binary rate multiplier. A 'G' bit BRM has a multiplication factor Q given by:

$$Q = \frac{Z}{Z_o} \qquad 8.18$$

where $0 \leqslant Z \leqslant (2^G - 1)$ and $Z_o = 2^G$.

Therefore for a particular required output frequency, the input frequency f_c to the BRM must be given by:

$$f_c = \left(\frac{Z_o}{Z} \right) . I$$

$$f_c = \left(\frac{Z_o}{Z} \right) \times \left(\frac{N.S.R}{60} \right) \text{ Hz} \qquad 8.19$$

The frequency f_c may be obtained from a higher frequency crystal oscillator of frequency f_h where $f_h = B.f_c$.

Hence the required clock frequency f_c is given by:

$$f_h = B. \left(\frac{Z_o}{Z} \right) x \left(\frac{N.S.R}{60} \right) \quad Hz \qquad 8.20$$

The selection of a suitable crystal frequency can be a problem since crystals are generally only available at popular frequencies. Therefore, the choice of the variables B and R can be crucial.

As stated earlier, it was decided to use a 16 bit measurement number (i.e. G = 16) and so the maximum rate number Z is $(2^G - 1) = 65535$. This rate number represents the speed in rpm and so the maximum speed that can be handled is 65535 rpm. The choice of the pulse rate variable R depends entirely on the construction of the speed transducer. For the magslip system described in Chapter 3, there are basically 6 pulses per revolution and for the 7 phase motor described in Chapter 5, there are basically 7 pulses per revolution. Therefore, for the magslip system R = 6 and for the 7 phase motor R = 7.

Substituting the various variables into equation 8.20 for the full speed condition and R = 6 gives:

$$f_h = B. \left(\frac{65536}{65535} \right) X \left(\frac{16 \times 65535 \times 6}{60} \right) \quad Hz$$

$$f_h = (104857.6\ B) \quad Hz$$

By selecting values of B equal to one, two, three, etcetera, it is possible to determine if f_h is a preferred crystal frequency.

For n = 40,

$$f_h = (40 \times 104857.6) \quad Hz$$

$$f_h = 4194304 \text{ Hz}$$

i.e. $f_h = 2^{22}$ Hz

This is a standard crystal used for time-keeping.

For R = 7 and the same full speed condition, the crystal frequency is given by:

$$f_h = B.\left(\frac{65536}{65535}\right)X\left(\frac{16 \times 65535 \times 7}{60}\right) \text{ Hz}$$

$$f_h = (122333.8\dot{6}7\ B) \text{ Hz}$$

A value of B = 49 gives:

$$f_h = 5994359.5 \text{ Hz}$$

This is very close to a standard crystal frequency of 6.0 MHz. The error incurred by using a 6.0 MHz crystal is only 0.094%. This is hardly worth thinking about since it only represents an error of 62 rpm in 65535.

Using equation 8.15, the time constant for the tachometer can be calculated.

For R = 6, $f_c = 104857.6$ Hz and $Z_o = 65536$;

therefore $T_m = \left(\frac{65536}{104857.6}\right)$ seconds

$$T_m = 0.625 \text{ seconds.}$$

If we assume five time constants are required for an exponential process to effectively reach its final value, then the tachometer will take 3.125 seconds to reach its final value with a constant input speed, but of course it is very close to the correct speed much sooner than this.

The limit speed of the system is given by $T_p = T_m$, where T_p is the time between speed pulses. For 6 pulses a revolution this gives a lower limit speed of 16 rpm. Therefore the tachometer will reliably track a speed between 16 rpm and 65535 rpm with a resolution of 1 rpm and a time constant of 0.625 seconds. The quantization noise places an error of 1 rpm on the output.

For R = 7, $f_c = 122333.867$ Hz and $Z_o = 65536$;

therefore $$T_m = \left(\frac{65536}{122333.867} \right) \text{ Hz}$$

$$T_m = 0.536 \text{ seconds}$$

giving a five-time-constant period of 2.68 seconds. The limit speed in this case is again 16 rpm, but the time constant is slightly faster.

The actual circuitry required to implement the complete tachometer system is shown in block diagram form in fig. 8.6.

The tachometer circuit, crystal timebase, binary to BCD converter and display multiplexer are described in sections 8.4.2, 8.4.3, 8.4.4 and 8.4.5 respectively.

The crystal board provides the necessary frequencies to run the whole system. The binary speed signal is calculated by the tachometer board. The binary data is converted into a five digit BCD form by the binary to BCD converter, and the BCD characters are multiplexed onto the LED display by the display multiplexer.

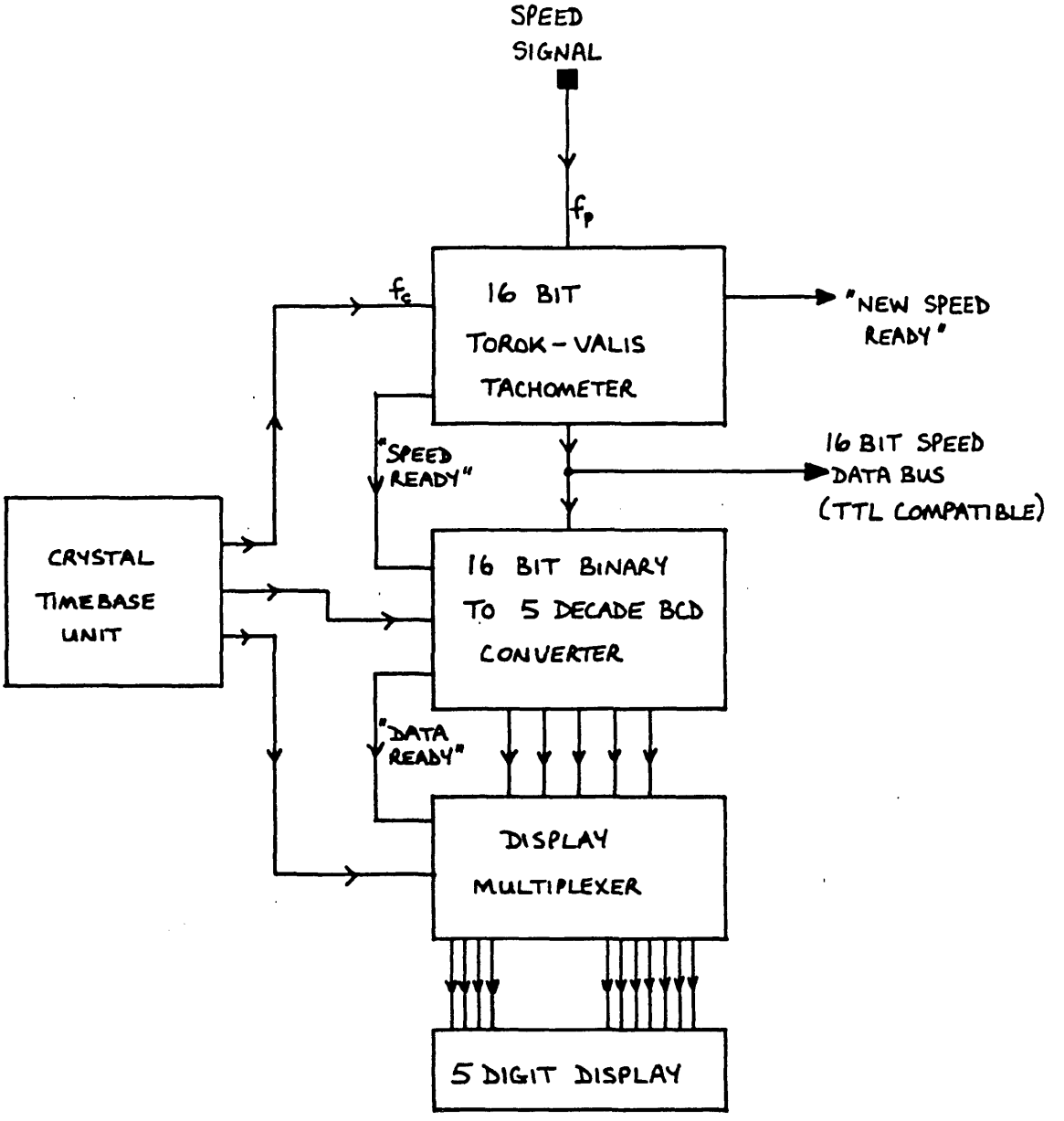


FIG. 8.6. BLOCK DIAGRAM OF THE DIGITAL TACHOMETER SYSTEM.

8.4.2 The Hardware Török-Valis Tachometer Circuit

The circuit diagram of the tachometer is shown in fig. 8.7 and it is based on a suggested circuit of Török and Valis. The majority of the circuit is built with TTL logic but the binary rate multipliers are CMOS CD4089 devices. (IC17, IC18, IC19, IC20.) It was necessary to use these rather than TTL because a 16 bit BRM was required. The TTL BRM integrated circuit is a 6 bit device and so larger BRM's implemented using it have to have integer multiples of 6 bits. Therefore it was not suitable for a 16 bit application, whereas the 4 bit CD4089 could be cascaded to form a 16 bit BRM. The only problem with using a CMOS BRM is that suitable interfacing is needed between the TTL and CMOS sections of the circuitry.

The old measurement number Z_k is stored in a 16 bit parallel register consisting of four 74LS175 TTL latches, (IC9, IC10, IC11, IC12). A 16 bit up/down binary counter consisting of four 74LS193 TTL counters, (IC5, IC6, IC7, IC8), is used as the arithmetic unit to calculate the measurement number Z_{k+1}. The counter is a "dead end counter" in order to prevent overflow of the measurement number and to set the lower limit of the frequency range to the limit frequency f_{pl}, which is equivalent to a speed of 16 rpm.

The operation of the frequency to digital converter is controlled by a six-phase control signal Q_1 to Q_6 which is generated in a ring counter. The ring counter is comprised of a series-in, parallel-out shift register, (IC2), and a NOR gate made up from open collector inverters, (IC 3a to f), and a nand gate, (IC16b). The ring counter shifts a logical "1" around at a shift rate seven or eight times higher than the desired clock frequency f_c, the rate being selected by a toggle switch. The two rates are required so that the tachometer can be used with speed transducers that produce six or seven speed pulses per revolution.

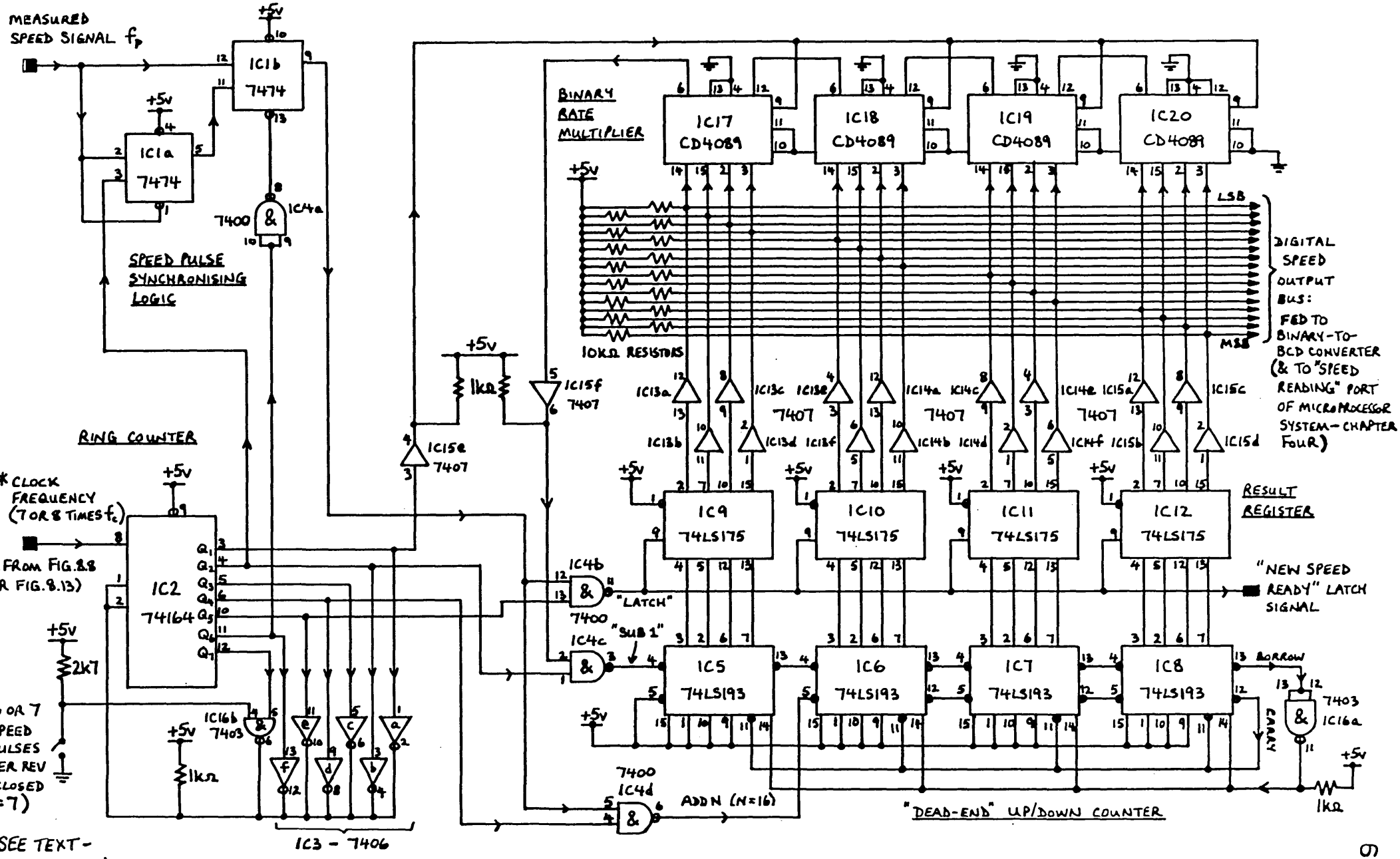


FIG. 8.7. CIRCUIT DIAGRAM OF THE 16 BIT "TÖRÖK-VALIS" DIGITAL TACHOMETER BOARD

The BRM is stepped forward one step during the Q1 phase.

Depending on its previous state and the current contents of the results register Z_k, the output signal of the BRM will be a logical "1" or "0". If the result is "1" the current contents of the counter will be decreased by 1 during the Q2 phase. At the same time the flip-flops IC1a and IC1b are set if the speed signal f_p has changed from logical "0" to "1"; i.e. if a new period has started for the speed pulse train whose frequency is being measured. Phase Q3 is an idle period. If flip-flop IC1b has been set, the number N is added to the contents of the counter during Q4 phase. The value of N is 16 since it is required that there are 16 internal pulses between each speed pulse. During phase Q5 the contents of the counter are copied into the result register and finally flip-flop IC1b is reset by phase Q6. If flip-flop IC1b has not been set during phase Q2, nothing happens during phases Q3 to Q6. If phase Q7 is selected it too is an idle period like Q3. There is one shift period at the end of the cycle when no output is active. This is when the feedback NOR gate generates a logic "1" for the serial input of the shift register. This shift period is also an idle period. The seven (or eight) phases are repeated cyclically at a frequency equal to the required clock frequency f_c of the frequency to digital converter. This means that during each period $T_{p,k}$ of the measured frequency, the contents Z_k of the counter will be gradually decremented by a total of N_d given by:

$$N_d = T_{p,k} \cdot \frac{Z_k}{Z_o} \cdot f_c \qquad 8.21$$

and at the end of the period will be increased by the number N = 16. This addition is conveniently achieved in this case by the ADD N line which operates the UP clock input of the second stage of the counter, thus incrementing the total counter contents by 16. Following the add-

ition, the contents of the counter will be copied into the result register as Z_{k+1}. Note that for a constant speed, the tachometer will stabilise such that $N_d = N$. The measurement number Z_k in the result register can assume any value between N and Z_o-1; i.e. 16 to 65535, thus giving an effective measurement range $f_{p\ min} : f_{p\ max}$ = 1:4096. It is interesting to note that the minimum speed that the system can display is actually the same as the lower limit speed calculated in section 8.4.1, and therefore full use is made of the tachometer measuring range. The latch signal generated on phase Q5 once every $T_{p,k}$ period is brought out of the circuit to be used as a "new speed ready" signal. The interfacing between the TTL results register and the CMOS BRM is achieved using open collector 7407 TTL buffers (IC13, IC14, IC15) with 10 kΩ pull up resistors. The digital speed output bus of the tachometer is also taken from these buffer outputs. The BRM clock signal and the BRM output pulse are similarly buffered. The "dead-end" up/down counter is prevented from counting down below zero or up above 65535 by feedback on to the clear and load inputs from the borrow and carry outputs respectively. The complete circuit was constructed on a 20 package dual in line (DIL) circuit board, and external connections were made via the gold plated edge connector.

8.4.3 The Crystal-Locked Reference Frequency Oscillators

The reference clock frequency, f_c, used by the tachometer circuit, must be stable and accurate to ensure good speed accuracy. This can easily be achieved by using a quartz crystal oscillator. Two crystal based frequencies are required to enable the tachometer to function with speed transducers that produce six or seven pulses per revolution. Suitable crystal frequencies are calculated in section 8.4.1 and are 4.194304 MHz and 6.0 MHz respectively.

For correct "6 pulse" operation, the crystal frequency

must be divided by forty to give f_c = 104857.6 Hz. The division ratio of forty is achieved in two stages. The crystal frequency is divided by five to give a frequency of 838860.8 Hz. This is then divided by eight by the ring counter on the tachometer board. The ring counter divides by eight when the toggle switch is set in the "6 pulse" position.

The circuit diagram of the "6 pulse" clock board is shown in fig. 8.8. The quartz crystal is connected in the feedback network around a single transistor. The output waveform is a distorted sinewave and this is squared up by a 74LS13 TTL schmitt trigger (IC1b). The schmitt trigger then clocks a TTL 74LS90 decade counter (IC2). This counter contains a divide by 5 and divide by 2 stage. The input frequency is first divided by 5 to give the 838860 Hz clock frequency required by the tachometer ring counter. The divide by 2 stage further divides the frequency to provide a 419430 Hz clock for use by the binary to BCD converter board. A 7493 TTL counter divides the 419430 Hz down to 26 kHz to provide the clock frequency for the display multiplex board. Thus all boards in the system are clocked by the central crystal oscillator.

The circuit was constructed on a DIL circuit board and occupied roughly a third of the board area. The crystal frequency was trimmed using the trimmer capacitor VC1. The signals were brought out on the edge connector.

A choice of techniques was available for the generation of the 122334 Hz clock frequency f_c, needed for the "7 pulse" operation of the tachometer. The techniques considered were:

(i) the construction of a crystal board similar to that shown in fig. 8.8, but using a 6.0 MHz crystal. The generated frequency then requires dividing by a factor of 49 and this can be achieved by an initial

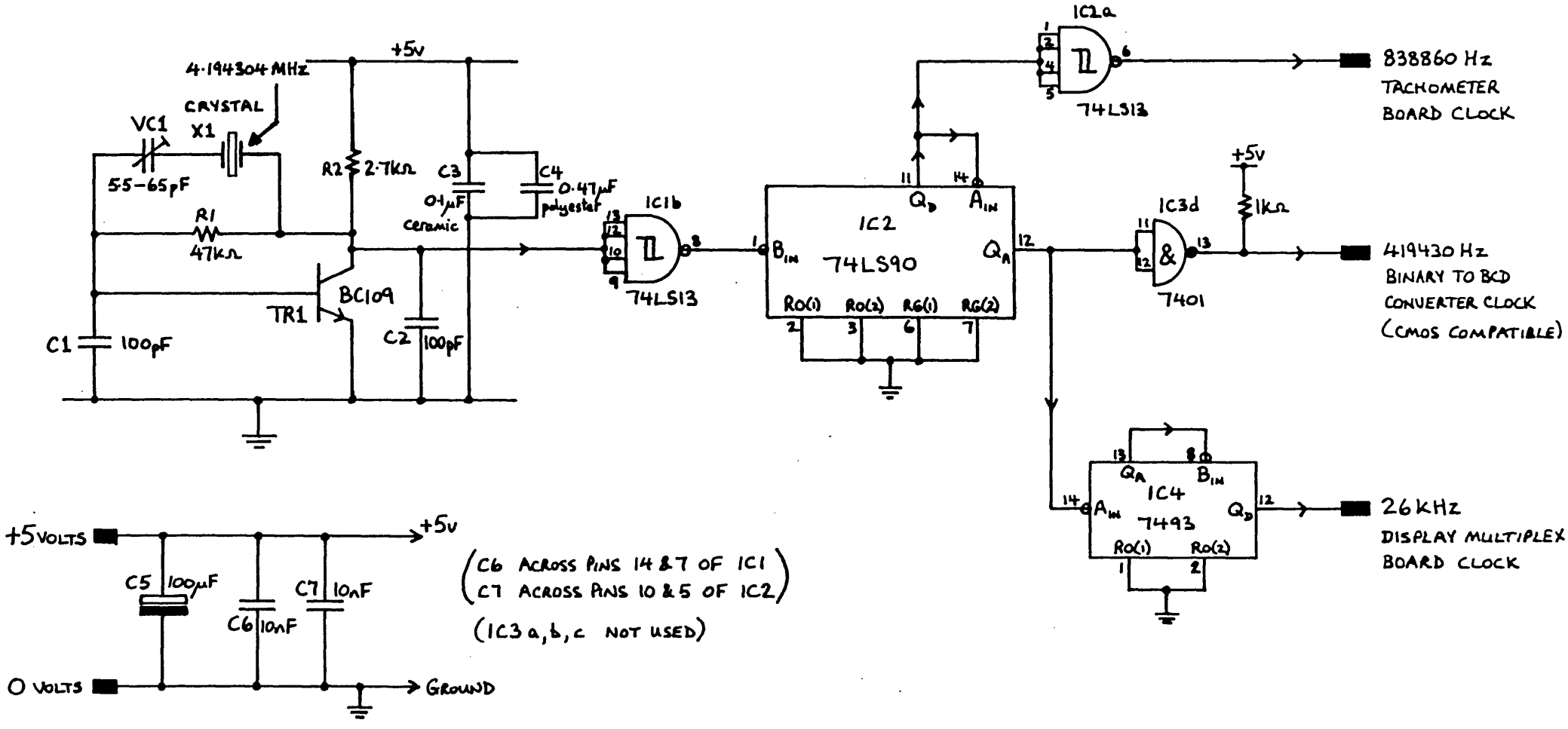


FIG. 8.8. THE DIGITAL TACHOMETER "6 PULSE" CLOCK BOARD.

division by seven, followed by a further division by seven performed by the tachometer ring counter (toggle switch in "7 pulse" position).

(ii) The scaling of the speed signal frequency f_p by a factor of 6/7 so that the "6 pulse" tachometer circuit and oscillator board can be used unmodified.*

(iii) The scaling of the "6 pulse" tachometer clock frequency f_c by a factor of 7/6 so that the "6 PULSE" tachometer and clock boards can be used unmodified.*

[* with toggle switch in "6 pulse" position : see fig. 8.7 — Ring Counter Circuit]

The initial hope was to avoid having to build a second crystal oscillator board, since a perfectly good crystal frequency was available even though it was of the wrong frequency. Therefore option (ii) was first considered and the block diagram of the solution is shown in fig. 8.9.

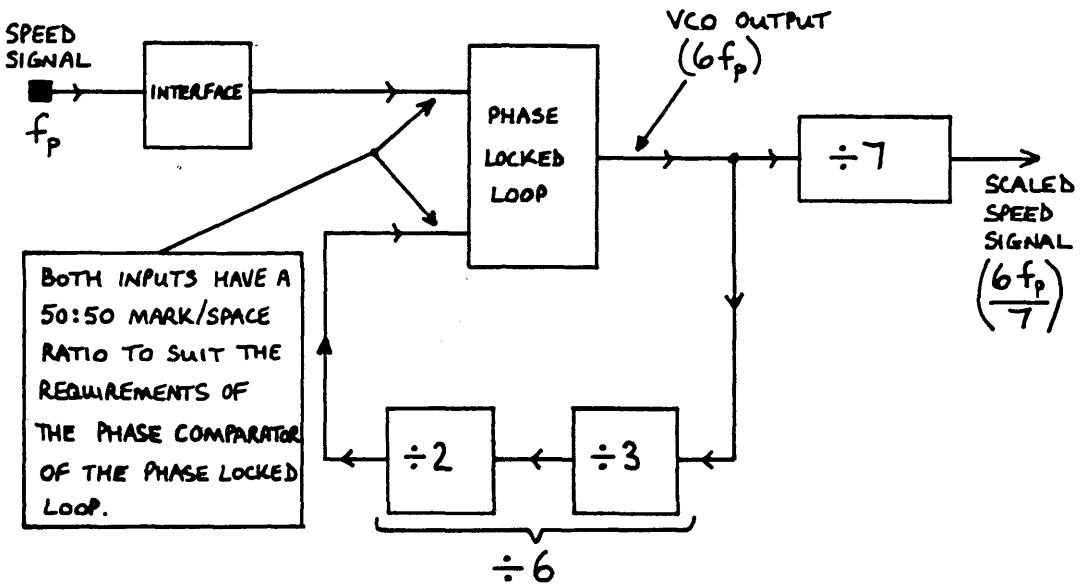


FIG. 8.9. BLOCK DIAGRAM OF A SCALING CIRCUIT TO ENABLE THE DIGITAL TACHOMETER TO OPERATE FROM A SHAFT TRANSDUCER PRODUCING A SPEED SIGNAL OF SEVEN PULSES PER REVOLUTION.

The basic task is to scale the speed signal frequency f_p by a constant factor of 6/7. The first problem, multiplying by six, is overcome by using a phase-locked loop (PLL) with a divide by six counter connected in the feedback loop between the voltage controlled oscillator (VCO) output and one of the phase comparator inputs. This causes the VCO to run at a frequency six times that of the input frequency, f_p. The divide by 2 section in the feedback ensures an even mark-space ratio for the feedback signal. This improves the operation of the phase detector. The divide by seven counter completes the conversion to give an output of (6/7) f_p.

Since the speed frequency varies over a wide range of at least 300 Hz to 2.5 kHz, the PLL must track the input frequency as it changes. The circuit was built using a NE565A PLL, with the counters being implemented by 7473 TTL JK flip-flops. The circuit is shown in fig. 8.10. It was very quickly discovered that the performance of the circuit was not adequate. The PLL could not lock and track frequencies over a wide enough range and there was an unacceptable amount of frequency jitter on the output. It was difficult to know where the centre frequency of the PLL should be fixed. The method was therefore abandoned but it was thought worthwhile trying to use a PLL in the fixed frequency conversion task outlined in option (iii).

The PLL no longer has to track a varying input frequency and so its frequency stability should be much better. A block diagram of the circuit to multiply the reference frequency f_x by a factor 7/6 is shown in fig. 8.11. The 419430 Hz signal available from the "6 pulse" oscillator board must first be divided by twelve to give a frequency of 34952.5 Hz. The PLL then multiplies this frequency by fourteen to give an output frequency of 489335 Hz at the VCO output. It was necessary to perform the division prior to the multiplication because the VCO has a maximum operating frequency of 500 kHz. The VCO output is finally

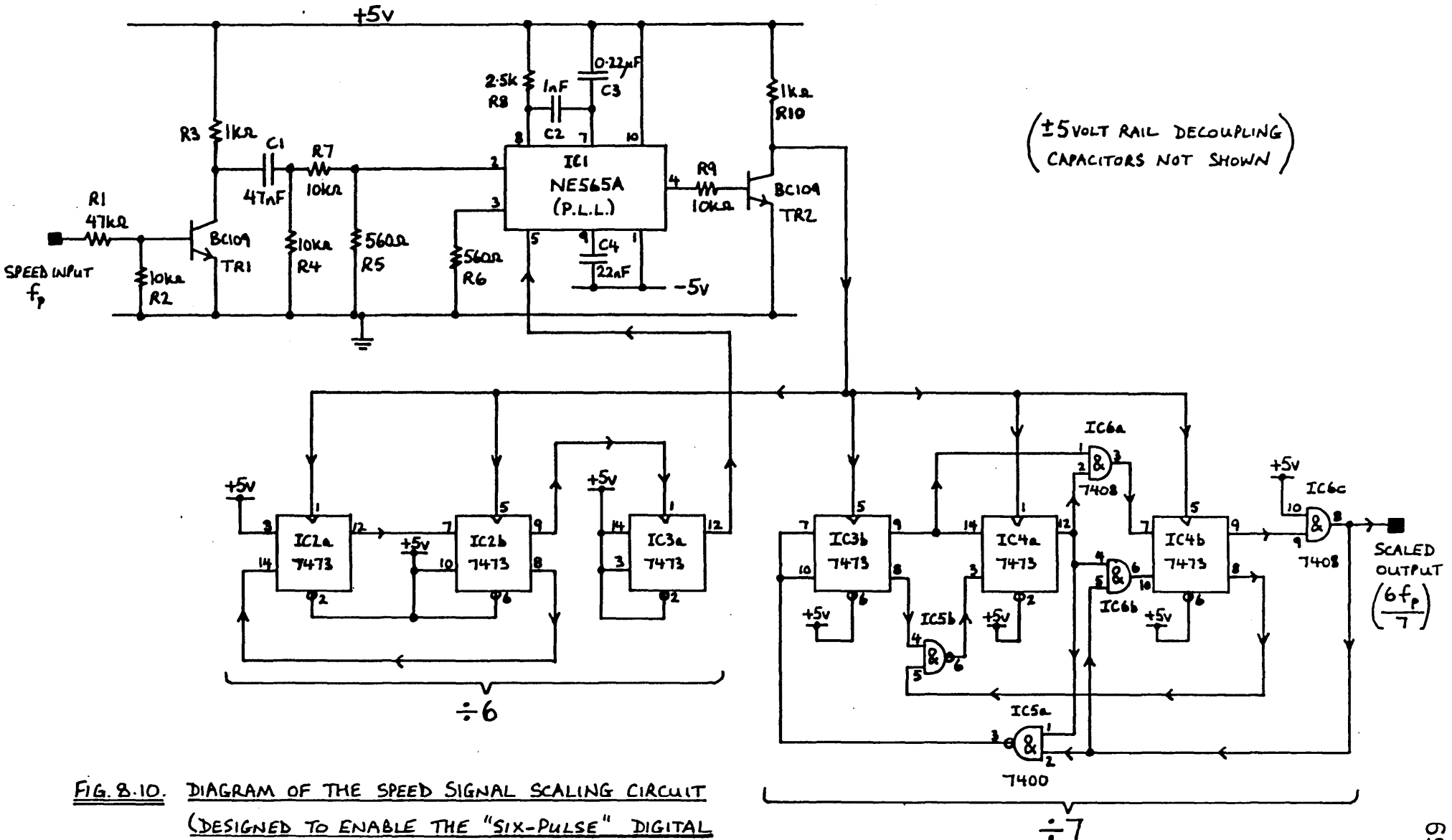


FIG. 8.10. DIAGRAM OF THE SPEED SIGNAL SCALING CIRCUIT (DESIGNED TO ENABLE THE "SIX-PULSE" DIGITAL TACHOMETER TO RUN FROM A "SEVEN-PULSE-PER-REVOLUTION" SHAFT TRANSDUCER).

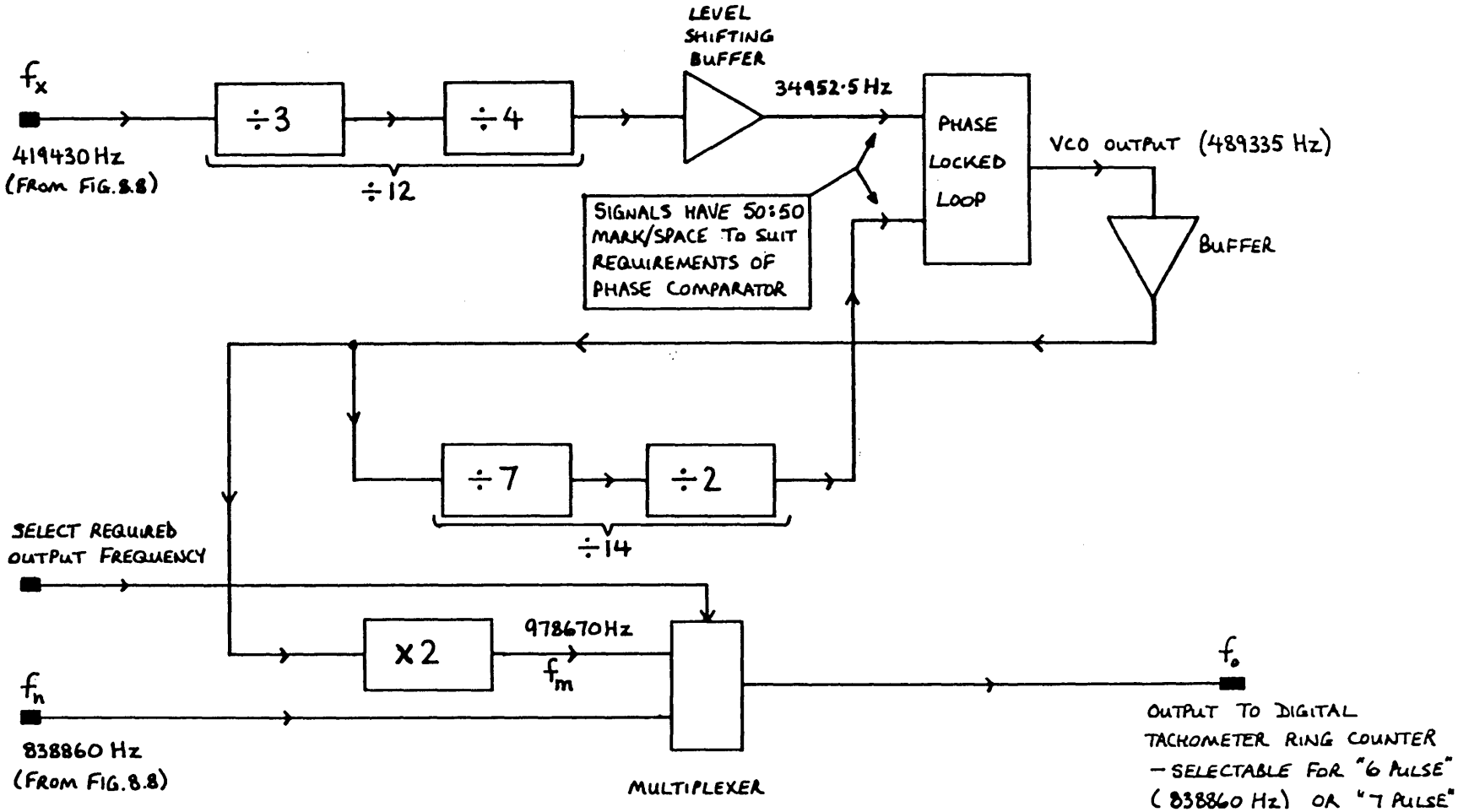


FIG. 8.11. BLOCK DIAGRAM OF A CLOCK FREQUENCY SCALING CIRCUIT TO ENABLE CORRECT OPERATION OF THE DIGITAL TACHOMETER FROM SPEED SIGNALS OF EITHER SIX OR SEVEN PULSES PER REVOLUTION.

multiplied by 2 to give a frequency f_m of 978670 Hz. This frequency causes the tachometer board to be correctly calibrated for "7 pulse" operation. By using a multiplexer to select 978670 Hz or 838860 Hz the tachometer is calibrated for "7 pulse" or "6 pulse" speed inputs respectively.

The circuit was again implemented with 7473 TTL JK flip flops and a NE565A PLL as shown in fig. 8.12. The final multiplication by 2 to get 978670 Hz was achieved using a pair of monostables (IC7a and b) which trigger on alternate edges of the input waveform. Their outputs are summed by a nand gate IC9c to give a double frequency output. The main problem with the circuit was again excessive jitter on the output frequency, and this could only be minimised by carefully trimming the transistor buffers so that the input waveforms to the PLL and the monostables were very close to an even mark-space ratio. In addition it was found that the PLL performance is very dependent on the supply voltage and the specified performance is only barely achievable when supply voltages of ±6 volts are used. The circuit of fig. 8.12 includes an extra signal channel to enable the tachometer to work with only 1 speed pulse per revolution.

Despite a great deal of effort, the performance of the fixed frequency converter was not adequate and so it was necessary to employ option (i) as the working solution.

The "7 pulse" oscillator board circuit diagram is shown in fig. 8.13. The 6.0 MHz shaped waveform is divided by seven using a ring counter (IC2 and IC5). The resulting frequency of 857140 Hz is buffered for use by the tachometer board. Further division by seven by the tachometer board ring counter results in an f_c of 122489Hz required for correct "7 pulse" operation. A D type flip-flop (IC4a) on the oscillator board divides the 857140 Hz signal by two to give a suitable clock signal for the binary to BCD circuit, and this is further divided by the 7493A counter (IC3) to provide a 26.7 kHz clock signal for

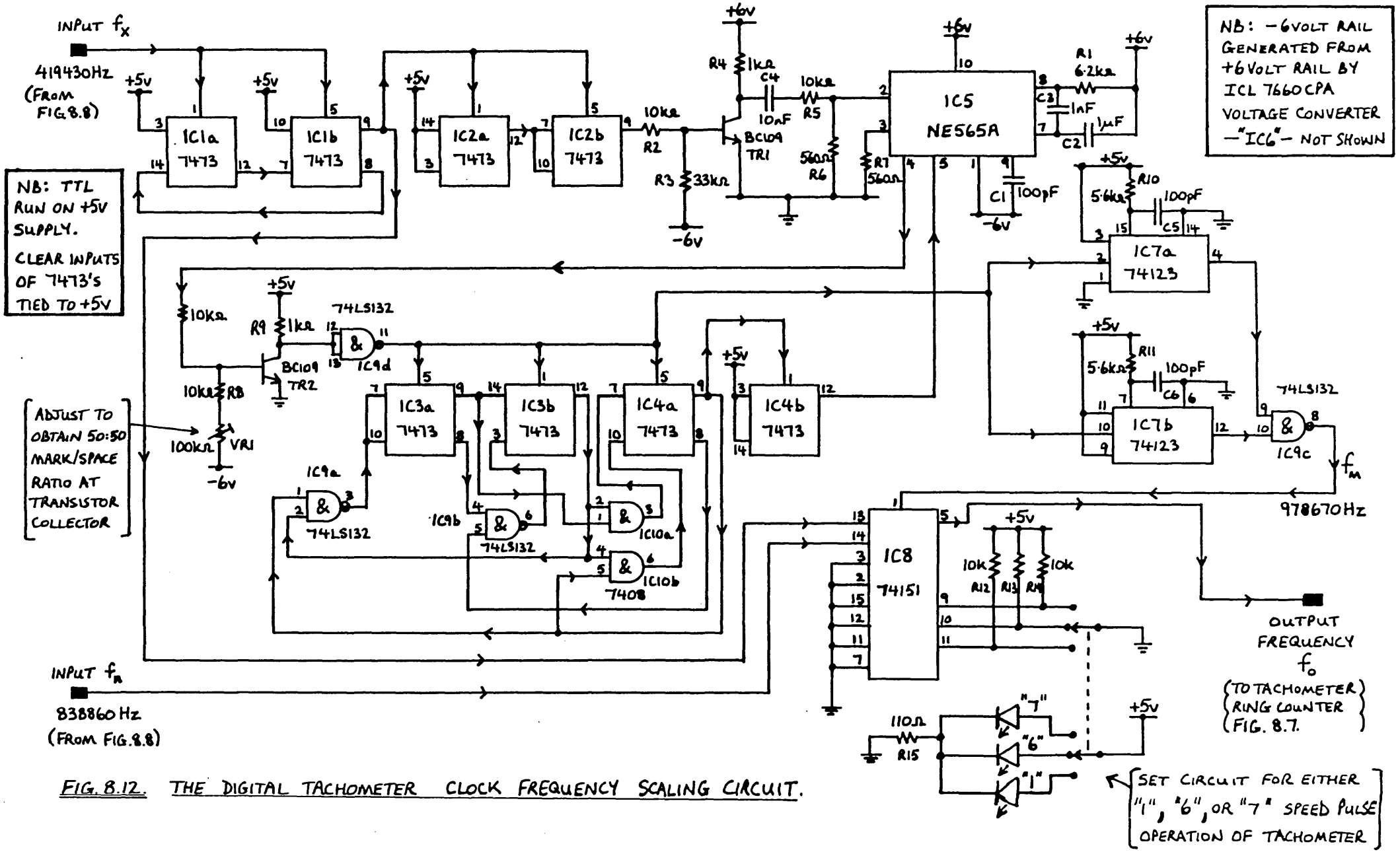


FIG. 8.12. THE DIGITAL TACHOMETER CLOCK FREQUENCY SCALING CIRCUIT.

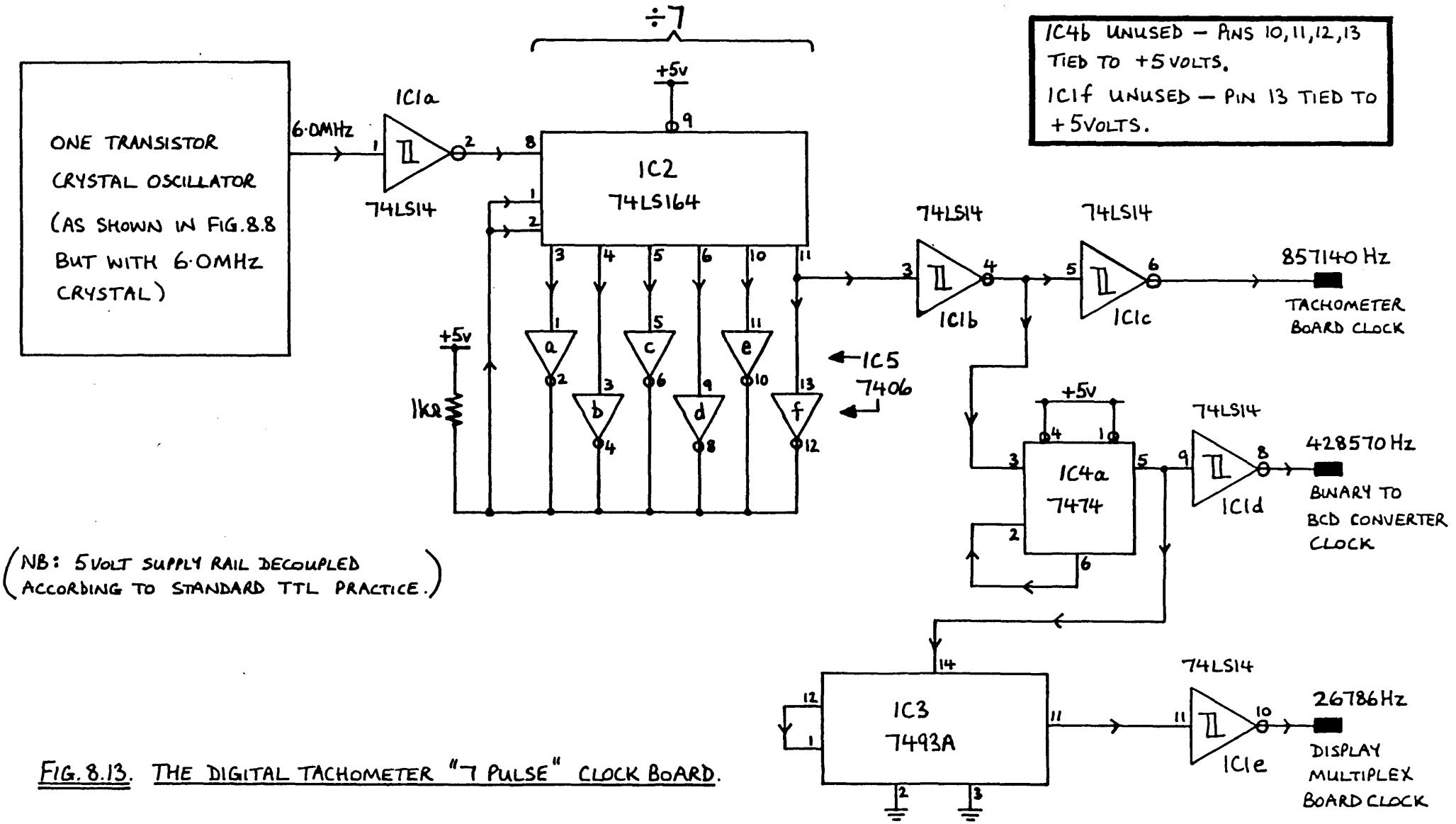


FIG. 8.13. THE DIGITAL TACHOMETER "7 PULSE" CLOCK BOARD.

the display multiplex board.

This circuit was assembled in a similar manner to the "6 pulse" oscillator board and to change the tachometer speed pulse calibration it is merely necessary to change the oscillator board in use and adjust the toggle switch on the tachometer board to suit.

8.4.4 The Binary to BCD Converter Circuit

The display of the speed information in a decimal form on a LED seven segment display requires the conversion of the binary speed signal generated by the tachometer into BCD form. Since the speed reading can only be read from the display at a relatively infrequent rate, (say once per second), it is not necessary to have a high speed converter. It is perfectly reasonable for the speed reading to be displayed for up to one second whilst the next reading is being converted ready for display.

Two methods were considered initially. They were:

(i) the use of a standard TTL binary to BCD converter circuit cascaded up to the required number of bits. The 74185A integrated circuit can convert a six bit binary word into the equivalent six bit BCD word. Fig. 8.14 shows how these circuits can be interconnected to achieve a 16 bit binary to five decade BCD conversion. The result is almost instantaneous; the only delay being propagation times through the cascaded circuits.

(ii) The use of the standard count up/count down method in which the binary number is loaded into a binary down counter and a BCD up counter is cleared. The binary down counter is then clocked down to zero whilst the BCD counter is clocked up. When zero is reached, the BCD counter contains the correctly converted data which can then be latched into the display

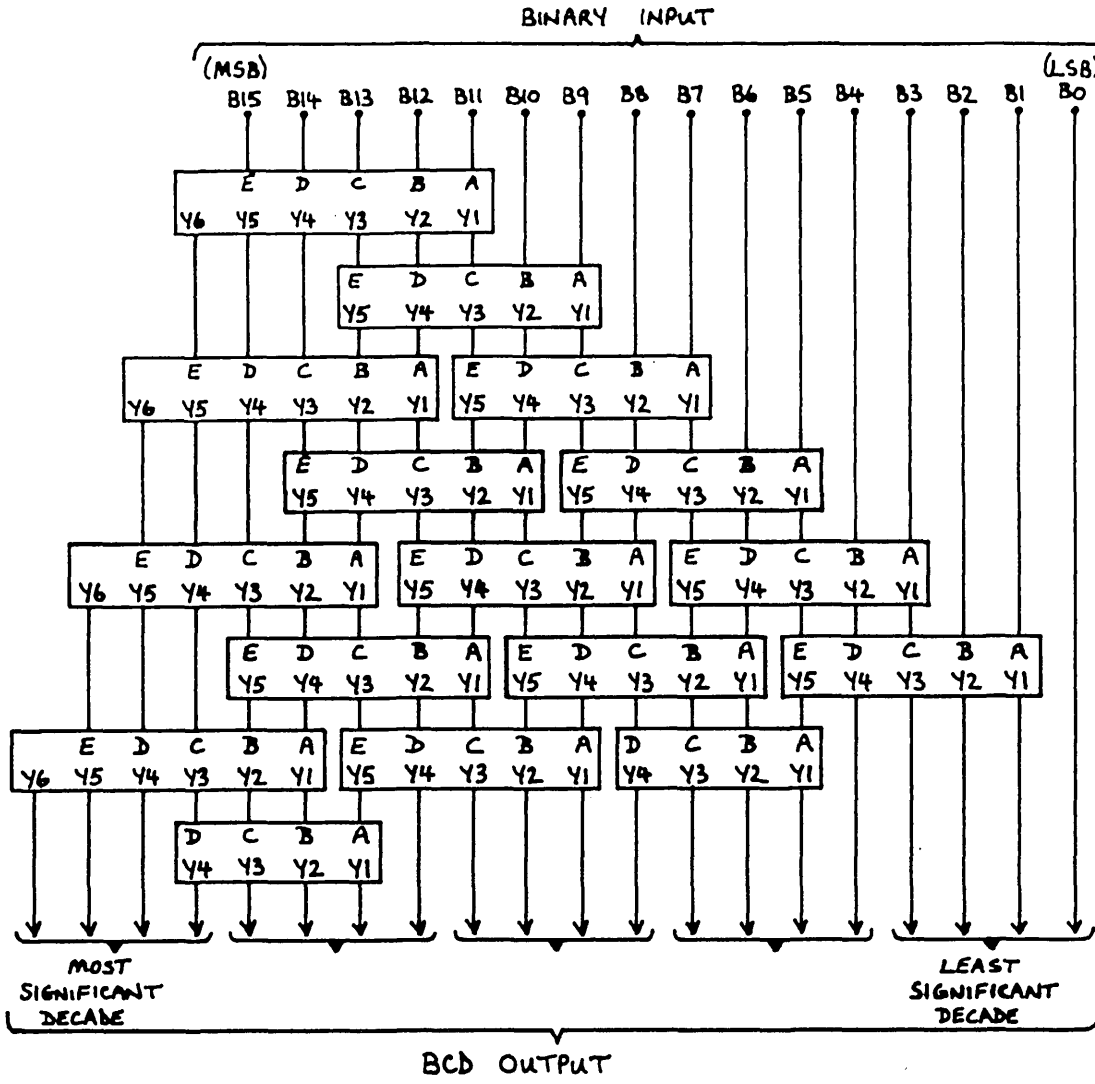


FIG. 8.14. 16 BIT BINARY-TO-BCD CONVERTER IMPLEMENTED WITH 74185A INTEGRATED CIRCUITS : EACH RECTANGLE REPRESENTS A 74185A; ALL UNUSED E INPUTS ARE GROUNDED.

unit. The conversion time depends on the size of the binary number to be converted and the clock frequency. In a given system the conversion time is proportional to the magnitude of the binary number, and so the rate at which a display can be updated varies. If a fixed update period is used, it must be long enough for the largest binary number to be converted by the system.

On the grounds of cost it was decided not to use option (i). Option (ii) was therefore developed and the circuit diagram is shown in fig. 8.15. The LED's driven by inverters IC6, IC7 and IC8 display the speed directly in binary form. When a new binary speed value is available from the tachometer, the latch input to the converter board is activated. This triggers the timing monostable IC3a.

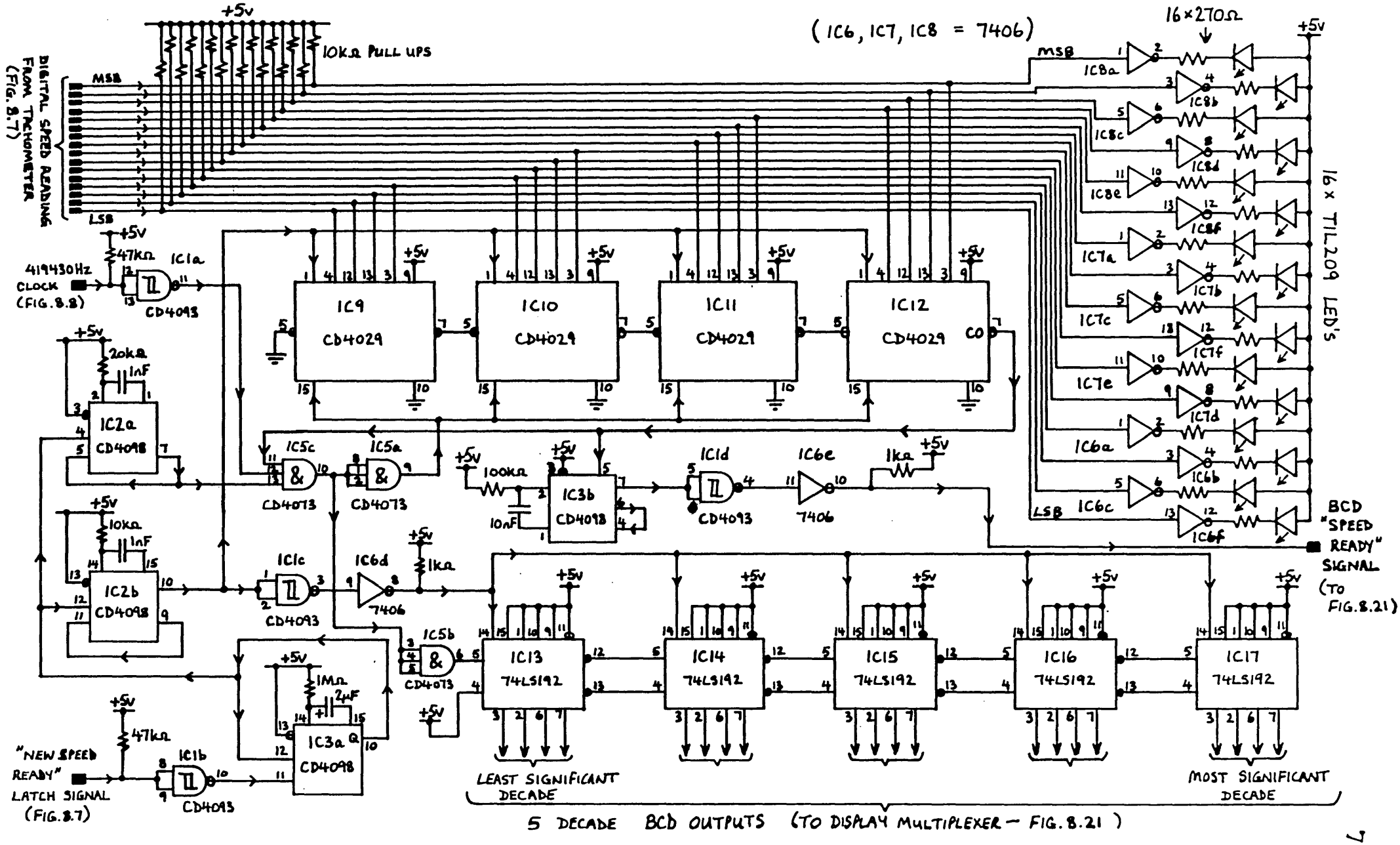


FIG. 8.15 CIRCUIT DIAGRAM OF THE COUNTER-BASED 16 BIT BINARY-TO-BCD CONVERTER.

This monostable has a 0.3 second period and is non-retriggerable. Its function is to lock out any further latch signals until the converter has dealt with the current speed value. The Q output of IC3a triggers monostable IC2b. The 10μs output pulse from IC2b is used to parallel load the binary speed number into the binary down counter (IC9, IC10, IC11, IC12) and also clear the decimal up counter (IC13, IC14, IC15, IC16, IC17). CMOS CD4029 counters are used for the decimal up counter. When the down counter is preset, the carry out (CO) output goes to logic 1 and this would enable the and gate (IC5c) and allow the 419430 Hz clock through to the counters. However, to stop the conversion beginning until the counters have been correctly initialised, a further monostable (IC2a) is triggered at the same time as IC2b and this disables the and gate (IC5c) for 20μs. The conversion is then initialised and the down counter is clocked down towards zero, with the up counter receiving the same number of clock pulses. When the down counter reaches zero the CO output goes to logic 0 and inhibits the clock. CO also triggers a monostable (IC3b) which produces a BCD latch signal to indicate to the display unit that a new BCD number is ready.

The circuit was built on a DIL circuit board. When tested in conjunction with the tachometer and the display unit it was found to malfunction.

It was found that the 74LS192 decade up counters were counting incorrectly. They could only be made to count correctly by placing 2200pF capacitors on the Q_a outputs (pin3) of all the counters (IC13 through IC17). In addition the down clock inputs on each counter were disconnected from the preceding counters borrow output and connected to logic 1 permanently. The interwire capacitance of the BCD signal outputs was also found to load the counters in such a way that the count could be corrupted. The layout of wires was therefore extremely critical. Finally, a strange flicker in the displayed speed occurred whenever the speed to

be converted exceeded 4095. This was eventually traced to a glitch on the CO output of the binary down counter IC12. The glitch is only present whenever IC12 is active, i.e. for numbers of 4096 or greater. There is no mention of the glitch in early data books but recent data books warn users of its presence and suggest methods of overcoming it. One method is employed successfully on the keypad data recall binary to BCD converter described in Chapter 7.

It was felt that the unreliable counting of the 74LS192's was a serious weakness in a system which potentially could achieve great accuracy. Rather than rebuild the circuit in a different layout, it was felt more instructive to consider a third possible option. This conversion method is popularly known as the "Add 3" method. To explain this method it is instructive to study some simple additions to reveal the relationships between binary, decimal and BCD systems.

Example 1. The sum of BCD numbers 0011 and 0101.

Solution:	0011	3
	+ 0101	+ 5
	1000	8

The addition of 3 + 5 in binary notation yields a valid sum in either binary or BCD form, since the BCD form consists of a group of four binary digits.

Example 2. The sum of BCD numbers 0110 and 0110.

Solution:	0110	6
	+ 0110	+ 6
	1100	12

The sum is valid in binary form, but it is invalid in BCD form because, just as in the decimal system, 9 is the largest digit representable. A carry is required but not generated.

Example 3. The sum of BCD numbers 1001 and 1000.

Solution:	1001	9
	+ 1000	+ 8
	10001	17

Just as in example 2, the sum is valid in binary form. In this case, a carry is generated for the BCD form, but the least significant digit (0001) is incorrect.

The problem in examples 2 and 3 is that the decimal carry occurs at 10 and the BCD carry occurs at 16 - a difference of 6. If the binary equivalent of 6 is added to the invalid sum and any overflow beyond four binary digits is added to the least significant digit of the next BCD group to the left, the sums will be valid BCD representations. In example 2, using the correction procedure, the sum of 0110 + 0110 gives 10010 which is the BCD for 12_{10}. Similarly with example 3, the incorrect sum is corrected to give the BCD equivalent of 17_{10} by the addition of 6. The examples demonstrate that the addition of 6 is necessary to maintain the decimal structure when operating in BCD notation.

Now any position b_x in a binary number has a decimal equivalent 2^x. For example, $b_4 = 2^4 = 16_{10}$. The decimal equivalent of a binary number may therefore be obtained by multiplying x times the radix (2) of the binary system itself.

i.e. $2^4 = 2 \times 2 \times 2 \times 2 = 16$

The decimal equivalent of a binary number (with the decimal equivalent expressed in BCD form) is the goal of binary to BCD conversion.

Multiplication by 2 is accomplished by shifting the binary number left (toward the most significant digit) one position.

Example 4.

$$\left.\begin{array}{rl} 0011 & = 3 \\ 0110 & = 6 \end{array}\right\} \quad 3 \times 2 = 6$$

This shift produces a valid BCD character.

Example 5.

$$\left.\begin{array}{rl} 1000 & = 8 \\ 10000 & = 16 \end{array}\right\} \quad 8 \times 2 = 16$$

This shift has not produced a valid BCD result. If the result is $\geqslant 10$, it can be corrected by adding 6_{10} to the least significant BCD group.

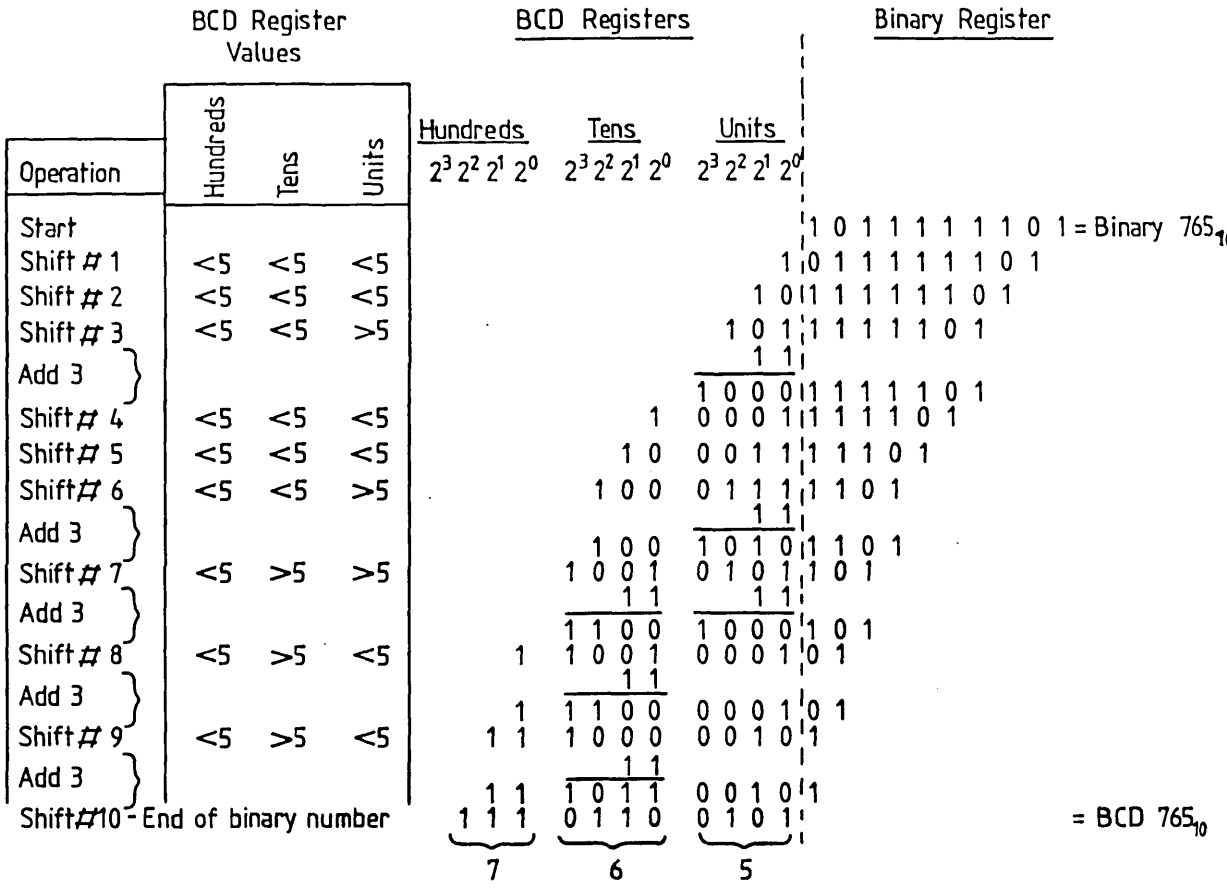
i.e.

$$\begin{array}{r} 10000 \\ \quad 0110 \\ \hline 10110 \end{array} \quad = 16_{10}$$

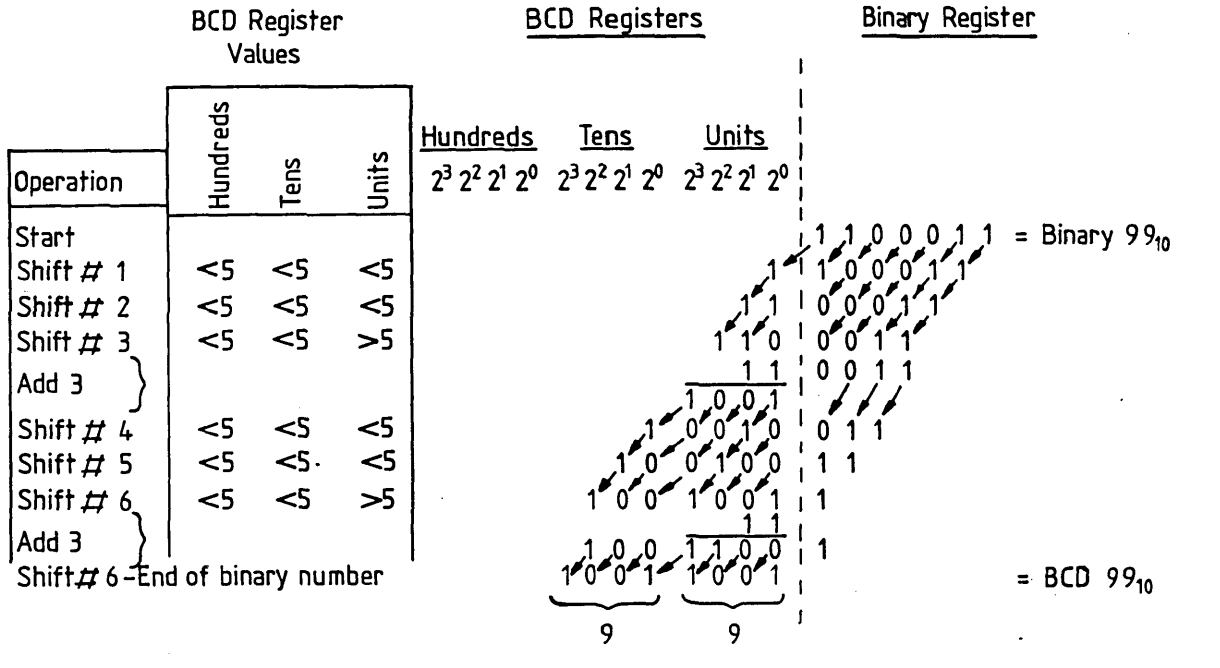
The result now has the correct BCD form.

Thus if the least significant BCD character produces a result that is $\geqslant 10_{10}$ when shifted, it requires a correction of 6_{10} to be added.

An alternative method of achieving this correction is to examine the BCD number prior to shifting. If it is $\geqslant 5$, it is necessary to add 3_{10} and then shift left. This is equivalent to checking for $\geqslant 10$ and adding 6 after the left shift. The examination, ADD 3 (if required) and the shift left operation are repeated until all binary digits are shifted. The binary number has now been multiplied by 2,



Example 1



Example 2

Fig 8.16 Two examples of binary-to-BCD conversion using the "ADD 3" method

the number of times required to convert it to its decimal equivalent, although the representation is in BCD form as required. Fig. 8.16 shows conversion of two binary numbers to BCD form.

Fig. 8.17 shows how the method can be implemented with shift registers. The binary number is inserted in parallel into a register with serial output capability. The number is then shifted into a serial/parallel register in a serial manner.

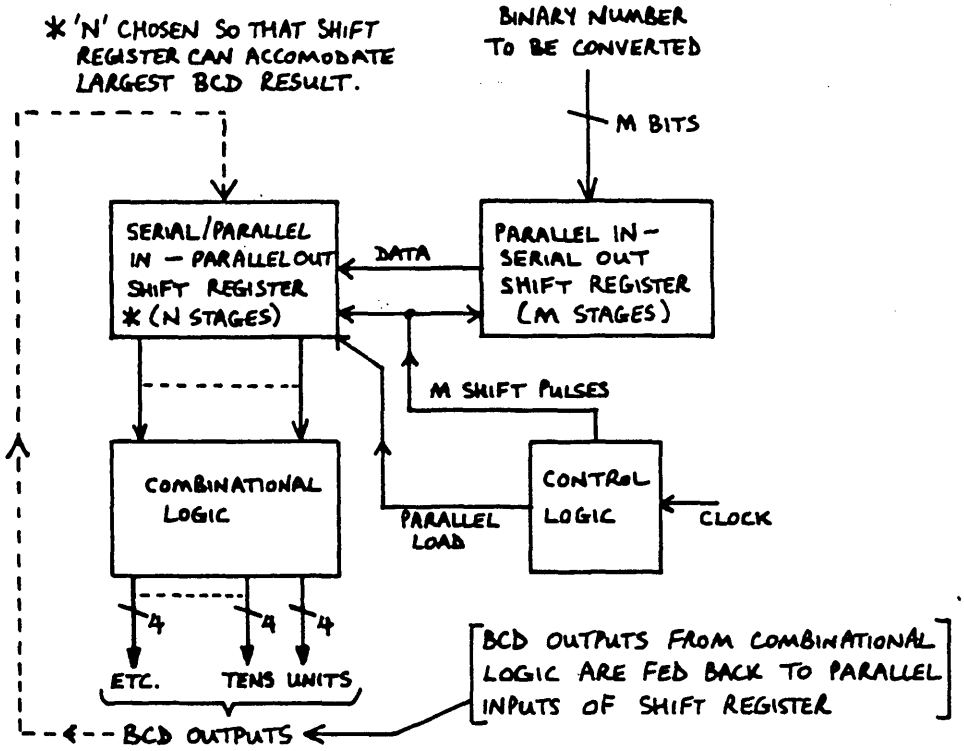


FIG 8.17 BLOCK DIAGRAM SHOWING THE IMPLEMENTATION OF THE "ADD 3" METHOD.

Examination of the three most significant bits is accomplished with a combinational logic circuit. If these bits represent a number ≥ 5, the combinational circuit adds 3 and feeds the modified number back into the serial/parallel register via the parallel inputs. Following the next shift operation, the number is again examined, 3 added if necessary, and another shift occurs. The control logic decides when to remove shift pulses, based on the total number of

bits in the binary number to be converted.

The ADD 3 method may also be implemented directly with gates and/or ROMs, leading to a faster conversion time at the expense of more circuitry. A shift register solution was more than fast enough for the tachometer system.

The circuit that was built to implement the ADD 3's method is shown in fig. 8.18a&b. The sequence of events is controlled by a multiphase clock unit which generates the signals S1 to S7 shown in the diagrams. When a new speed value is available from the tachometer, the latch input triggers the D type flip flop (IC6a) and its Q output goes to logic 1. This triggers monostable IC10a which generates control signal S1. This signal loads the new binary speed value into a parallel-in/serial-out shift register (IC1, IC2) and also clears the dual function parallel/serial in/out shift register (IC3, IC4, IC5) which eventually holds the converted BCD numbers. At the end of control signal S1, a D type flip flop (IC6b) is set to enable the conversion clock. The 419.4 kHz clock signal is fed to the 74LS393 (IC17) integrated circuit which contains two negative edge triggered four bit binary counters, whose purpose is to split the conversion time into 16 cycles of 4 sections each. The clock signal is used to pulse counter "1" of IC17. The two least significant outputs ($1Q_a$, $1Q_b$) of counter 1 go through a repeating binary sequence every four clock cycles and they drive a 74LS138 selector circuit (IC18) which activates four outputs Y0, Y1, Y2 and Y3 sequentially every four clock cycles, of which Y0 is unused. The three other signals in combination with the clock signal generate control signals S2, S3, S4, S5, S6, and S7. Output $1Q_b$ of counter 1 in IC17 also clocks counter "2" in IC17 and this counter counts 16 cycles before triggering the "end of conversion" monostable IC10b. This monostable provides a "speed ready" pulse for the display circuit and resets the D type flip flops (IC6a, IC6b).

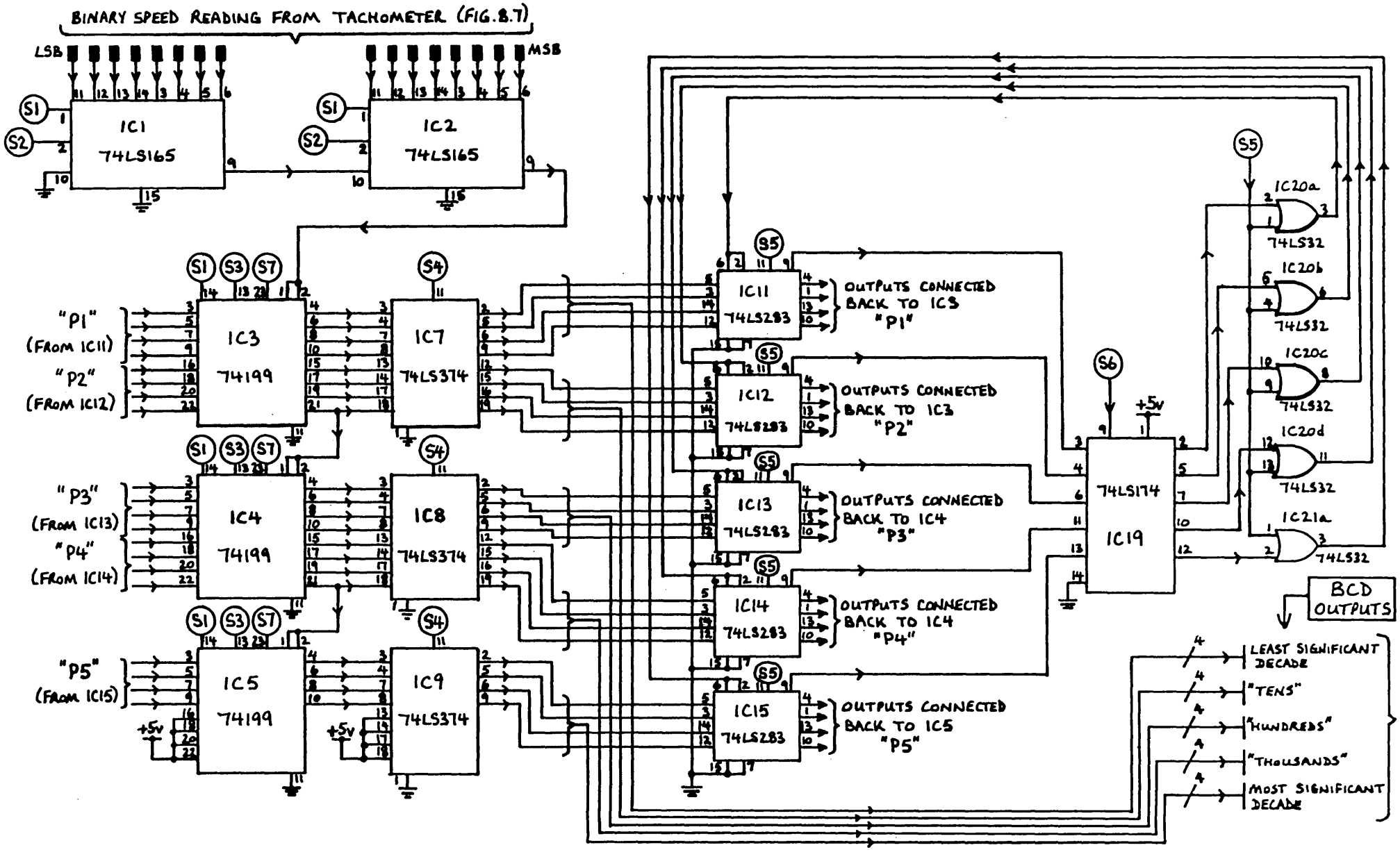
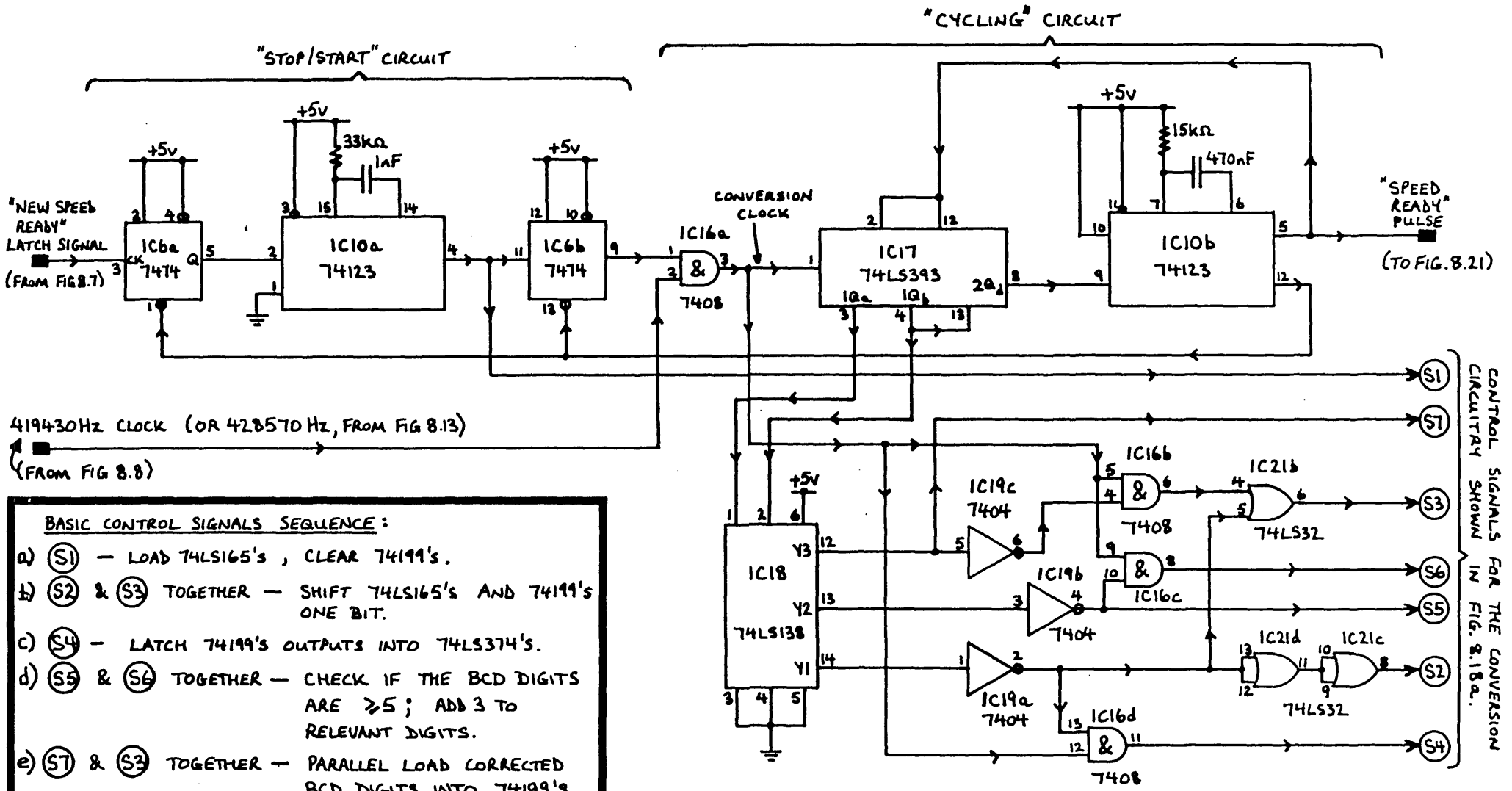


FIG.8.18a. THE CONVERSION CIRCUITRY OF THE "ADD 3" 16 BIT BINARY TO BCD CONVERTER.
CONTROL SIGNALS (S1) TO (S7) GENERATED BY CIRCUIT SHOWN IN FIG.8.18.b.



BASIC CONTROL SIGNALS SEQUENCE:

a) (S1) — LOAD 74LS165's, CLEAR 74199's.

b) (S2) & (S3) TOGETHER — SHIFT 74LS165's AND 74199's ONE BIT.

c) (S4) — LATCH 74199's OUTPUTS INTO 74LS374's.

d) (S5) & (S6) TOGETHER — CHECK IF THE BCD DIGITS ARE ≥5; ADD 3 TO RELEVANT DIGITS.

e) (S7) & (S3) TOGETHER — PARALLEL LOAD CORRECTED BCD DIGITS INTO 74199's.

STEPS b TO e REPEAT A FURTHER 15 TIMES.

NOTE: IC21b, IC21c, IC21d ARE MOUNTED ON THE CONVERSION CIRCUITRY BOARD (FIG. 8.18a)

FIG. 8.18.b. THE CONTROL CIRCUITRY OF THE "ADD 3" 16 BIT BINARY TO BCD CONVERTER. SIGNALS (S1) TO (S7) ARE USED BY THE CONVERSION CIRCUIT SHOWN IN FIG. 8.18a.

The conversion process proceeds from the end of control signal S1 as follows. On the falling edge of the 419.4 kHz clock, counter 1 in IC17 changes state and Y1 becomes active. This activates control signals S2 and S3 and clocks the most significant bit of the binary number in IC1 and IC2 into the receiving shift register (IC3, IC4, IC5). Half way through the Y1 period the 419.4 kHz clock goes to logic 1 and control signal S4 is activated. This latches the new contents of the receiving shift registers (IC3, IC4, and IC5) into the temporary memories (IC7, IC8 and IC9). At the next falling clock edge, Y2 becomes active, generating control signal S5. This causes the full adder circuits (IC11, IC12, IC13, IC14 and IC15) to add 11_{10} to each BCD digit held in the temporary memories. If any BCD digit is $\geqslant 5$, the carry output of the relevant adder is set. When the clock signal next rises to logic 1, control signal S6 stores any carries in a register (IC19): the contents of this "carry latch" are arranged to add 3 to any BCD digit that was $\geqslant 5$. The results are available at the full adder outputs and these wrap back onto the parallel inputs of the receiving shift register. The corrected BCD digits are parallel loaded into this register when Y3 generates control signal S7 followed by control signal S3. Having dealt with the first shifted bit, the system repeats the procedure 15 more times, although a correction is not required on the last shift since the least significant bit is shifted in which is the same in binary or BCD code. However, no action is required to stop any correction since the outputs of the temporary memories hold the uncorrected data after the last shift, and the output is therefore taken from there. The timing of the control signals is shown in fig. 8.19 to help to explain the circuit operation. The majority of the circuit was assembled on a DIL circuit board but insufficient room was available for the stop/start circuitry or the cycling circuit. These were mounted on the spare area of the clock board and the control signals were transmitted to the main converter via an interboard link. The conversion requires

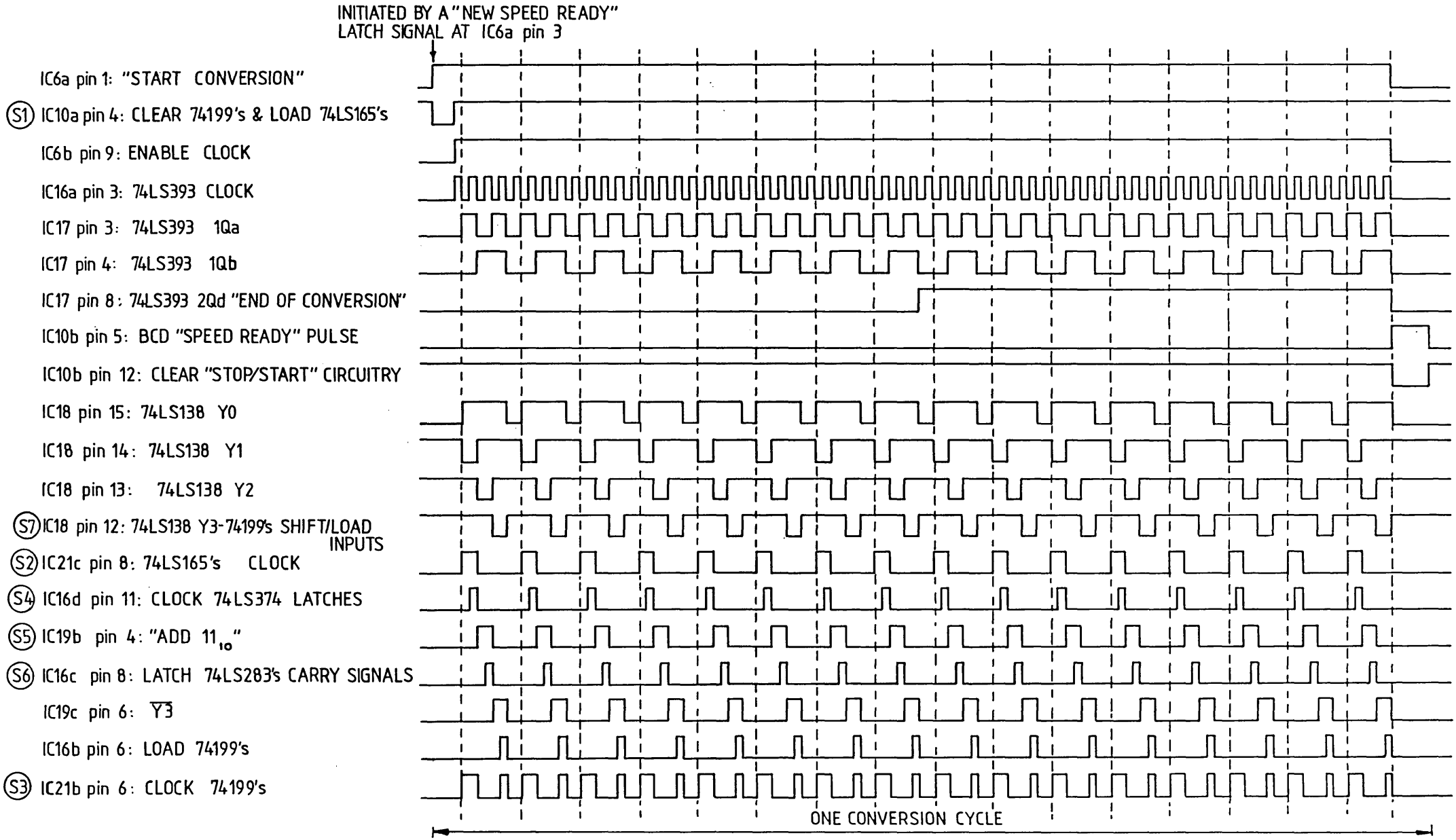


FIG 8.19 SKETCH SHOWING THE TIMING RELATIONSHIPS FOR THE "ADD 3" CIRCUIT CONTROL-SIGNALS GENERATED BY THE STOP/START AND CYCLING CIRCUITRY SHOWN IN FIG 8.18b

64 clock cycles which results in a conversion time of 0.153 ms for a clock frequency of 419.4 kHz.

The circuit has operated reliably with no problems at all.

8.4.5 The Speed Display Circuit

The display of binary coded decimal data is perhaps one of the easiest problems to solve nowadays. There are many display units available commercially which can literally be plugged on to a data bus and immediately display the data in the required format. The tachometer required a 5 decade 7 segment display. Basically three options were available to achieve this with minimum hardware and/or minimum interconnections between the display and data source. These were:

(i) the use of totally self-contained latch/decoder/display units such as the TIL 311. This unit can latch a 4 bit code into its internal memory and decode the data into the necessary drive signals for the display LED's. The unit is no larger than other standard LED 7 segment displays. Unfortunately each display costs about £10 which made the total cost of a 5 character display too expensive to consider any further.

(ii) The use of a display driver integrated circuit which can input the BCD display data and store it internally, and provide decoded 7 segment drive signals for four, six or eight 7 segment displays, depending on the circuit type. A typical 4 digit circuit is the Intersil ICM7212 which is a 40 pin integrated circuit: a typical 8 digit circuit is the Intersil ICM7218 (24 pin*). These circuits simplify the display hardware but extra hardware can be necessary to form a complete display system.

* LOW PIN NUMBER ACHIEVED BY MULTIPLEXING DISPLAY DIGITS.

(iii) The use of a purpose built multiplexed display unit which by careful circuit design minimises the number of components required. The multiplexing of the display reduces the number of interconnections between the display and drivers from 40 to 12.

It was decided to pursue option (iii) and a circuit based on the block diagram of fig. 8.20 was developed. This is based on a 20 bit parallel load shift register, into which the 5 BCD characters are loaded. The characters are then circulated around the shift register, 4 bits at a time, and each time a character is aligned with the latch it is stored. The stored character is decoded and displayed on the appropriate display. In the meantime, the shift registers shift the next 4 bit character into position and the latch then samples it and displays it. The 5 BCD characters are displayed in turn and then automatically come round again for redisplay.

The complete circuit diagram is shown in fig. 8.21. The 20 bit shift register is formed by IC1, IC2 and IC3. The data within these registers is continuously shifted and recirculated at a rate set by the clock input of 26 kHz. New data is only parallel loaded into the shift register at the end of a scanning cycle. The divide by four circuit formed by IC4a and IC6a ensures that a new character is clocked into the latch (IC8) every four shifts. The completion of a cycle of four shifts is counted by the divide by five counter (IC4b), the outputs of which are decoded into five digit select lines, only one of which is active at any time. When a "Speed Ready" pulse occurs, it is recorded by a D type flip flop (IC5a). At the completion of a scanning cycle, digit 5 select line (Y4 on IC9) goes to logic 1 and latches the "Speed Ready" signal into the next D type latch (IC5b). The Q output of this latch triggers a 0.12 μs monostable (IC7a) which enables the parallel load inputs of IC2 and IC3 causing them to enter the new data on their parallel input lines. The same pulse

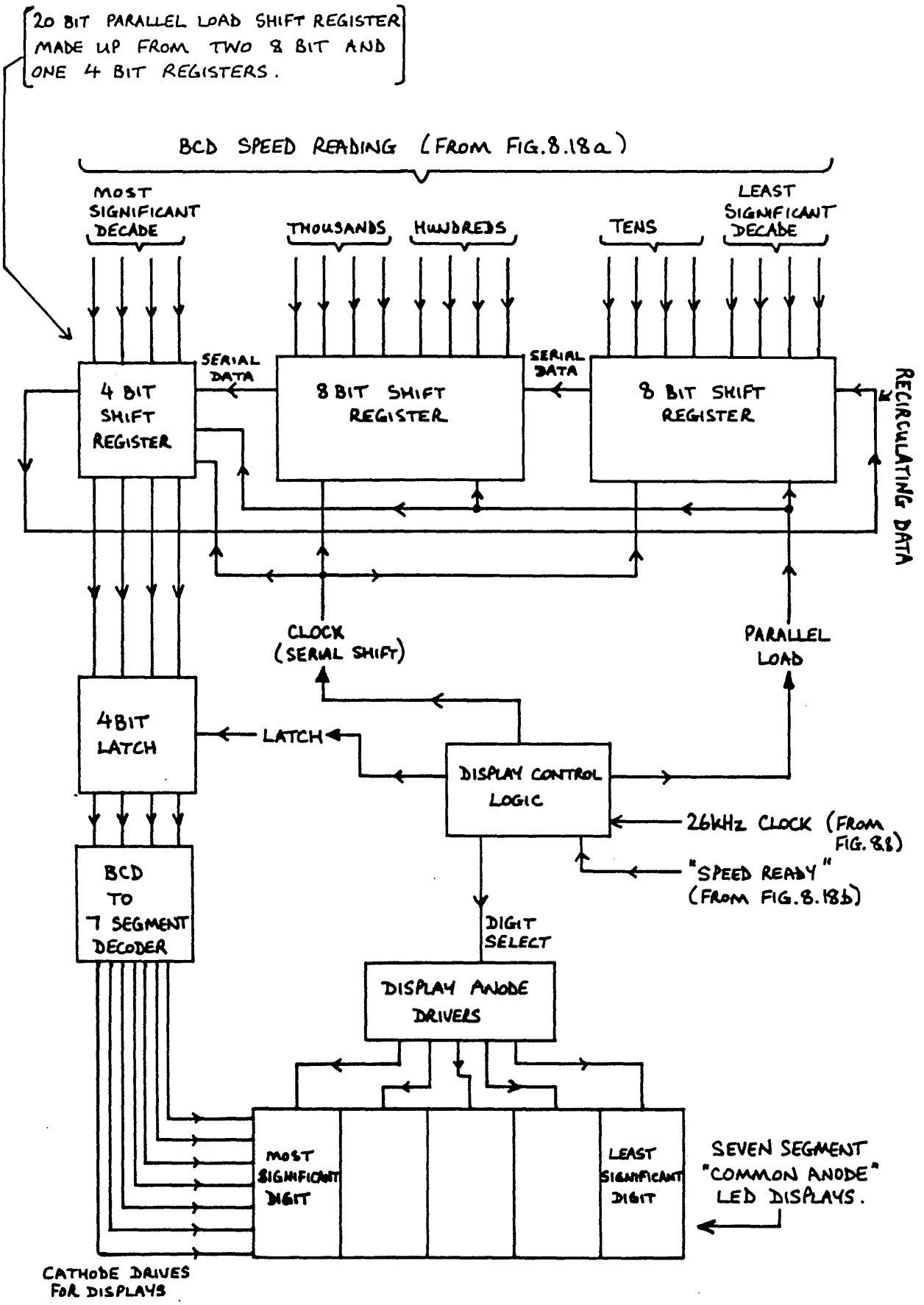


FIG.8.20. BLOCK DIAGRAM OF THE DIGITAL TACHOMETER DISPLAY CIRCUIT

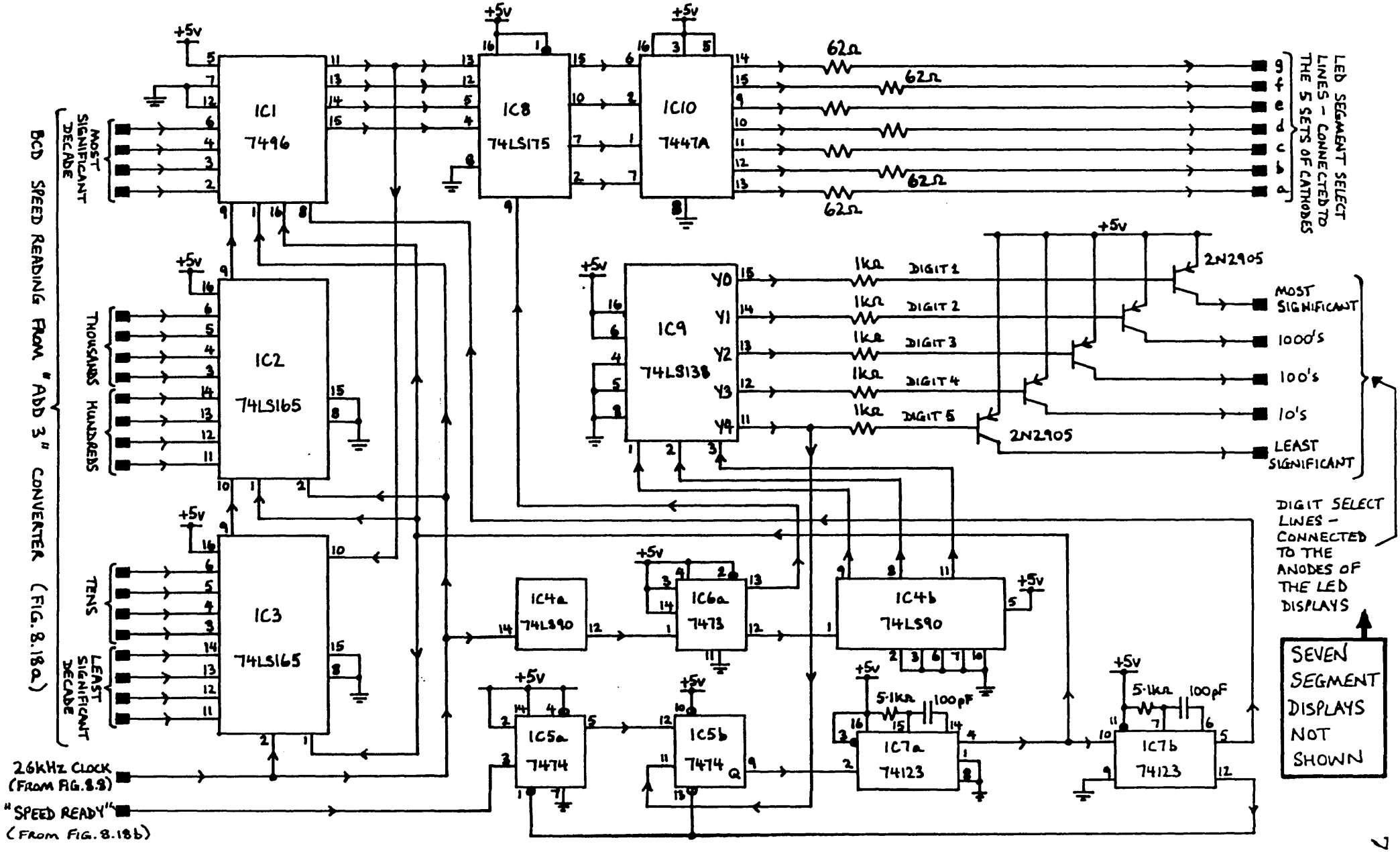


FIG. 8.21. THE DISPLAY MULTIPLEX CIRCUIT FOR THE DIGITAL TACHOMETER.

clears IC1 shift register. At the termination of this pulse, a second 0.12 μs monostable (IC7b) is triggered and this enables the parallel load input of IC1 causing it to enter the new data on its parallel input lines. Also the two D type flip flops are cleared ready for the next "Speed Ready" pulse. The whole load procedure takes a mere fraction of one clock cycle. The digit select signals are buffered by pnp transistors (2N2905) and drive the common anode connection of their respective displays. The BCD character data held in the latch (IC8) is converted into 7 segment drive signals by the 7447A (IC10) and this is connected to the cathodes of the displays via 62Ω current limit resistors.

The circuit fits easily onto a DIL circuit board and there have been no operating problems.

8.4.6 Power Supply for the Tachometer Unit

A self-contained ± 5 volt power supply was constructed for the tachometer system. The total current consumption of the tachometer is of the order of 850 mA. The power supply needs to be stable to ensure correct operation of the crystal oscillator and the TTL circuitry. The CMOS circuitry is less susceptible to power supply voltage variations. The power supply circuit is shown in fig. 8.22. The five amp regulator circuit was mounted on a 2.1° C/W heatsink, which dissipates about 10 Watts in operation.

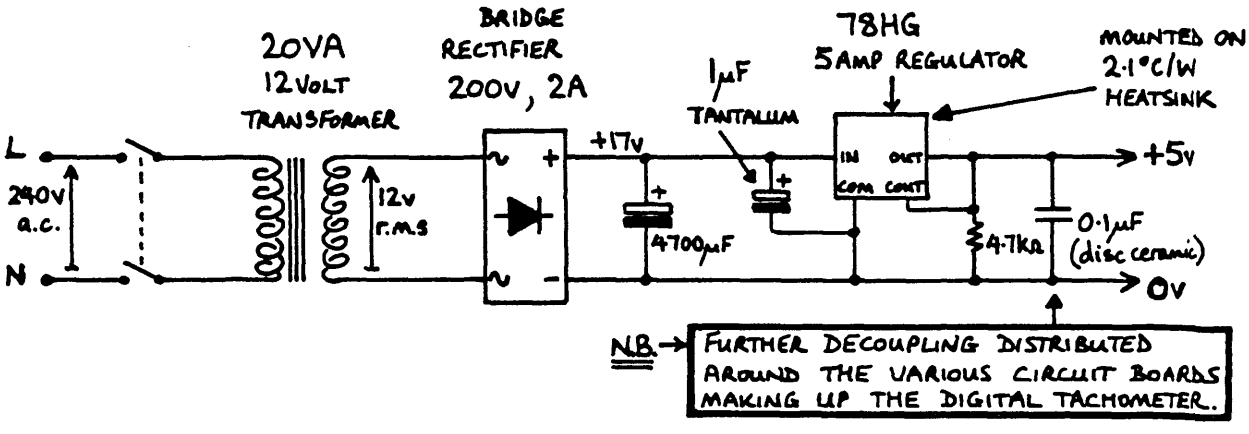


FIG. 8.22. THE +5VOLT POWER SUPPLY FOR THE DIGITAL TACHOMETER

8.4.7 Tachometer Minimum Speed Detector Circuit

A minor problem occurs with the tachometer system when the speed pulse frequency is zero or well below the cut off speed. The up/down counter is not troubled by this situation since it merely counts down to zero and jams. However, in the absence of any speed pulses, the ring counter outputs, Q3 to Q6, are inactive and so the new minimum count is not latched into the result register. Therefore if the speed of the shaft under measurement comes to a halt rapidly, an old speed value remains in the results register and cannot be replaced by the correct speed due to the absence of a latch signal. This could have serious consequences for a system which monitors the speed on the outputs of the result register, since it would appear the speed was not zero when in fact it was. In practice this problem occurred frequently with the LED display indicating a speed reading somewhere between 16 rpm and the previous running speed. The condition occurs when the motor stops in a time that is shorter than the tachometer time constant T_m.

To overcome the problem, a circuit was required that could detect if the speed pulses had fallen below the minimum speed frequency, and if so it should then pulse the speed input occasionally to ensure that the up/down counter contents are latched into the result register and into the display sections of the tachometer. The circuit of fig. 8.23 was designed and developed to do this.

The period between speed pulses at the lowest measurable speed of 16 rpm is 0.625 seconds for a speed transducer producing 6 pulses per revolution and 0.536 seconds for a 7 pulse per revolution system. It was therefore decided to detect if pulses occurred at less than 1 per second and if this is detected, "false" speed pulses should be supplied to the tachometer. The circuit functions as follows. If no speed pulses are being received, the monostables (IC1a) and (IC1b) are both reset and IC1b holds the D type

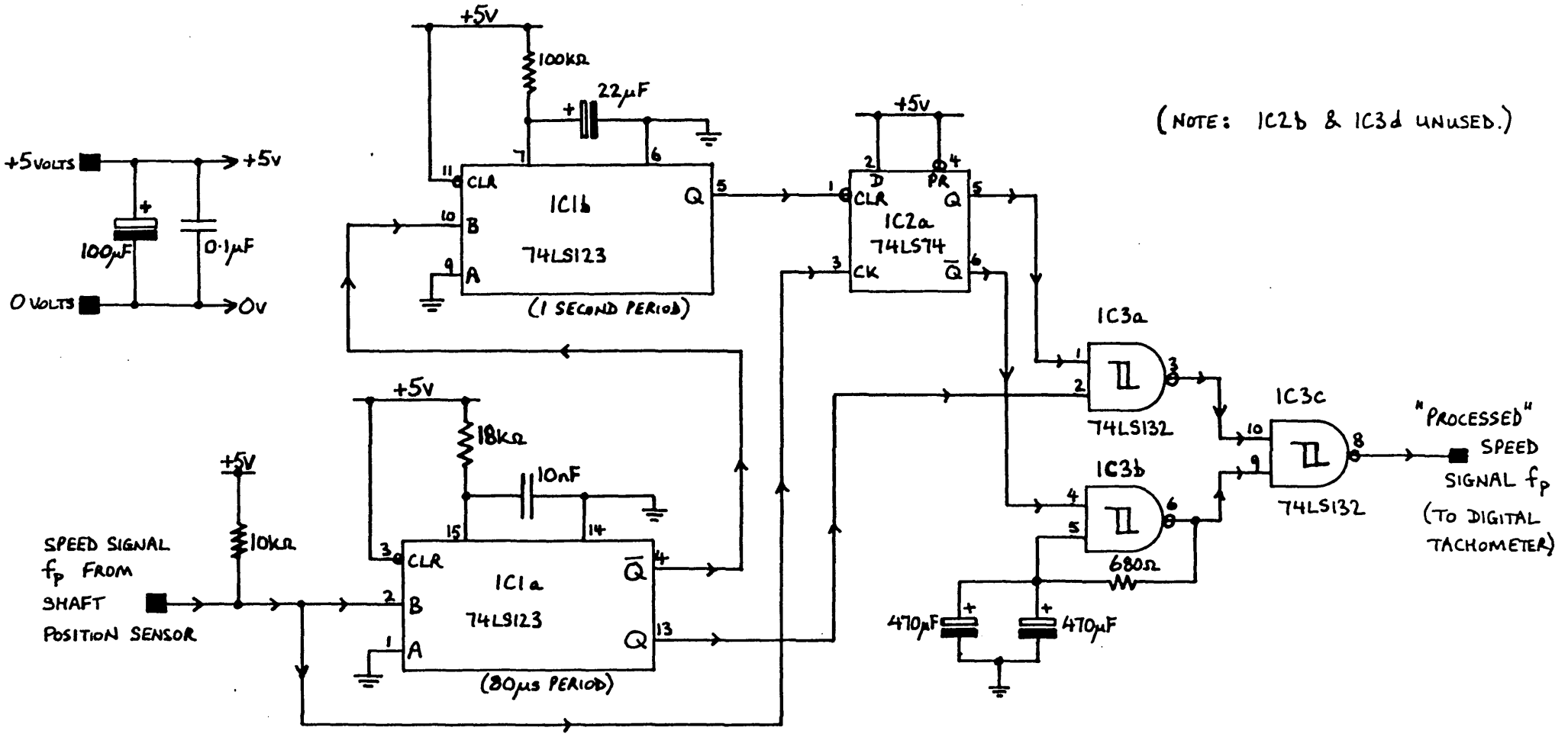


FIG. 8.23. THE DIGITAL TACHOMETER MINIMUM-SPEED (16 RPM) DETECTOR CIRCUIT.

flip flop (IC2a) in a reset state. The $\overline{Q}$ output of the flip flop is at logic 1 and enables the schmitt trigger oscillator (IC3b) which oscillates at approximately 1.1 Hz, providing a "false" speed signal to the tachometer.

If the speed pulses now increase in frequency but do not occur at an interval shorter than the period of the 1 second monostable (IC1b), nothing changes. This is because IC1b is in a reset state whenever a speed pulse occurs and so the D type flip flop is held reset even when clocked. IC1b is pulsed at the end of the IC1a timing period but it resets before the next speed pulse. However, when the speed pulse frequency rises above 1 Hz, the Q output of IC1b is still at logic 1 when the next speed pulse clocks the flip flop IC2a. Since the clear input on the flip flop is not active, the flip flop toggles and the real speed pulse is gated through to the tachometer whilst the 1.1 Hz oscillator is disabled. Meanwhile the monostable IC1b is triggered for a further 1 second period.

The circuit was constructed as an add on interface for the tachometer system and it derives its +5 Volt supply from the tachometer. In operation, the tachometer speed reading is rapidly set to the minimum speed of 16 rpm whenever the shaft comes to a halt or falls below the minimum speed.

8.4.8 Extensions in the Application of the Tachometer

The tachometer can easily be used with speed transducers that provide speed pulses which are integer multiples of 6 or 7 pulses per revolution. All that is required is a divider circuit to reduce the pulse rate to that necessary for correct tachometer operation. For example, the tachometer can be used with a disc producing 56 pulses per revolution if a divide by eight circuit is used. Further details of the divide circuits used to interface the magslip and seven phase motor to the tachometer are given in Chapter 4.

A very useful extension of the tachometer would be to use it to improve the position resolution available from a given speed/position transducer. The internal pulse train f_r in effect constitutes fictitious marks on the pulse wheel and is thus equivalent to a fictitious pulse wheel with N times as many "teeth" as the real wheel. This enables an angular measurement with extremely high resolution, which can be used in position measurement. Its use for high resolution load angle control is an interesting area to investigate in any future work.

8.4.9 Photographs of the Tachometer System

Photographs of the tachometer circuit boards and the case they were housed in are shown in figs. 8.24 and 8.25 respectively.

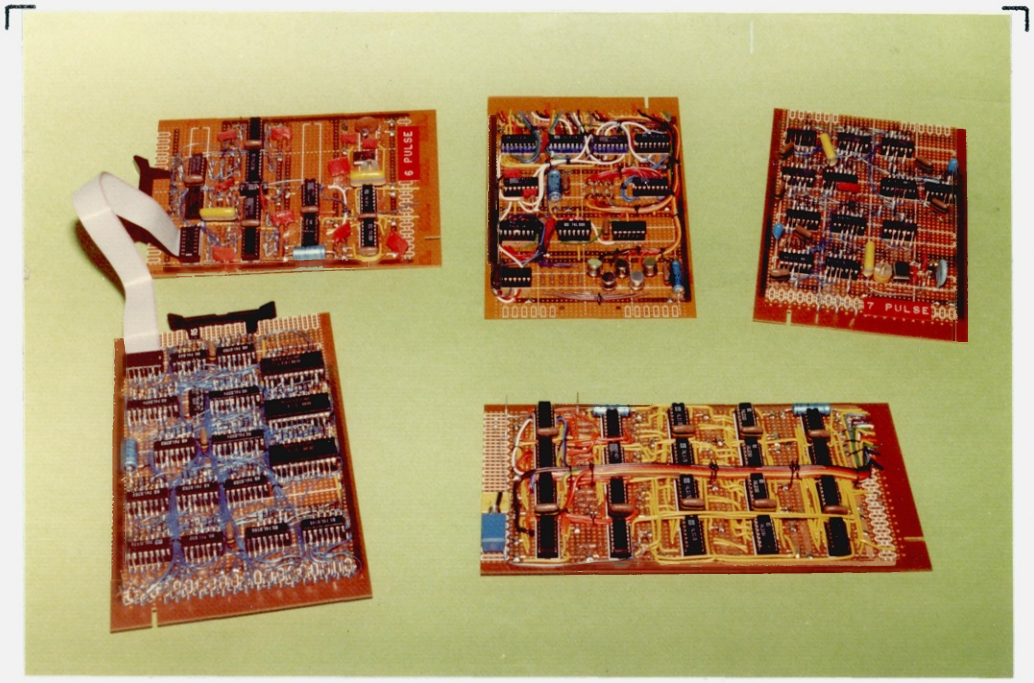


FIG.8.24. PHOTOGRAPH OF THE DIGITAL TACHOMETER CIRCUIT BOARDS. TOP LEFT: THE "SIX PULSE" CLOCK BOARD (ALSO HOUSING THE "ADD 3" CONVERTER'S CONTROL CIRCUIT) — TOP MIDDLE: THE DISPLAY MULTIPLEX CIRCUIT — TOP RIGHT: THE "SEVEN PULSE" CLOCK BOARD (WITH "ADD 3" CONTROL CIRCUIT) — BOTTOM LEFT: THE "ADD 3" BINARY TO BCD CONVERTER BOARD — BOTTOM RIGHT: THE DIGITAL TACHOMETER BOARD.

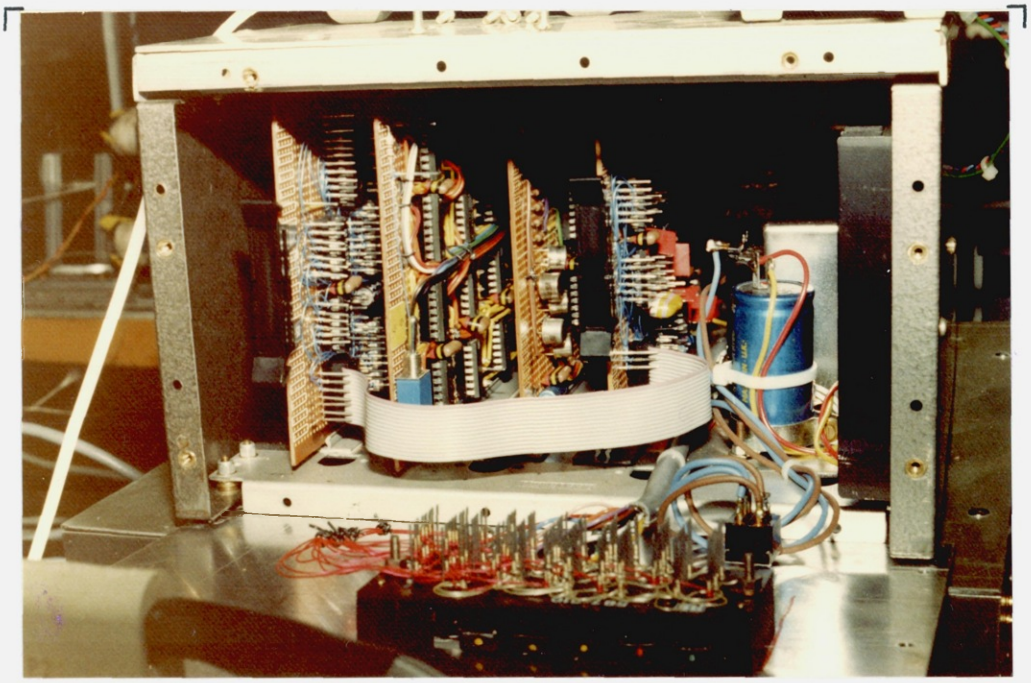


FIG. 8.25. PHOTOGRAPH OF THE DIGITAL TACHOMETER BOARDS HOUSED IN THE TACHOMETER CASE. THE BACK OF THE FIVE DIGIT LED DISPLAY CAN BE SEEN IN THE FOREGROUND.

8.4.10 Software Implementation of Török-Valis

A microprocessor can be used to perform most of the functions required by the Török-Valis tachometer algorithm. The microprocessor is ideal for calculating the new measurement number Z_{k+1} and for storing it for future use. The only part of the system where the use of a microprocessor might be difficult is the digital rate multiplier.

The BRM used in the hardware Török-Valis tachometer can be interfaced with a microprocessor to enable a software tachometer to be implemented. A possible system is shown in block diagram form in fig. 8.26. If a 16 bit microprocessor is used such as the 9900, and the BRM is clocked by the same frequency f_c as in the hardware case, the tachometer will have the same resolution and speed range. The sequence of events will be as follows.

1. On receipt of a speed pulse interrupt, the "speed" register in the microprocessor will have 16 added to

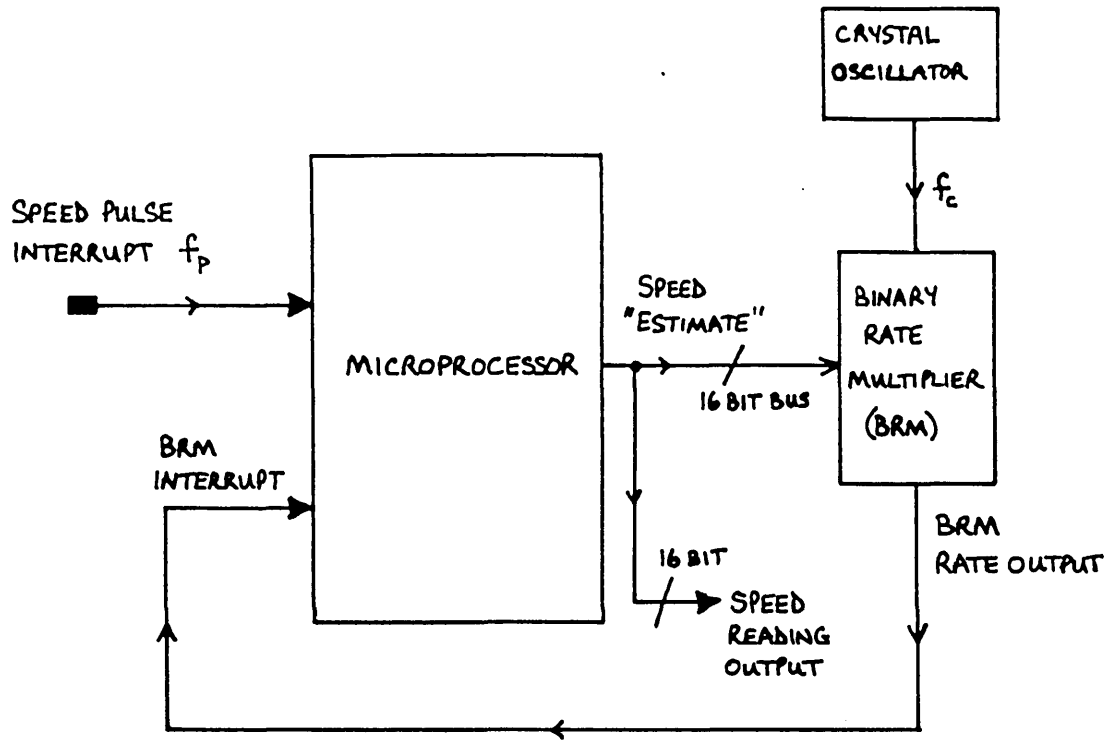


FIG. 8.26. BLOCK DIAGRAM OF A MICROPROCESSOR-BASED TÖRÖK-VALIS DIGITAL TACHOMETER.

its contents.

2. The new speed number Z_{k+1} , will be output to the BRM.

3. The BRM produces output pulses at a given rate, and each output pulse generates a BRM interrupt. On each BRM interrupt, 1 is subtracted from the "speed" register.

4. At the next speed pulse interrupt, if the system is in equilibrium, the "speed" register contents will be the same as they were prior to step 1.

The sequence continues indefinitely and the necessary software is very simple, and of course, extremely easy to change if the performance of the tachometer should require modification.

Fig. 8.27 shows the digital tachometer circuit suitably modified to allow the BRM and ring counter to be used in a microprocessor-based Török-Valis tachometer system.

By taking the $\overline{\text{BRM SELECT}}$ line to logic 0, the hardware tachometer is disabled. This could be selected by the microprocessor. The up/down counter has all its outputs forced to logic 1 and the latch signal strobes the logic 1's into the result register. The 7407 buffers are placed in a logic 1 state, and since they are open collector devices they have no effect on the 16 bit bus. The microprocessor can therefore place its "measurement number" onto the bus for the BRM to act on. The BRM output is routed out as the BRM interrupt. The tachometer port on the microprocessor interface board was in fact arranged so that it could act either as a 16 bit input or output port for the tachometer, and the circuitry is shown in Chapter 4. It was lack of time that prevented any further work on the software tachometer. It is interesting to note that the display unit will show the speed calculated by either the hardware or software tachometer, since it is connected directly to the 16 bit bus.

8.5 Tests and Conclusions

The tachometer circuit has worked successfully and reliably for a period of two years. The digital transfer of speed data from it to the 9900 microprocessor has been successfully achieved. The time constant of the circuit has been verified by means of a stop watch. An "instantaneous" top speed of 65535 rpm was input to the tachometer from a signal generator, and the tachometer took about five time constants to give a reading of 65535 after starting from 16; i.e. the time taken was just over 3 seconds. Since the binary to BCD converter converts so quickly, there are many more samples per second given to the display than are strictly required, and the small amount of flicker

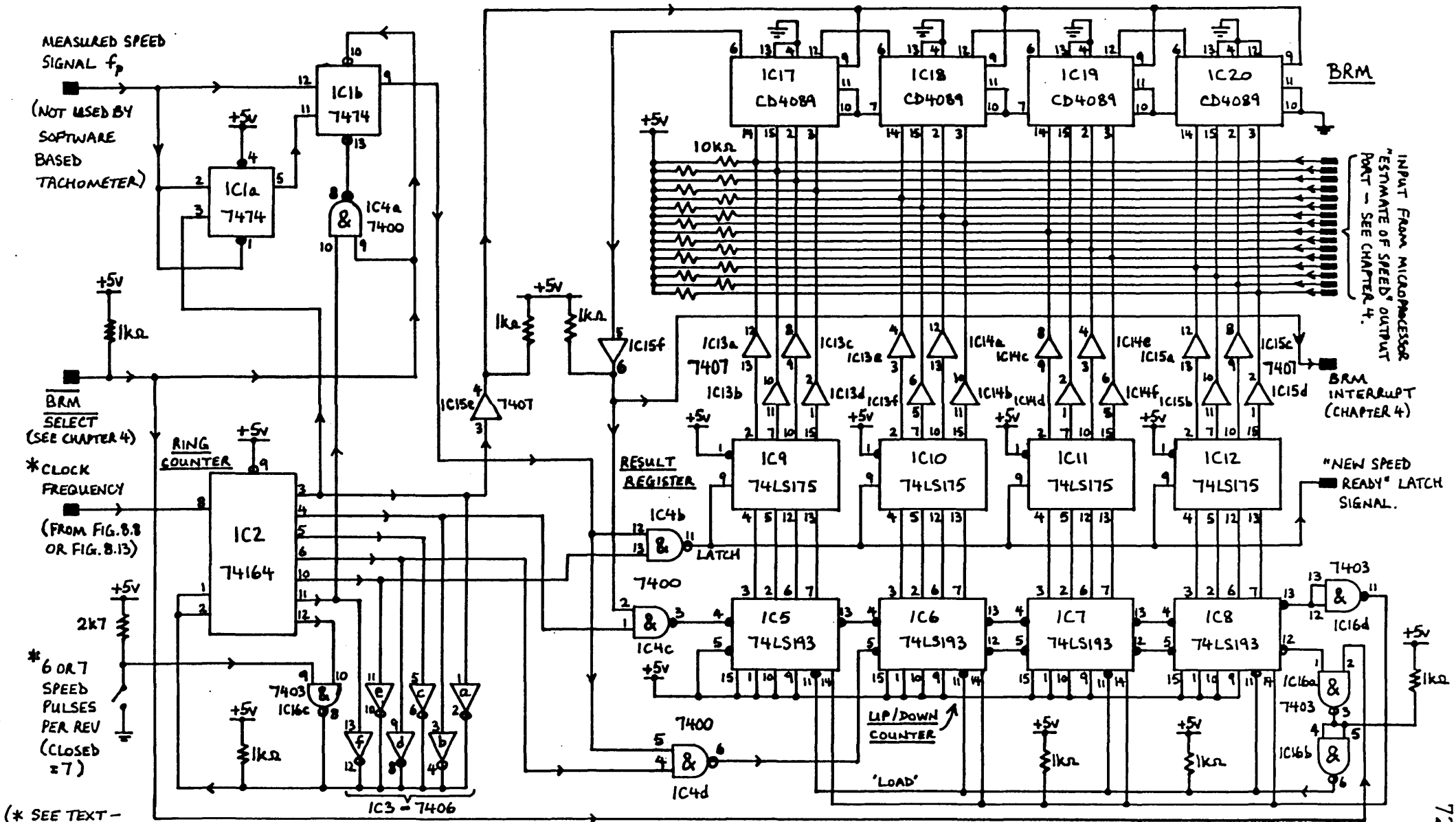


FIG. 8.27. THE 16 BIT TÖRÖK-VALIS DIGITAL TACHOMETER CIRCUIT MODIFIED FOR SOFTWARE BASED OPERATION. (SETTING BRM SELECT TO LOGIC 0 ENABLES A MICROPROCESSOR TO ACCESS THE BRM'S AND RECEIVE BRM INTERRUPTS).

on the least significant digit could easily be eliminated by reducing the display sample rate.

In all respects the tachometer has been a success. It has functioned without fault as a microprocessor peripheral, providing speed data in the range 16 rpm to 65535 rpm in 1 rpm steps, (a resolution of 0.0015 percent of full scale).

CHAPTER 9

THE PERFORMANCE TESTS CARRIED OUT ON THE 7 PHASE "SQUARE-WAVE" SYNCHRONOUS MOTOR

9.1 Introduction

The experimental tests carried out on the 7 phase square-wave motor are briefly described in this chapter, and the results are presented and discussed. The motor was driven by the MOSFET inverter described in Chapter 5 under the control of the TMS9900 microprocessor system described in Chapter 4. Steady state tests at speeds of up to 25000 rpm were possible with the system.

The experimental test rig is described in section 9.2. General comments on the motor performance are given at the beginning of section 9.3. The later sub-sections of section 9.3 cover the various motor test results in detail. Sections 9.4, 9.5, and 9.6 comment on the inverter efficiency, the accuracy of the motor loss-torque characteristics, and the phase current waveshape, respectively.

In general the results indicate that the "square-wave" motor/inverter concept is viable, and the areas in which significant motor improvements could be made are discussed where appropriate.

9.2 The Experimental Test Rig

The 7 phase motor and the eddy current dynamometer

were mounted in line on a dural baseplate, as previously described in Chapter 6. The necessary connections between the various components of the drive system were made as shown schematically in fig. 9.1. The locations in the system at which voltage, current, and power levels were monitored are indicated in fig. 9.2. The positive and negative supply rail currents to the inverter were measured by 30 amp f.s.d. moving coil ammeters, and the corresponding rail voltages were measured by a Philips digital multimeter (type PM2517X). The same Philips meter was used (set onto an a.c. voltage range), to measure the phase 1 inverter output and motor input voltages. Another Philips digital multimeter was used, set to its 10 amp a.c. range, to measure the phase 1 motor current. It should be noted that the readings obtained via the Philips digital multimeters were true r.m.s. The series resistor-inductor protection components (the precautionary current limit components described in Chapter 5), were connected between the inverter and the 7 phase motor as shown in fig. 9.2. The phase 1 inverter output and motor input powers were measured separately by Feedback electronic wattmeters (type EW604). The difference in the two power readings can be accounted for by the power dissipation in the protection components. The phase 1 voltage and current waveforms were monitored on an oscilloscope at the points indicated on fig. 9.2. Observation of the voltage and currents in phases 2 to 7 inclusive, indicated that the amplitudes measured in phase 1 were representative and so only phase 1 was monitored during the tests. The motor speed was measured by means of the digital tachometer described in Chapter 8.

The arrangement of the motor, inverter, and microprocessor system was such that the microprocessor was kept well away from the "electrically noisy" inverter and motor (the microprocessor was in fact about 4 metres from the inverter). This would probably not be possible in a commercial system, but it was desirable in this case to minimise the chances of interference. Fig. 9.3 is a photograph

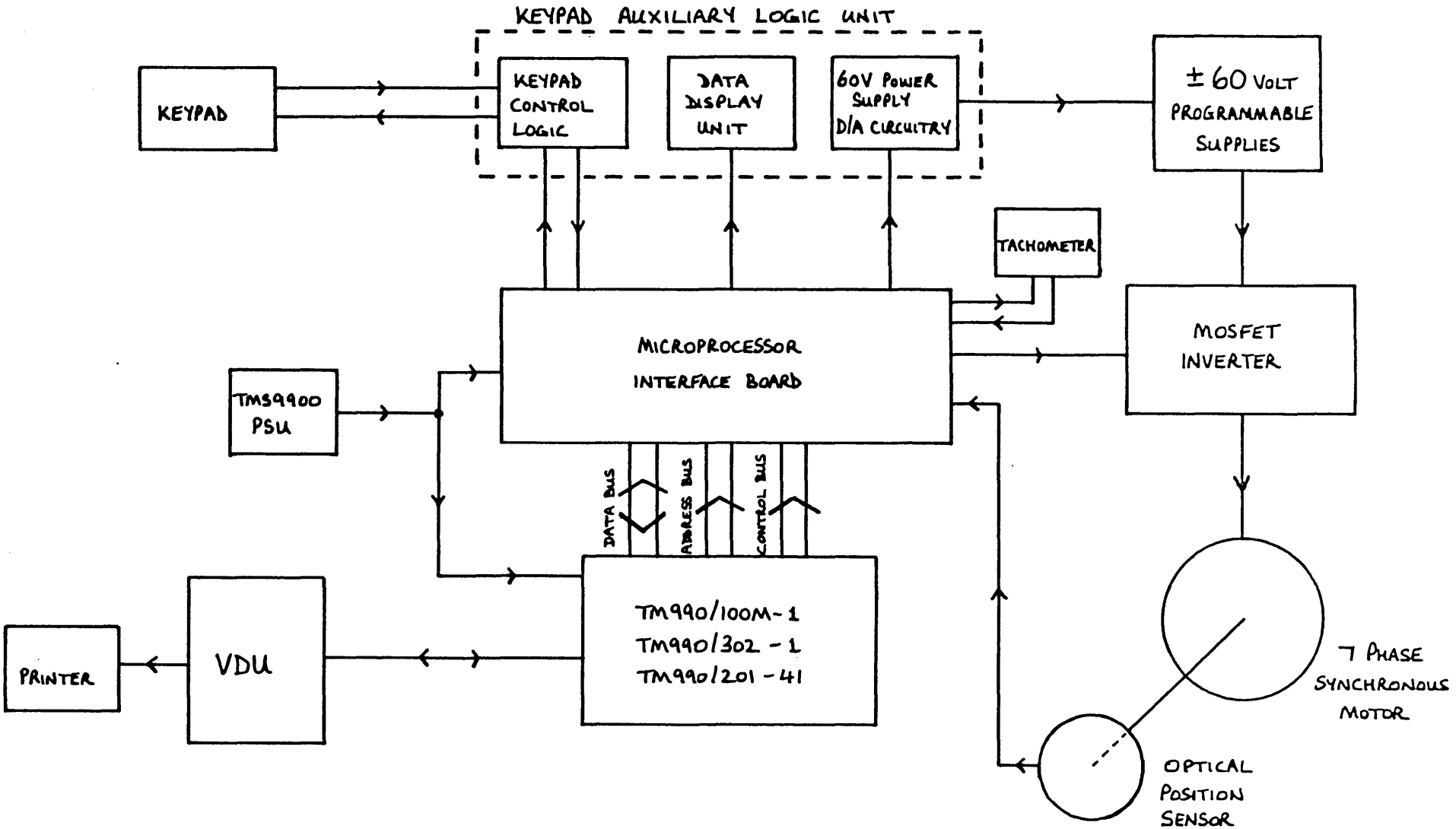
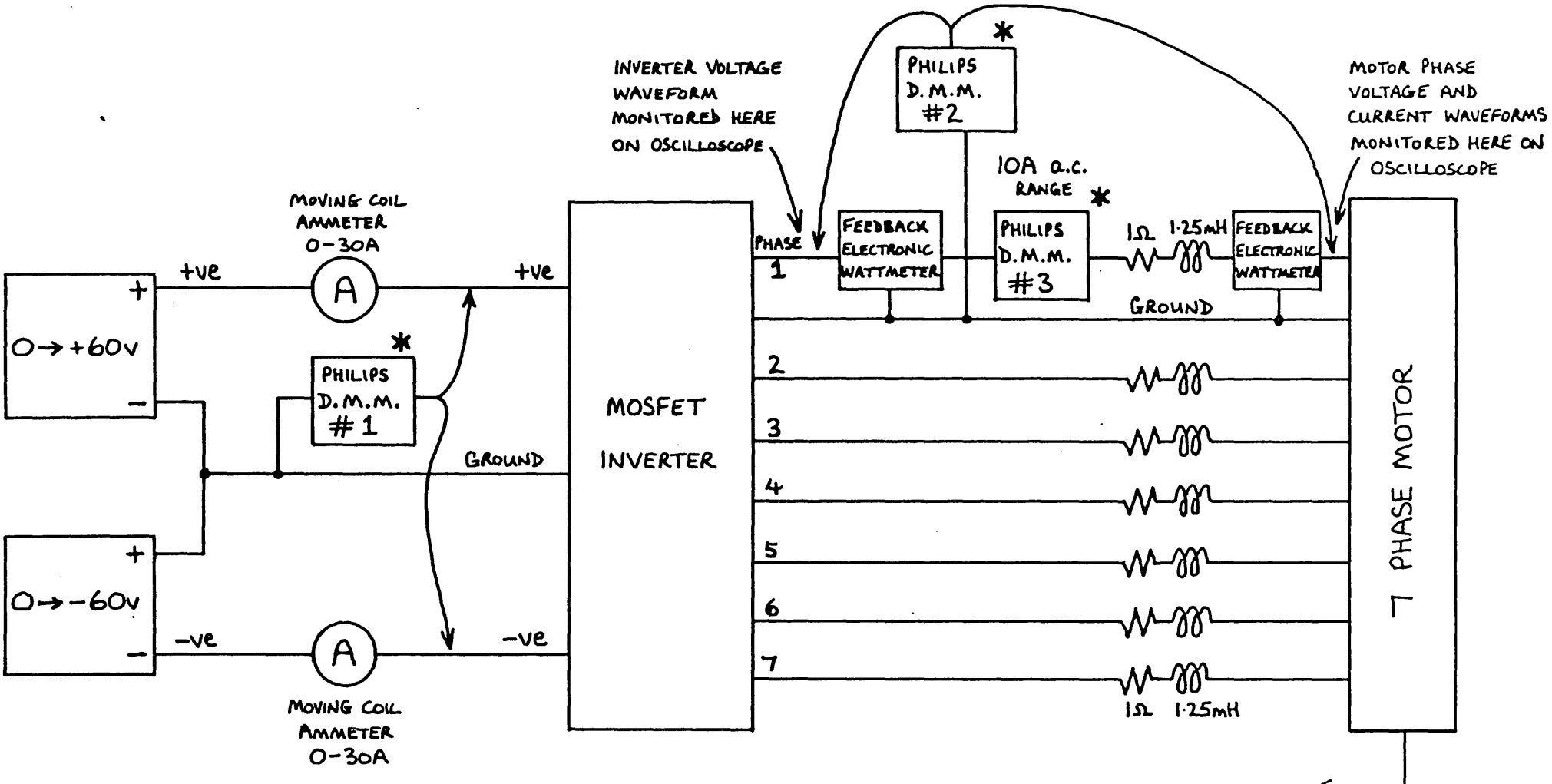


FIG. 9.1. SCHEMATIC DIAGRAM OF THE SQUARE WAVE MOTOR DRIVE SYSTEM



*NOTE – ALL PHILIPS D.M.M. (DIGITAL MULTIMETER) ARE TYPE PM2517X
#1 & #2 ARE ACTUALLY SAME METER.
#1 ON D.C. VOLTS RANGE ; #2 ON A.C. VOLTS RANGE;
#3 ON A.C. CURRENT RANGE.
(ALL A.C. RANGES ARE TRUE R.M.S.)

FIG. 9.2. THE TEST INSTRUMENTATION ON THE MOTOR DRIVE SYSTEM

FIG. 9.3. VIEW OF THE MOTOR TEST AREA

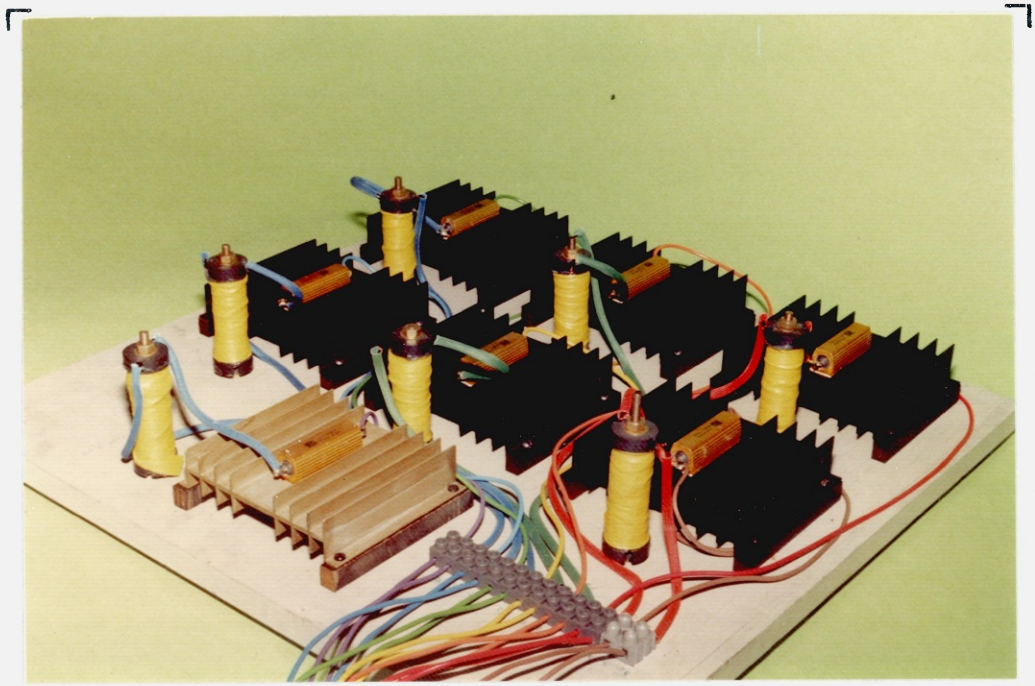


FIG. 9.4. VIEW OF THE INVERTER PROTECTION COMPONENTS

of the test rig and it shows the layout of the motor, the ±60 volt supply, the inverter, and the protection components. A close-up photograph of the protection components is shown in fig. 9.4. Fig. 9.5 is a photograph of the "microprocessor area", and it shows the microprocessor card cage, the VDU, the keypad with its auxiliary logic unit, and the printer used for program listings.

The rigorous development work carried out on the autopiloting software during the magslip tests was found to have been worthwhile, because the 7 phase motor system ran at the first attempt. No software problems were experienced during the 7 phase tests. Full details of the software are given in Chapter 4.

The 7 phase motor was run at a speed of about 6000 rpm, for a period of five minutes, prior to the start of any set of load tests. This was done to warm up the rotor bearings, in an attempt to reduce changes in bearing friction over the period of a test.

The measurement of motor torque was mainly achieved by use of the eddy current brake dynamometer. A photograph of the dynamometer is shown in fig. 9.6. The UF1 load cell was energised from a 5.0 volt stabilised d.c. supply, and the output voltage from the cell was measured by a true r.m.s. Data Precision 2480R digital multimeter. A true r.m.s. multimeter was used so that any noise on the d.c. output of the load cell was filtered out. The load cell sensitivity varied slightly from day to day, and so it was calibrated prior to each set of tests. It had a value of about 43.40g/mV. With the load cell connected to the torque arm at the 20cm radius point, this gave a torque sensitivity of 85.15mN/mV. The friction in the dynamometer bearings was reduced as much as possible by degreasing them and lubricating them with light oil. In addition, the alignment of the two bearings was carefully adjusted to minimise friction. The friction could have

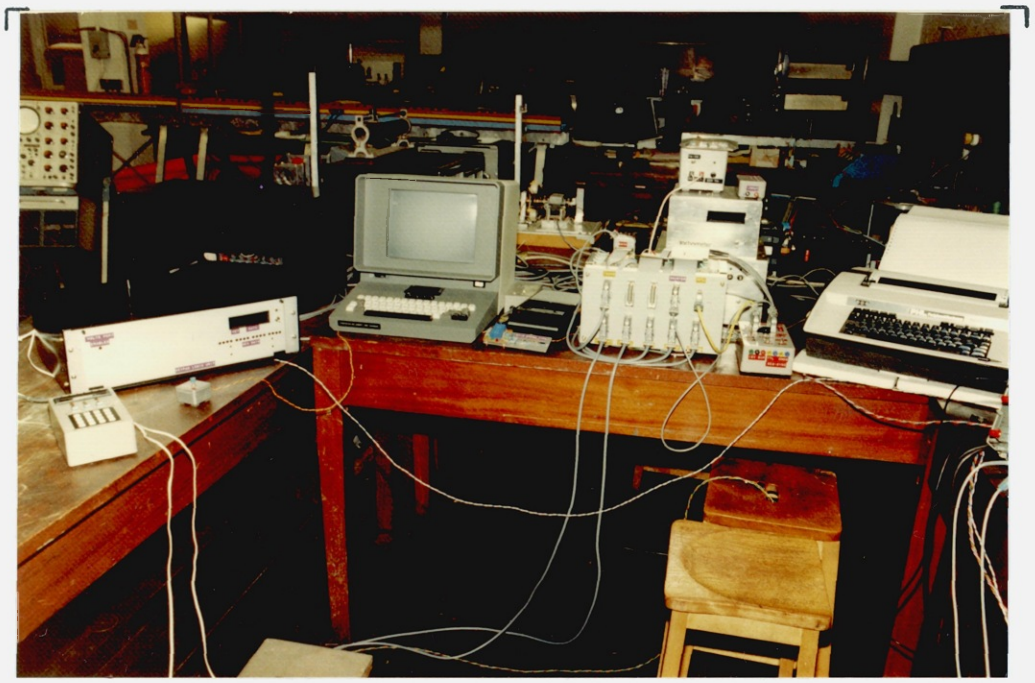


FIG.9.5. VIEW OF THE "MICROPROCESSOR AREA"

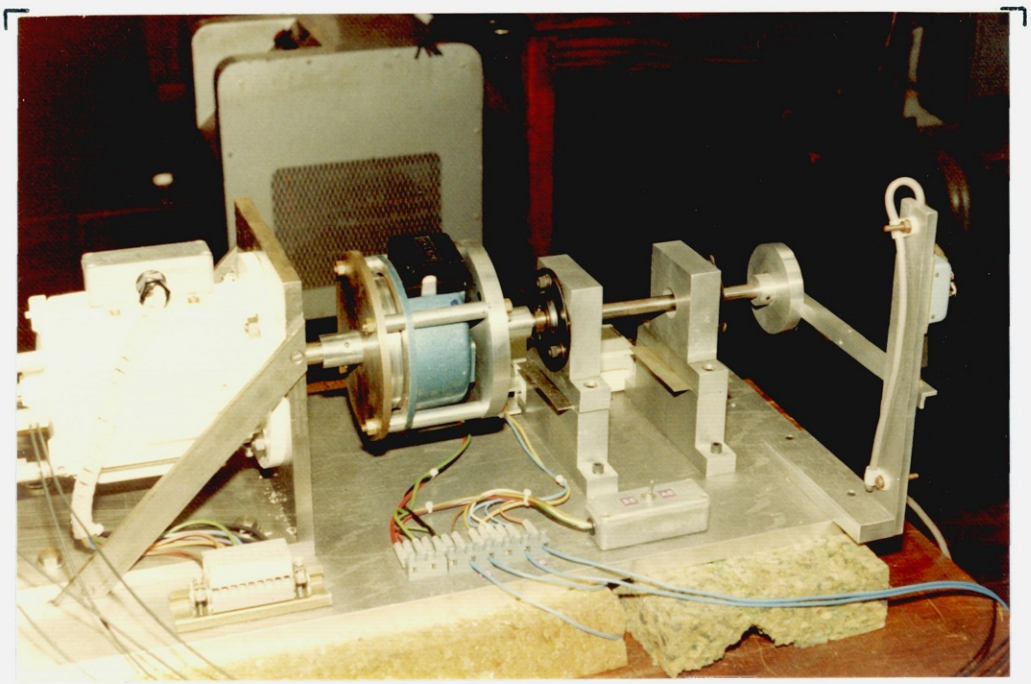


FIG.9.6. VIEW OF THE EDDY CURRENT DYNAMOMETER

been further reduced by the use of self-aligning bearings, but this was not felt to be worthwhile. The "stiction" on the dynamometer was measured and found to have a worst case value of 29.4mNm. The stiction was not found to be a problem in practice, since the vibration of the test rig tended to overcome the stiction even when small load torques were applied.

The eddy current brake was energised from a variac, and the imposed load torque was reduced to zero after each torque reading. This procedure allowed any drift in the zero-torque load cell reading to be quickly detected.

It was found in practice that the eddy current brake dynamometer was inadequate, since it could not load the 7 phase motor sufficiently. This can be confirmed by reference to the torque measurements presented in sub-sections 9.3.3 and 9.3.4. It was therefore necessary to test the maximum torque capability of the motor by another means. This was done by coupling a grooved 10.4cm diameter pulley directly onto the motor shaft (having previously removed the eddy current brake assembly). A piece of string was held under tension in the groove of the pulley by means of spring balances attached to each end. The spring balances were hand-held, and by suitably tensioning the string it was possible to load the 7 phase motor up to its full output torque. Unfortunately, tests at very high speeds with the "string dynamometer" were rather unsuccessful owing to rapid burning of the string. An attempt to use wire to overcome this problem failed, because insufficient friction existed between it and the pulley.

The very good reliability of the inverter during the tests, is probably due in part, to the fact that it was never fully loaded by the motor up to its 3kW capability. However, probably more significant is the fact that the total inductance per phase (due to the motor winding and the protection component inductor), is only of the order

of 1.35mH. This value of inductance is very small in comparison with the values found in "standard" motor windings. Such a small value reduces the stresses imposed by the load on the inverter switching components, and so helps to increase the reliability of the system.

Finally, it is worth mentioning that high-density foam rubber was wedged round the motor during the very high speed tests to damp the vibration and noise.

9.3 The Measurement Tests

This section contains details of all the performance tests carried out on the 7 phase motor. The scope of the tests was limited by both the lack of available time and the inadequacies of the test rig. Nevertheless, it is felt that the results obtained from the tests are valuable and so warrant presentation.

9.3.1 General Comments on the System Performance

There are a number of general comments about the motor system performance that are worth mentioning.

The cogging torque of the 7 phase motor was significant, and the measured value was found to be 0.271Nm as described in Chapter 6. There were 14 distinct "resting" locations for the rotor caused by the 14 open slots in the stator.

There were no problems in starting the motor. The large cogging torque did cause the rotor to move somewhat jerkily until synchronisation was achieved, but this did not trouble the starting procedure software. The motor ran "smoothly" in either direction. It was surprising how quiet the motor was in operation, especially in view of the large cogging torque which might normally be expected to make a motor noisy. The motor was virtually silent at

speeds of up to 10000 rpm. However, above 10000 rpm the noise output did rise, and can best be described as being similar to the whine emitted by gas turbine engines.

A combination of severe vibration in the eddy current brake assembly, and objections to the noise levels and safety aspects of the test rig, led to a decision to limit torque tests using the eddy current dynamometer to speeds below 16000 rpm. There was some concern that the dural disc attached to the motor shaft might disintegrate if severe vibration occurred over a long period. Therefore, for the no-load tests at speeds in excess of 16000 rpm, the dural disc was removed from the 7 phase motor shaft and the eddy current brake was taken off the test rig.

A maximum speed of 25000 rpm was achieved during the no-load tests. This speed was in fact the limit for the microprocessor control software. At 25000 rpm there are 23333 SEG interrupts a second, and the microprocessor is almost fully occupied by the autopiloting task. There is no reason why the motor should not reach its designed top speed of 30000 rpm if the software is improved (see Chapter 4).

The motor was extremely noisy at 25000 rpm but vibration was not excessive. This is remarkable because the rotor was not dynamically balanced during its manufacture. The motor was examined at the end of the tests for signs of damage. The rotor was in its original condition, and the bearings were very smooth and free running.

All the motor tests were performed with the motor rotating in the same direction. The chosen direction was reverse (selected via the appropriate keypad control button). This direction was used so that in the event of the dural disc coming off the motor shaft, it would hopefully run away from the test area into an unoccupied area of the laboratory.

It was observed during the tests that the motor ran well with less than 7 phases connected. At one point in the tests, when one of the inverter's hexfet boards was being repaired, only 6 phases were connected and yet the motor still started and ran smoothly. The possibilities of cutting out phases in light-load conditions is discussed in the conclusions to the thesis (Chapter 10).

Some jitter on the phase current waveforms was observed during the early tests whenever the motor speed exceeded 16400 rpm. The jitter was found to be due to the incorrect orientation of the optical position sensor disc on the rotor shaft, as explained in Chapter 4. A photograph of the current waveform jitter, at a speed of 16430 rpm with a 0° load angle, is shown in fig. 9.7. The orientation of the optical position sensor disc was corrected and the jitter problem disappeared. Fig. 9.8 is a photograph which shows "jitter-free" waveforms at a motor speed of 22250 rpm with a 0° load angle.

The "square-wave" matching of the motor and inverter was excellent. The back-emf generated in each phase winding was close to the desired waveshape. A photograph of the open-circuit back-emf in phase 1 is shown in fig. 9.9. The back-emf rises and falls linearly over the period taken for the rotor pole tips to sweep over approximately one slot pitch. The rise and fall times could possibly be shortened by closing the stator slots to reduce rotor flux fringing effects. The good match between the inverter output voltage and the motor back-emf is illustrated by the photograph in fig. 9.10. The photograph was taken with the motor unloaded at 15730 rpm and with a 0° load angle.

The efficiency of the hexfet inverter was found to be around 90% (this efficiency does not include the power consumed from the +5 volt and +15 volt supplies within the inverter). The power losses within the inverter can be

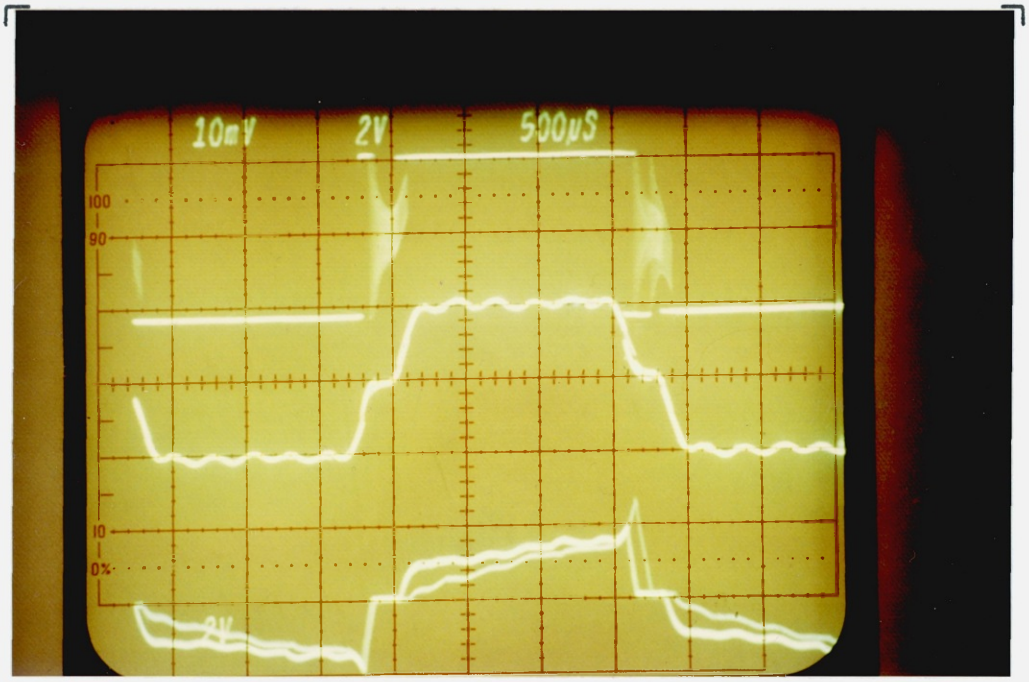


FIG. 9.7. PHASE CURRENT WAVEFORM JITTER AT 16430 RPM CAUSED BY THE INCORRECT ORIENTATION OF THE OPTICAL POSITION SENSOR DISC. TOP TRACE : INVERTER OUTPUT VOLTAGE (20V/cm). MIDDLE : MOTOR PHASE VOLTAGE (20V/cm). BOTTOM : PHASE CURRENT (1A/cm). — 500µs/cm

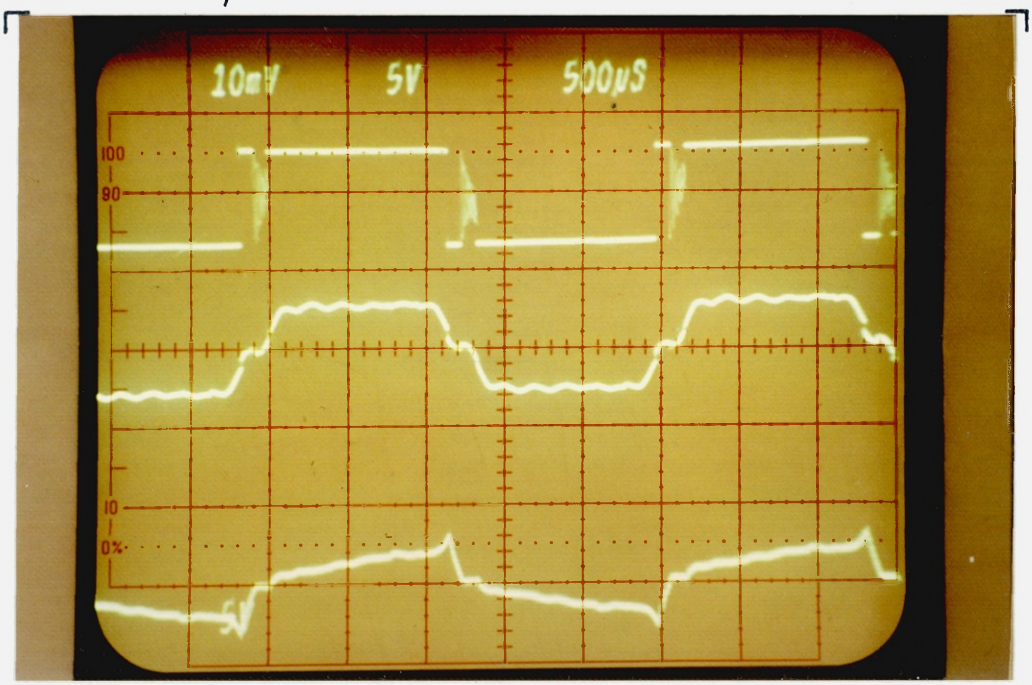


FIG. 9.8. PHOTOGRAPH SHOWING THE ABSENCE OF WAVEFORM JITTER WITH CORRECT ORIENTATION OF THE OPTICAL POSITION SENSOR DISC (SPEED = 22250 RPM). TOP TRACE : INVERTER OUTPUT VOLTAGE (50V/cm). MIDDLE : MOTOR PHASE VOLTAGE (50V/cm). BOTTOM : PHASE CURRENT (5A/cm). — 500µs/cm.

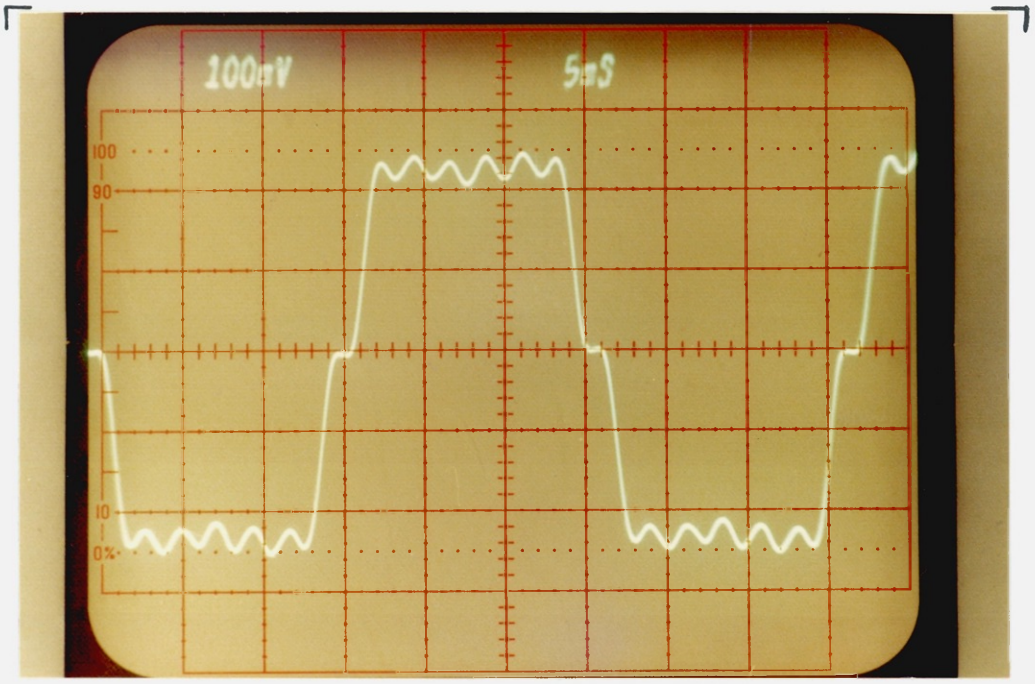


FIG.9.9. PHASE ONE OPEN-CIRCUIT BACK-EMF AT 1900 RPM (1V/cm, 5ms/cm)

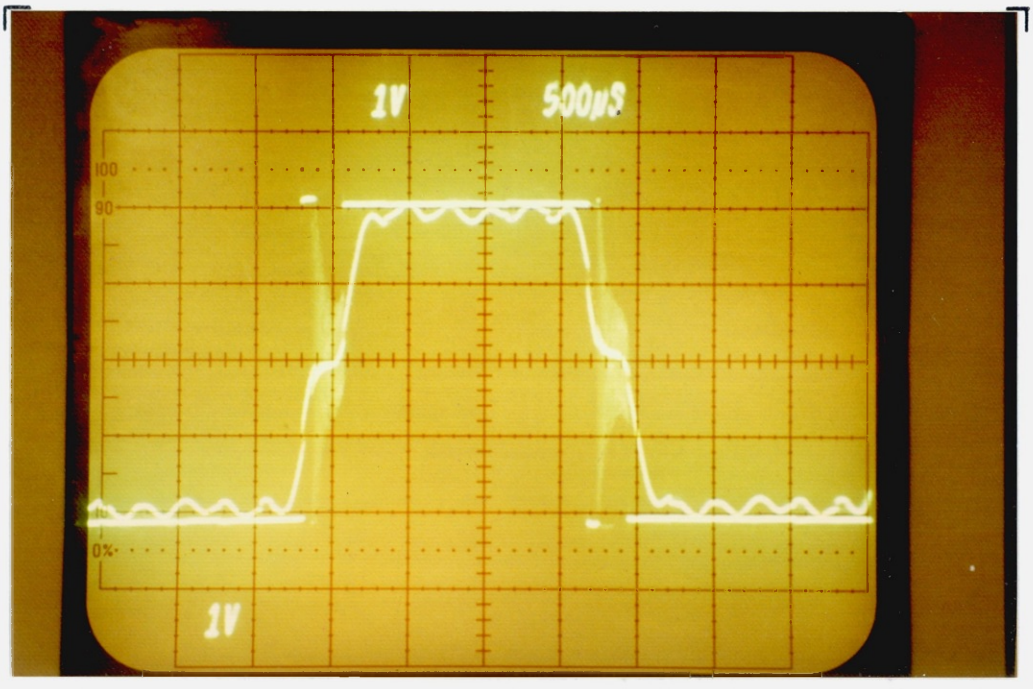


FIG.9.10. PHOTOGRAPH SHOWING THE GOOD MATCH BETWEEN THE INVERTER OUTPUT VOLTAGE AND THE MOTOR BACK EMF: 15730 RPM, ±21 VOLT INVERTER SUPPLY, 1·070A PHASE CURRENT, 0° LOAD ANGLE. BOTH TRACES 10V/cm, 500μs/cm.

accounted for by the on-resistances of the hexfet transistors.

For the benefit of the reader it is worth summarising at this point the average resistance and inductance per phase. The average resistance and inductance of the motor phase windings are 0.101Ω and 99μH respectively. The average resistance and inductance of each set of protection components are 1.086Ω and 1.25mH respectively (the resistance value includes the resistance of the inductor). Hence the total resistance and inductance per phase are 1.187Ω and 1.35mH respectively.

Finally the value of air-gap flux density at the beginning of the tests (as stated in Chapter 6) was 0.298T.

9.3.2 The Measurement of the Combined Value of the Friction, Windage, and Other Loss Torques Suffered by the Rotor of the 7 Phase Motor

It is generally necessary to measure the magnitude of the friction and windage losses suffered by a motor, so that a more accurate value of its electrical to mechanical energy conversion efficiency can be determined. The fact that the 7 phase motor was intended for very high speed operation, made the need for an assessment of windage losses even more desirable than usual. In addition it was evident during the early motor tests that significant stator eddy current losses were present, since the motor body was becoming warm even on no-load after about ten minutes running. The stator eddy currents are induced largely by the rotor flux. The energy dissipated by the eddy currents is transferred from the rotor shaft, and so manifests itself on the rotor as a braking torque. Hence the bearing friction torque, windage torque, and "stator-loss" torque all act as retarding torques on the rotor, and they can be lumped together as a total "Iron, Windage and Friction" (I.W.F.) loss torque, T_{lt}. The friction torque component is usually

more or less speed independent, but the windage and stator-loss torques are both speed dependent. Therefore, a loss torque curve has to be measured over the range of speeds that the motor operates.

9.3.2.1 Principle of the Loss Torque Measurement Method

The I.W.F. loss torque measurement method relies on a knowledge of the moment of inertia (J) of the rotor (and load if coupled). In addition it is necessary to obtain an accurate speed/time deceleration characteristic for the motor from the maximum speed of interest down to the minimum speed of interest. The only applied torque present within the motor during the measurement of its deceleration curve is the loss torque T_{lt} because the motor is unenergised. The nett torque within the motor must be zero and therefore it is possible to state:

$$T_{lt} = J.\frac{d\omega}{dt} \qquad 9.1$$

where ω is the angular velocity of the rotor in radians per second. In discrete form equation 9.1 becomes:

$$T_{lt} = J.\frac{\Delta\omega}{\Delta t} \qquad 9.2$$

or $$T_{lt} = J.\Delta\omega.(1/\Delta t) \qquad 9.3$$

Equation 9.3 is used in conjunction with the speed/time curve to calculate a loss-torque versus speed curve as follows. The speed/time curve is typically of the form shown in fig. 9.11. The speed axis of the speed/time curve is divided into a series of equal intervals $\Delta\omega$. For the example interval shown in fig. 9.11:

$$\Delta\omega = \omega_1 - \omega_2 \qquad 9.4$$

and the average speed ω_n at the middle of the speed interval

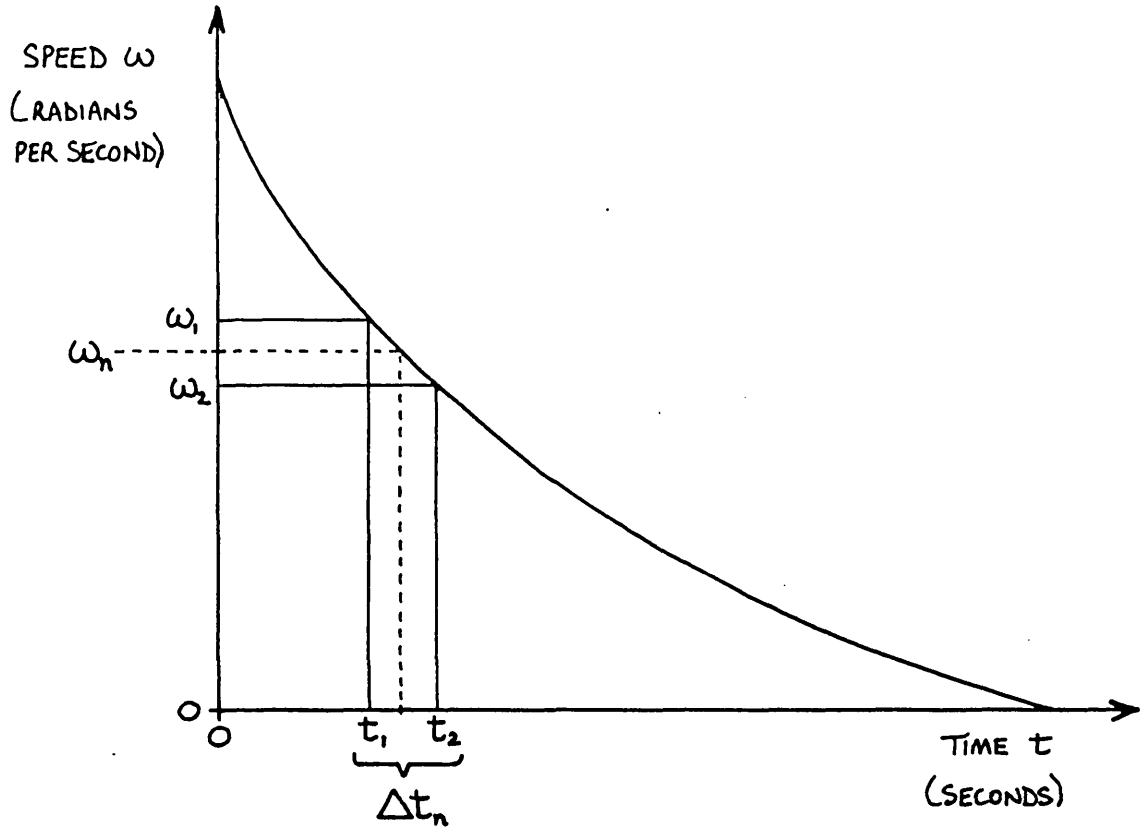


FIG. 9.11. TYPICAL DECELERATION SPEED/TIME CURVE

is given by:

$$\omega_n = (\omega_1 + \omega_2)/2 \qquad 9.5$$

The time interval Δt_n corresponding to the speed interval centred on ω_n is given by:

$$\Delta t_n = t_2 - t_1 \qquad 9.6$$

Substitution of the value of Δt_n into equation 9.3 with the appropriate values for the moment of inertia J and speed interval $\Delta\omega$ gives a value of $T_{lt(n)}$ corresponding to the average speed ω_n. Since J and $\Delta\omega$ are constants, equation 9.3 can be written as:

$$T_{lt(n)} = k/\Delta t_n \qquad 9.7$$

where $$k = J.\Delta\omega \qquad 9.8$$

It is then a simple matter using equation 9.7 with the Δt values obtained from the speed/time curve to calculate a set of $T_{lt(n)}$ values, which, when plotted against the appropriate values of ω_n, produce a loss-torque versus speed curve of the form shown in fig. 9.12. In practice the graphical basis of this method does mean that there is some scatter in the calculated loss-torque points and so it is necessary to draw a best fit curve through the points to obtain a usable characteristic.

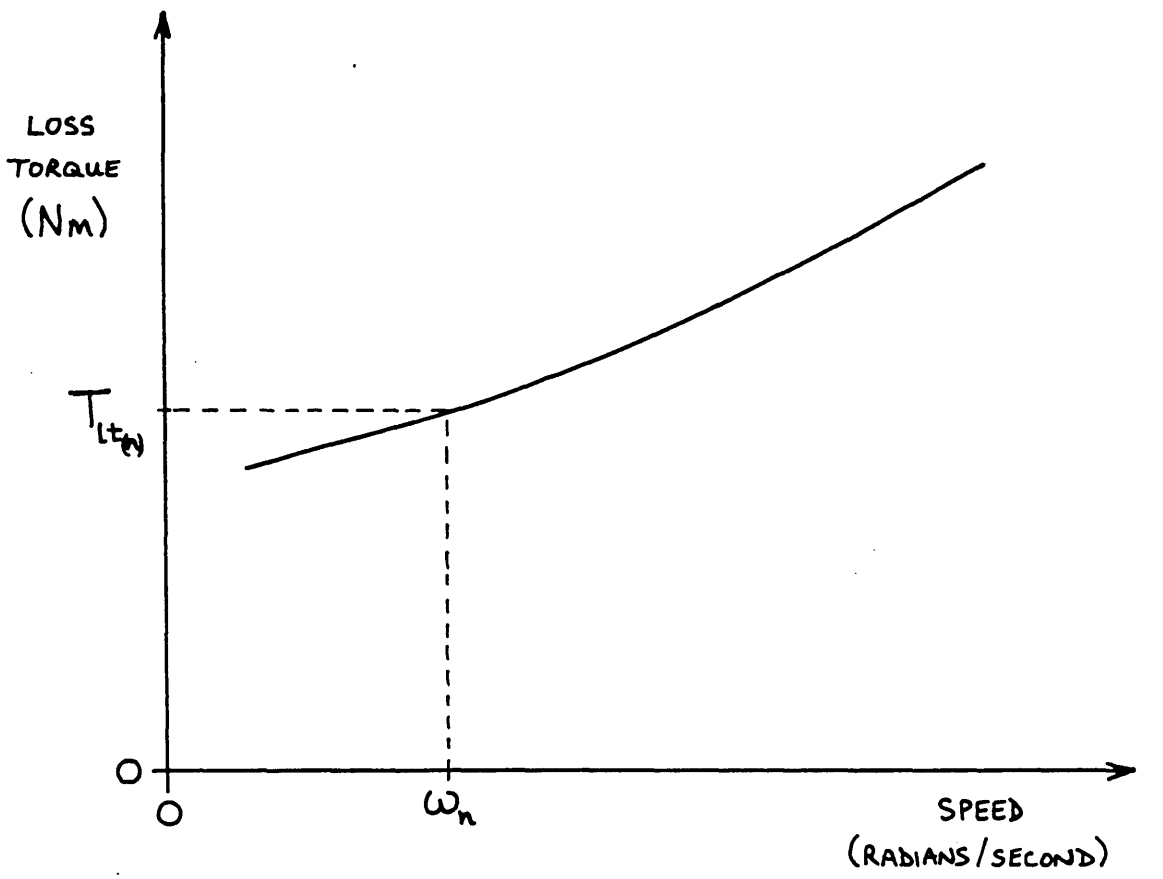


FIG. 9.12. TYPICAL LOSS-TORQUE VERSUS SPEED CHARACTERISTIC

Once an I.W.F. loss-torque characteristic has been obtained, it can be used to calculate an I.W.F. loss power curve. Alternatively, it can be combined with a measured torque/speed curve to give a gross motor torque versus speed characteristic.

9.3.2.2 Practically Determined Loss Characteristics

The speed/time deceleration characteristics were measured by means of a y:t plotter and a frequency to analogue voltage (F/V) converter. The SEG' pulses from the optical position detector were fed to the F/V converter so that an analogue voltage proportional to the rotor speed was available. This "speed voltage" was fed to the y axis input of the y:t plotter and the sweep time of the t axis was set to suit the run-down time of the rotor. With a piece of graph paper aligned on the plotter bed, the motor was run up to the maximum speed of interest and the recording pen was lowered onto the paper. The t axis was then triggered to begin the pen sweeping across the paper and the microprocessor system controlling the motor was reset. Resetting the microprocessor causes the MOSFET inverter to be immediately disabled and hence the unenergised motor decelerates to rest with its speed being recorded via the pen displacement on the y axis.

Three deceleration speed/time characteristics were obtained as follows: 10600 rpm to 0 rpm; 15400 rpm to 0 rpm; and 23900 rpm to 0 rpm. The 23900 rpm characteristic was intended for use with the no-load high speed test results. As mentioned earlier, speeds above about 16000 rpm were only possible with the dural eddy current brake disc removed from the rotor shaft. Hence the 23900 rpm speed/time characteristic did not suffer from the windage losses due to the dural disc, whereas the 10600 rpm and 15400 rpm characteristics did.

The speed/time characteristics were arranged to be as

large as possible (that is, to fill the available paper area) so that the accuracy of the Δt values measured from the curves could be maximised. It was difficult to reduce the characteristics to a suitable size for display in this thesis and so they have been omitted.

The moments of inertia of the rotor assembly and the dural disc were calculated by standard techniques and were found to have values of 2.93018×10^{-4} Kg.m^2 and 3.75202×10^{-4} Kg.m^2 respectively. Hence the total moment of inertia for the 10600 rpm and 15400 rpm characteristics was 6.6822×10^{-4} Kg.m^2.

The values of $\Delta\omega$ chosen for use in the calculation of the loss-torque curves were 21.468 radians per second for the 10400 rpm characteristic, 41.888 radians per second for the 15400 rpm characteristic, and 85.347 radians per second for the 23900 rpm characteristic.

By using equation 9.8, values for the constant "k" of 0.01435, 0.02799, and 0.02501 were calculated for the 10600 rpm, 15400 rpm, and 23900 rpm characteristics respectively. The k values were then used in equation 9.7 in combination with the appropriate sets of measured Δt values to produce the loss-torque curves. The errors in measuring the Δt values resulted in a large scatter in the plotted points. Hence the only way to obtain meaningful and useful characteristics was to draw best-fit curves through the points. Figs. 9.13 and 9.14 show the loss-torque versus speed characteristics obtained from the 15400 rpm and 23900 rpm speed/time deceleration curves respectively. The loss-torque versus speed characteristic obtained from the 10600 rpm speed/time curve was found to be identical to the "low speed" (0 to 10000 rpm) section of the characteristic shown in fig. 9.13 and so it has not been presented. The calculated points have not been shown on the characteristics for reasons of clarity, but the author fully acknowledges that the characteristics are smoothed results and

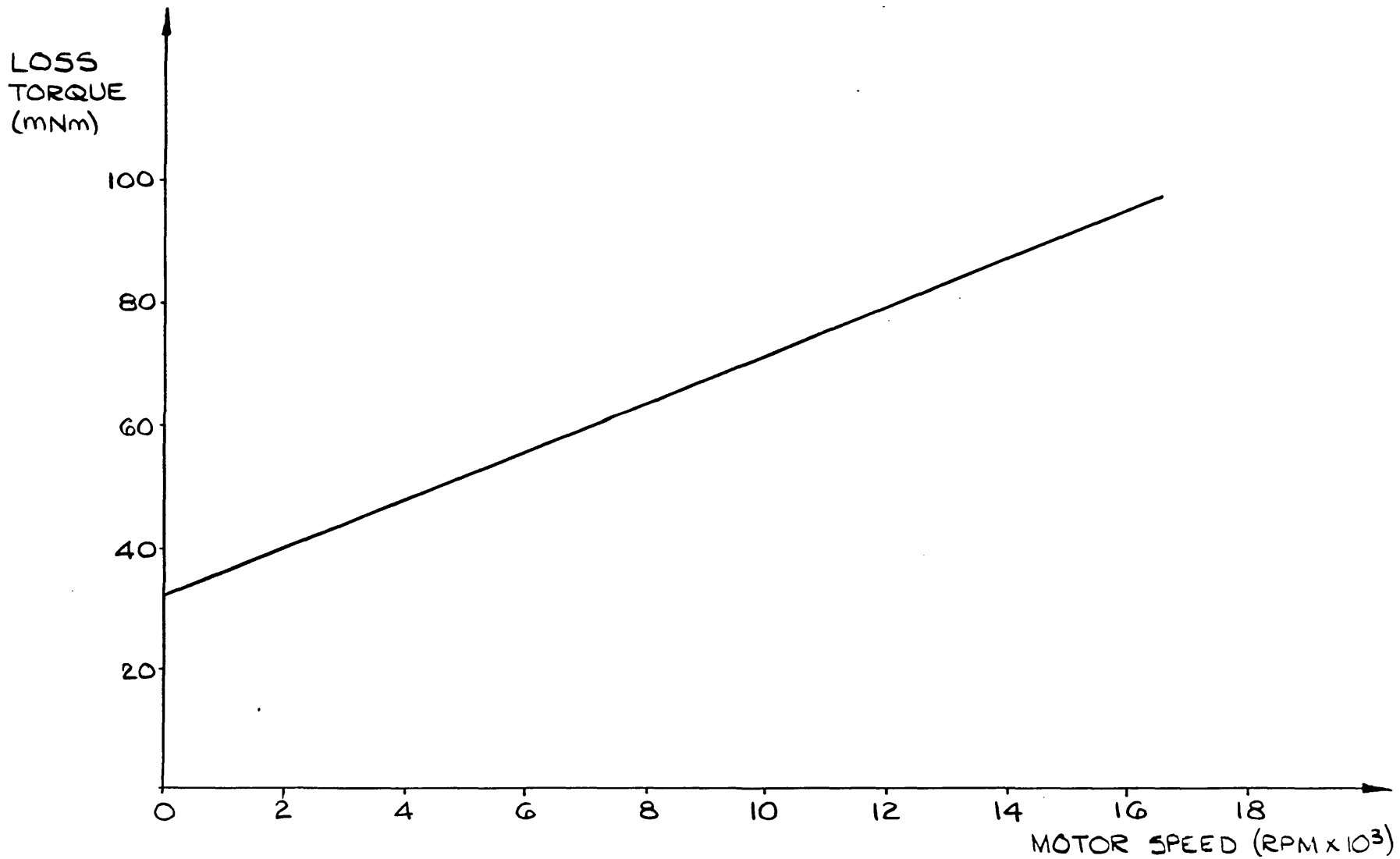


FIGURE 9.13 LOSS-TORQUE VERSUS SPEED FOR ROTOR WITH DURAL DISC ON

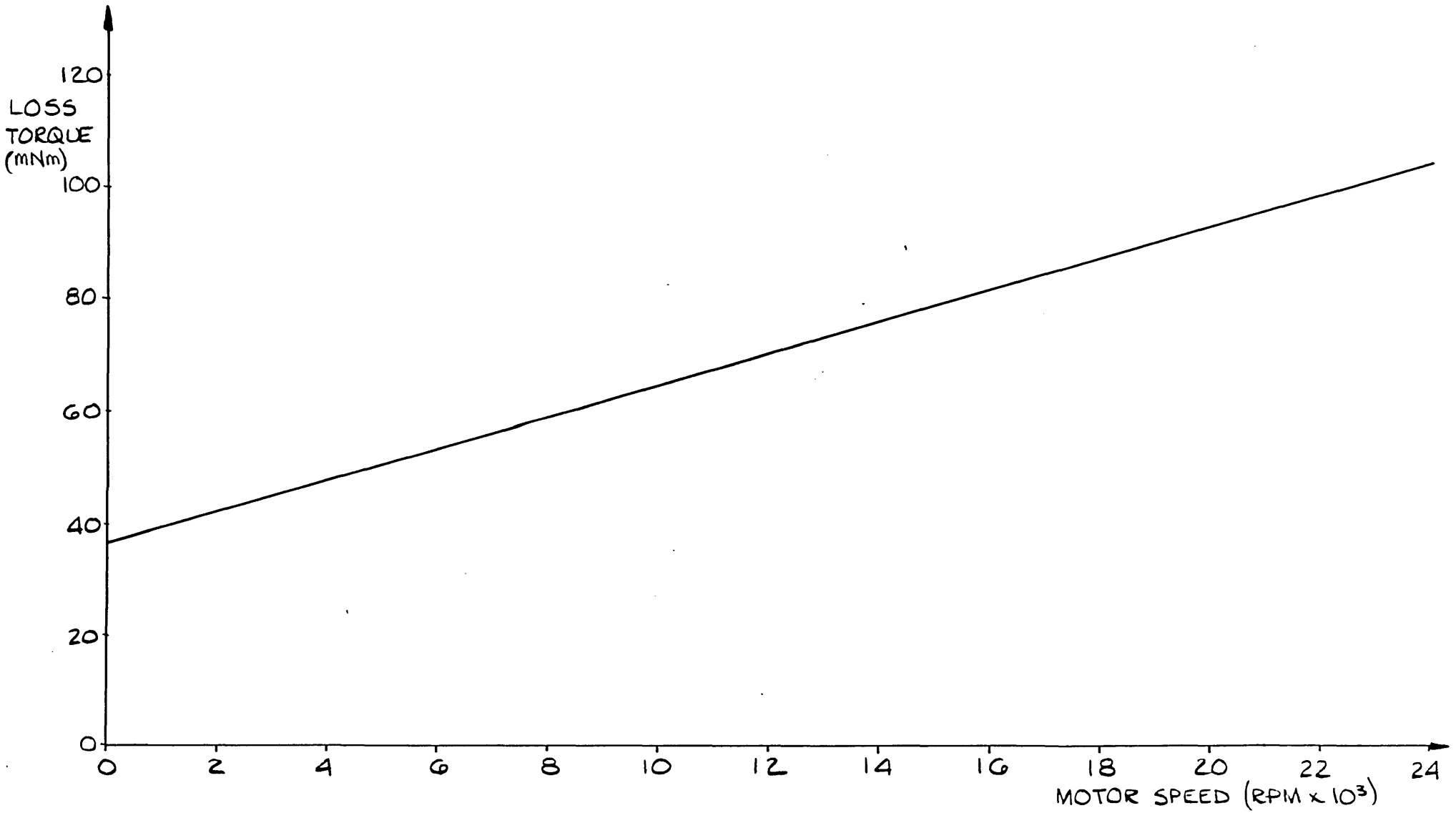


FIGURE 9.14 LOSS-TORQUE VERSUS SPEED FOR ROTOR WITHOUT DURAL DISC ON

are not actual "raw" experimentally obtained results. It is interesting to note that the smoothed characteristics show that the loss-torque increases linearly with speed. However, in view of the simple nature of the experimental apparatus used for the loss-torque measurements, it is felt that the averaged results should be regarded only as giving a first-order idea of the magnitude of the losses.

The linear loss-torque characteristics can be expressed easily in equation form. The equation for fig. 9.13 is:

$$T_{lt} = (3.9 \times 10^{-6}.S + 3.219 \times 10^{-2}) \quad \text{Nm} \qquad 9.9$$

where S is the rotor speed in revolutions per minute. The equation for fig. 9.14 is:

$$T_{lt} = (2.846 \times 10^{-6}.S + 3.626 \times 10^{-2}) \quad \text{Nm} \qquad 9.10$$

The maximum speeds that can be substituted into equations 9.9 and 9.10 are 16400 rpm and 23900 rpm respectively.

The loss-torque characteristics are used in the following load-test sub-sections in an attempt to assess the efficiency of the motor.

9.3.3 The 7 Phase Motor Performance Characteristics for Constant Inverter Supply Voltages

A series of tests were performed on the 7 phase motor system with the inverter supply voltage rails held at various constant values. Voltages in the range between ±10 volts up to ±22 volts were supplied to the inverter and the resulting torque/speed characteristics were measured using the eddy current dynamometer. Motor load angles of 0°, 6.43°, and 12.86° were selected during the tests to investigate the effect of load angle on the motor's performance. These load angles are henceforth referred to as 0°, 6°, and 13° respectively.

Both the nett and gross motor torque/speed characteristics were obtained for each voltage and load angle selection. The gross torque/speed characteristics were constructed by combining the measured nett torque/speed characteristics with the loss-torque/speed curve shown in fig. 9.13.

The nett torque/speed characteristics for load angles of 0°, 6°, and 13° with an inverter supply voltage of ± 10 volts are shown in fig. 9.15. The corresponding gross torque/speed characteristics are shown in fig. 9.16. The smooth nature of the performance curves is a good indication that the measurement technique was good.

Fig. 9.17 is a photograph showing the waveforms of the inverter output voltage, motor phase voltage, and motor current on phase 1 for the 0° load angle no-load condition with ± 10 volts supplied to the inverter (speed = 7150 rpm). The extremely good square-wave current waveshape can be clearly seen. There is a short freewheel period in the inverter output voltage waveform caused by the inductance of the phase winding and protection components. The "fuzzy" areas in the zero volt sections of the inverter output voltage waveform are actually due to electromagnetic pickup from the 100kHz hexfet gate driver power supplies. The reactive impedance of the 1.25mH inductor in each set of protection components prevents the 100kHz pickup from being shorted to ground via the low impedance motor phase winding. During the "on" sections of the inverter output voltage waveform, one or other of the hexfet transistors is conducting and so the 100kHz pickup is absorbed into one of the inverter power supply rails. The pickup is also absorbed into the power supply rails during the freewheel periods. The pickup is therefore only visible during the zero volt periods of the waveform when both hexfets are off.

The change in the waveforms shown in fig. 9.17 when the motor was loaded, is illustrated by the photograph of

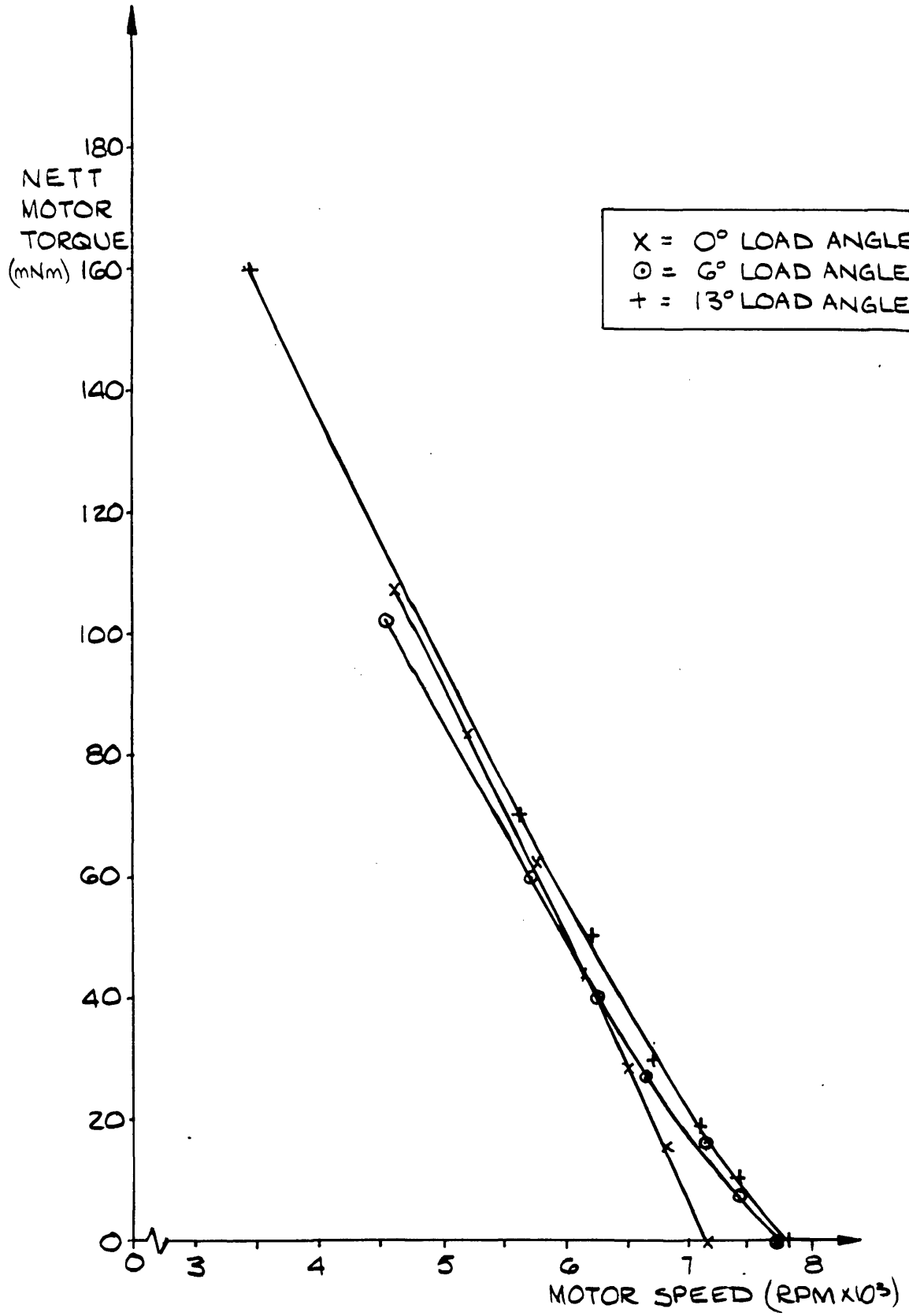


FIGURE 9.15 7 PHASE MOTOR: ± 10 VOLT SUPPLY TO INVERTER - NETT TORQUE FOR LOAD ANGLES OF 0°, 6°, AND 13°

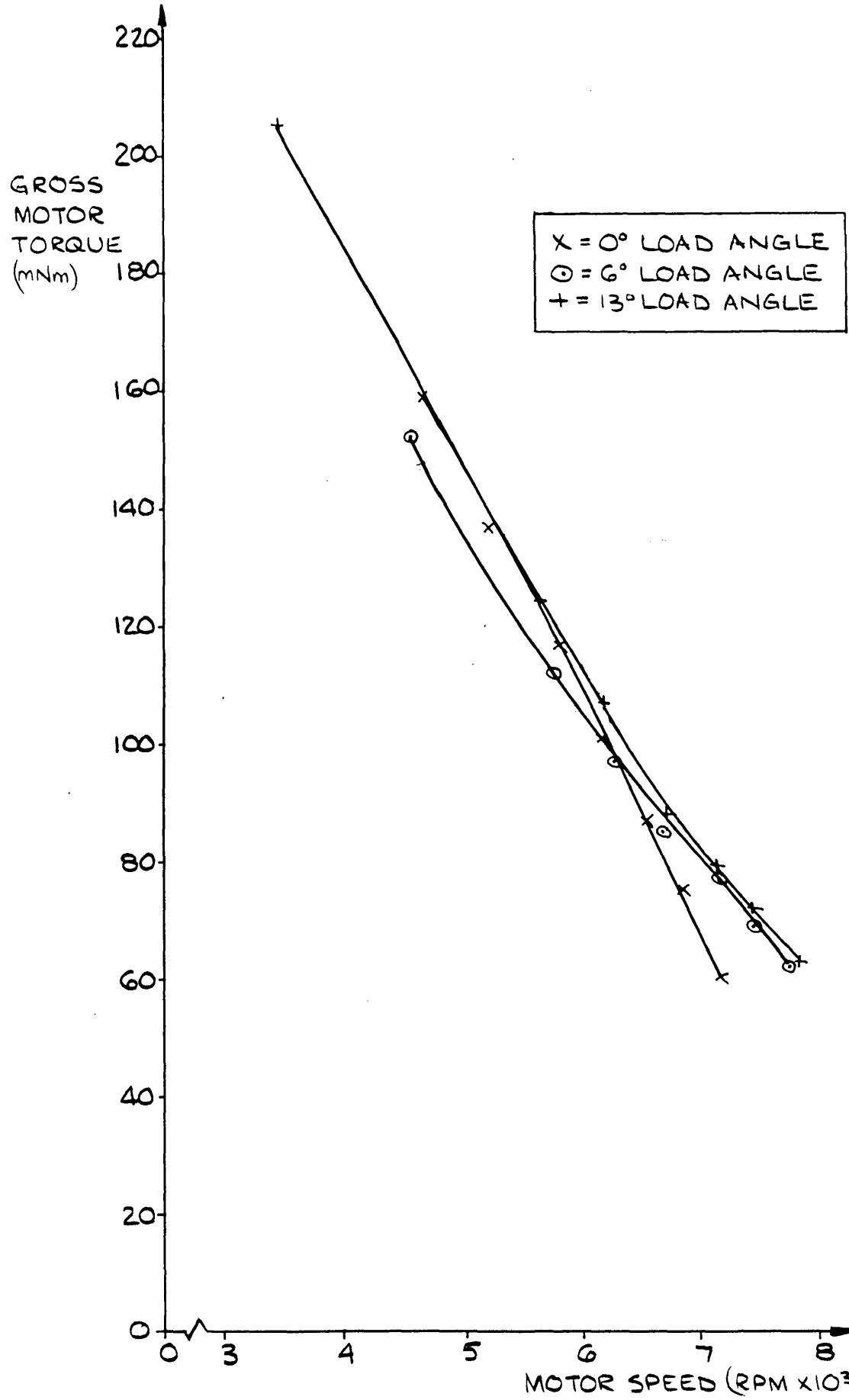


FIGURE 9.16 7 PHASE MOTOR: ± 10 VOLT SUPPLY TO INVERTER - GROSS TORQUE FOR LOAD ANGLES OF 0°, 6° AND 13°

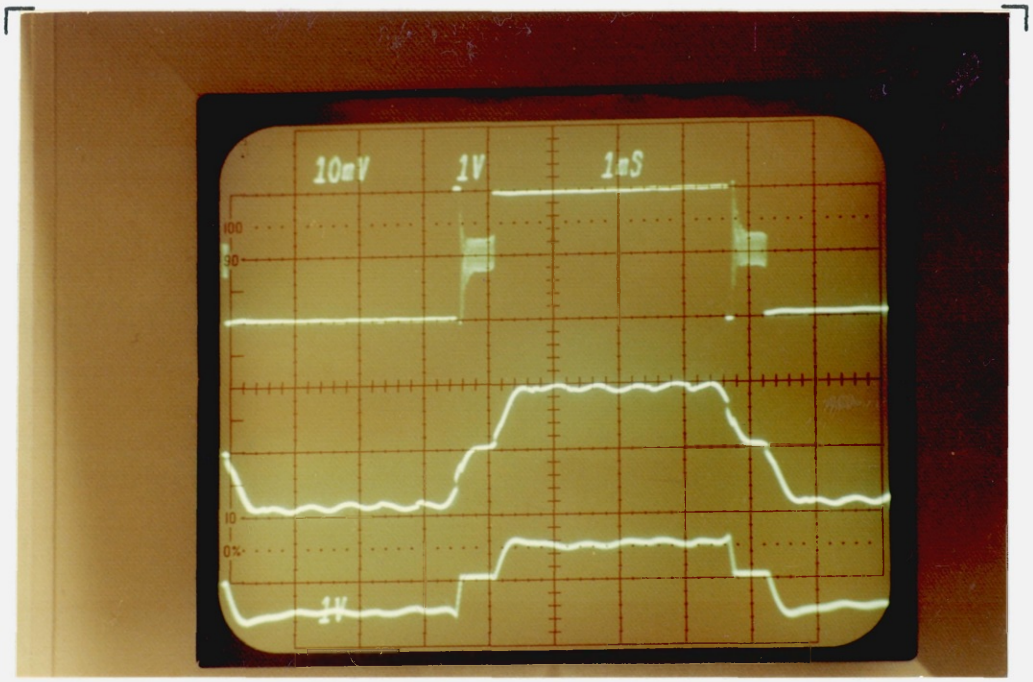


FIG. 9.17. PHASE ONE WAVEFORMS : 0° LOAD ANGLE ; ±10 VOLT INVERTER SUPPLY ; NO LOAD (0 NETT TORQUE) 7150 RPM — TOP : INVERTER OUTPUT VOLTAGE (10V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (10V/cm) — BOTTOM : PHASE CURRENT (2A/cm) — 1ms/cm.

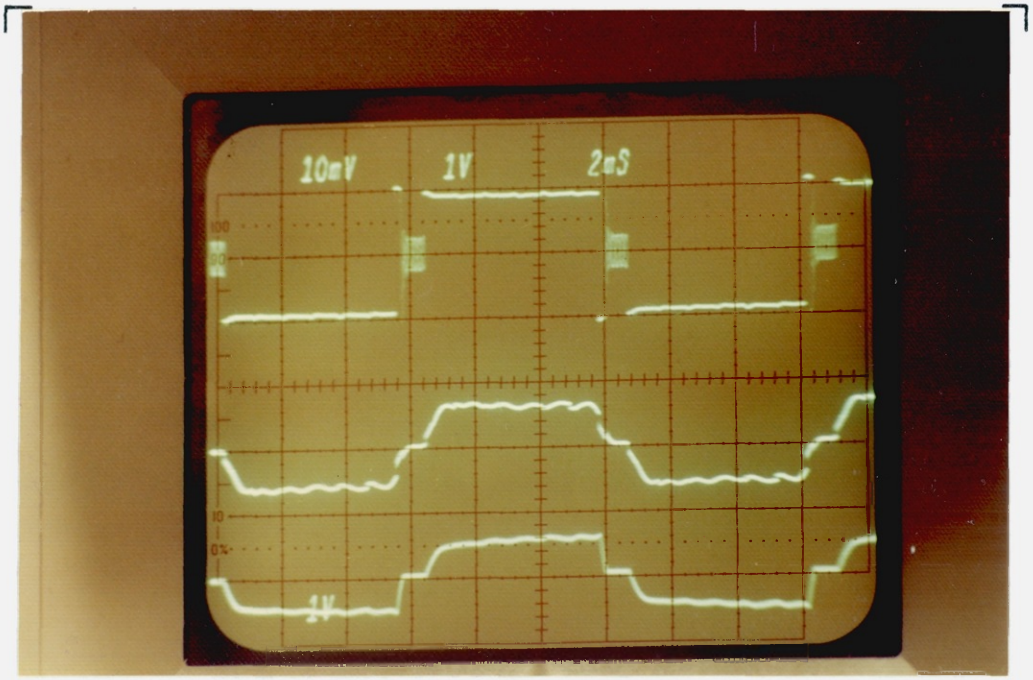


FIG. 9.18. PHASE ONE WAVEFORMS : 0° LOAD ANGLE ; ±10 VOLT INVERTER SUPPLY : 108mNm NETT TORQUE, 4630 RPM — TOP : INVERTER OUTPUT VOLTAGE (10V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (10V/cm) — BOTTOM : PHASE CURRENT (5A/cm) — 2ms/cm.

fig. 9.18. This photograph was taken with the motor producing a nett output torque of 108mNm at a speed of 4630 rpm (±10 volt inverter supply voltages; 0° load angle). The motor phase current reaches a peak value of about 2.5 amps and it has a longer rise time than under no-load conditions. The current waveshape has a "rounded" rising edge but even so, it can still be correctly described as quasi-square in form. The fact that the current waveform appears to depend on the magnitude of the current is discussed in section 9.6.

The torque/speed characteristics presented in figs. 9.15 and 9.16 show that the load angle does affect the motor's performance to some extent. The changes in the torque/speed characteristics are a result of the changes in the phase current waveshape. Waveform photographs taken during the 6° and 13° load angle operation of the motor with ±10 volts supplied to the inverter are shown in figs. 9.19, 9.20, 9.21, and 9.22. Each photograph shows the inverter output voltage waveform, the motor phase voltage waveform, and the phase current waveform. The waveforms in fig. 9.19 are for 6° load angle operation with zero nett load torque at a speed of 7730 rpm. The current waveform rises steeply due to the 6° load angle and it peaks at just less than 2.0 amps before falling to a value of about 1.0 amp for the majority of the on period. The current fall time at the end of each conduction period is also rapid. Fig. 9.20 shows how the phase current waveform "squares up" as the motor is loaded. The waveforms in fig. 9.20 are for operating conditions of 6° load angle, 102mNm nett torque and 4520 rpm. The current waveform has the desired quasi-square shape and has an almost constant on-value of about 3.0 amps. The 6° load angle therefore enables a better square wave current to be achieved with "large" phase currents than is possible with a 0° load angle (compare fig. 9.20 with fig. 9.18). Figs. 9.21 and 9.22 show waveforms for 13° load angle operation of the motor: fig. 9.21 is for zero nett torque at 7800 rpm; fig. 9.22 is for

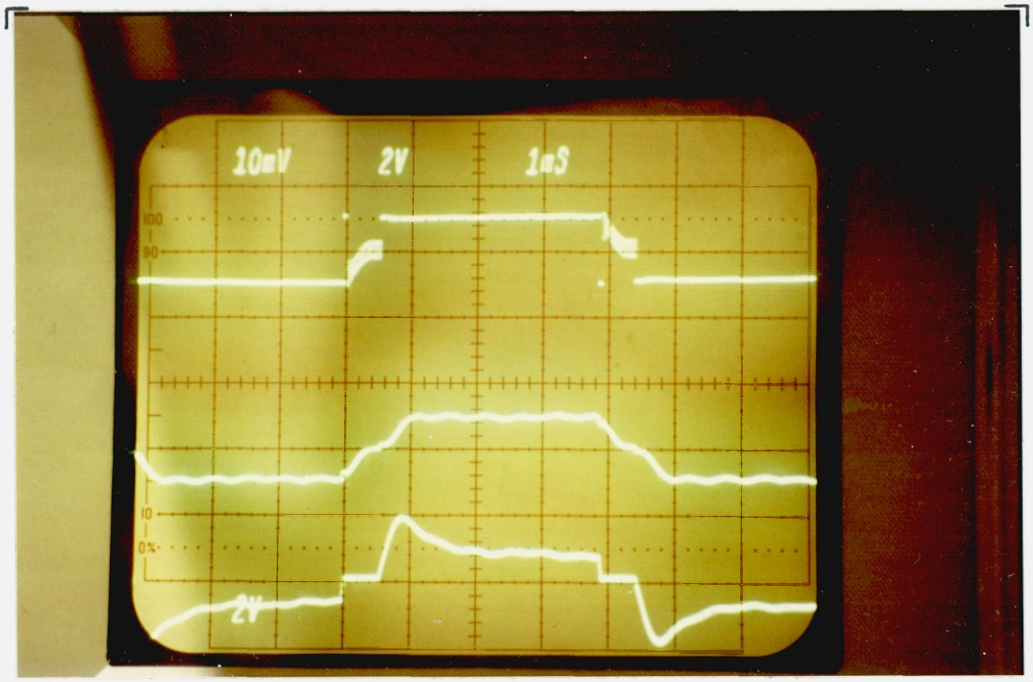


FIG. 9.19. PHASE ONE WAVEFORMS : 6° LOAD ANGLE ; ±10 VOLT INVERTER SUPPLY ; NO LOAD (0 NETT TORQUE), 7730 RPM — TOP : INVERTER OUTPUT VOLTAGE (20V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (20V/cm) — BOTTOM : PHASE CURRENT (2A/cm) —— 1ms/cm.

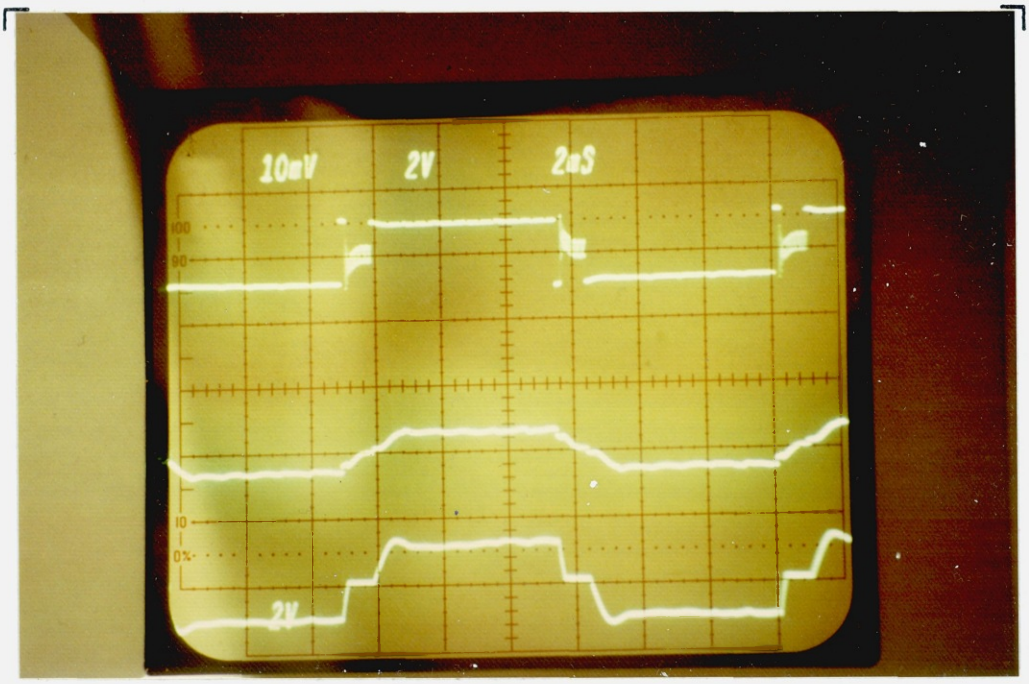


FIG. 9.20. PHASE ONE WAVEFORMS : 6° LOAD ANGLE ; ±10 VOLT INVERTER SUPPLY ; 102 mNm NETT TORQUE, 4520 RPM — TOP : INVERTER OUTPUT VOLTAGE (20V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (20V/cm) — BOTTOM : PHASE CURRENT (5A/cm) —— 2ms/cm.

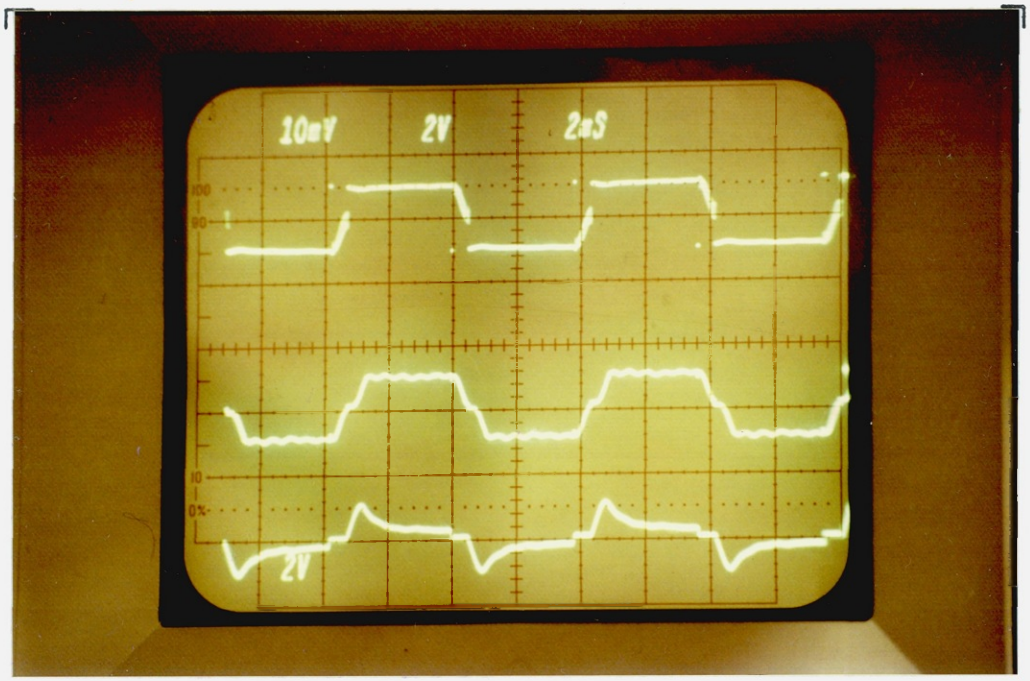


FIG. 9.21. PHASE ONE WAVEFORMS : 13° LOAD ANGLE ; ±10 VOLT INVERTER SUPPLY ; NO LOAD (NO NETT TORQUE), 7800 RPM — TOP : INVERTER OUTPUT VOLTAGE (20V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (20V/cm) — BOTTOM : PHASE CURRENT (5A/cm) — 2ms/cm.

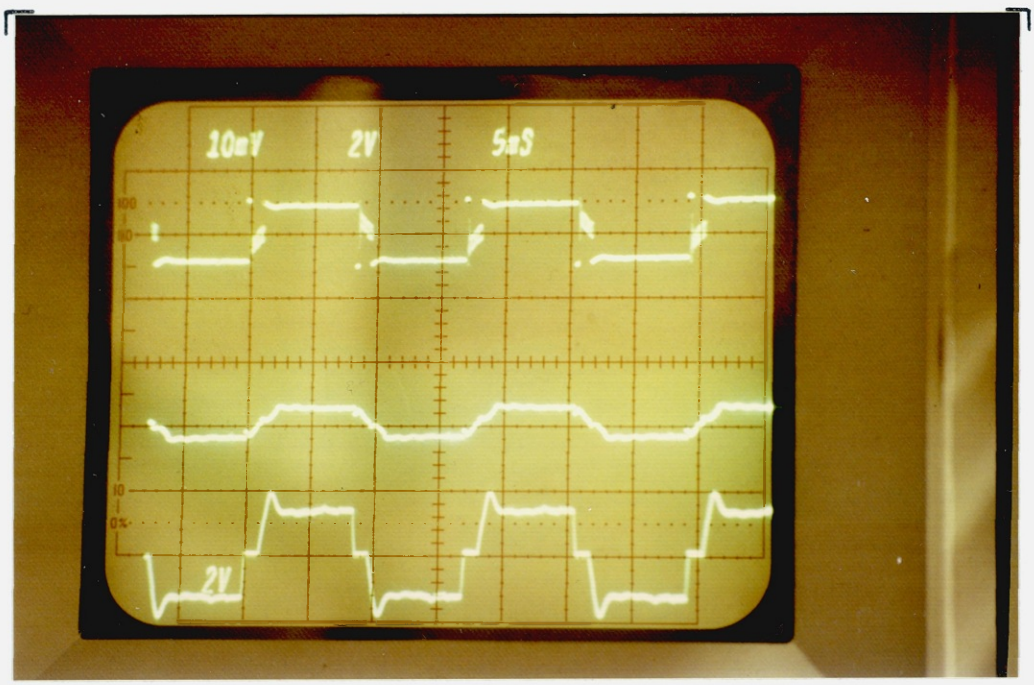


FIG. 9.22. PHASE ONE WAVEFORMS : 13° LOAD ANGLE ; ±10 VOLT INVERTER SUPPLY ; 160mNm NETT TORQUE, 3430 RPM — TOP : INVERTER OUTPUT VOLTAGE (20V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (20V/cm) — BOTTOM : PHASE CURRENT (5A/cm) — 5ms/cm.

160mNm nett torque at 3430 rpm. Fig. 9.21 shows that the phase current peaks very sharply under no-load conditions, the peak and average values of current being about 3.0 amps and 1.0 amps respectively. However, the "on-load" current waveform shown in fig. 9.22 is very close to the desired quasi-square shape, although there is still a small switch-on overshoot up to 5.0 amps before the current settles down to an on-value of about 3.5 amps.

The waveforms shown in figs. 9.17 to 9.22 inclusive, show that the ability to adjust the motor load angle is a useful aid in achieving a good quasi-square phase current waveform over a range of operating torques.

The motor torque (nett and gross), motor efficiency (nett and gross), overall system efficiency (nett and gross), r.m.s. phase current, and inverter efficiency are shown plotted against speed for load angles of 0°, 6°, and 13° with a ± 10 volt inverter supply voltage in figs. 9.23 to 9.28 inclusive. For guidance, fig. 9.23 presents the nett motor torque, nett motor efficiency, phase current, and inverter efficiency for 0° load angle operation: fig. 9.24 presents the gross motor torque, gross motor efficiency, and the nett and gross overall system efficiencies for 0° load angle operation. Figs. 9.25 and 9.26 present the results in a likewise manner for 6° load angle operation and fig. 9.27 and 9.28 present the 13° load angle results.

Three points are worth mentioning with regard to the results presented in figs. 9.23 to 9.28.

Firstly, the plotted points are such that smooth curves can be drawn through them. There is therefore good reason to believe that the experimental technique was accurate and reliable.

Secondly, the inverter efficiency is consistently in the region of 90%. When compared with the nett motor

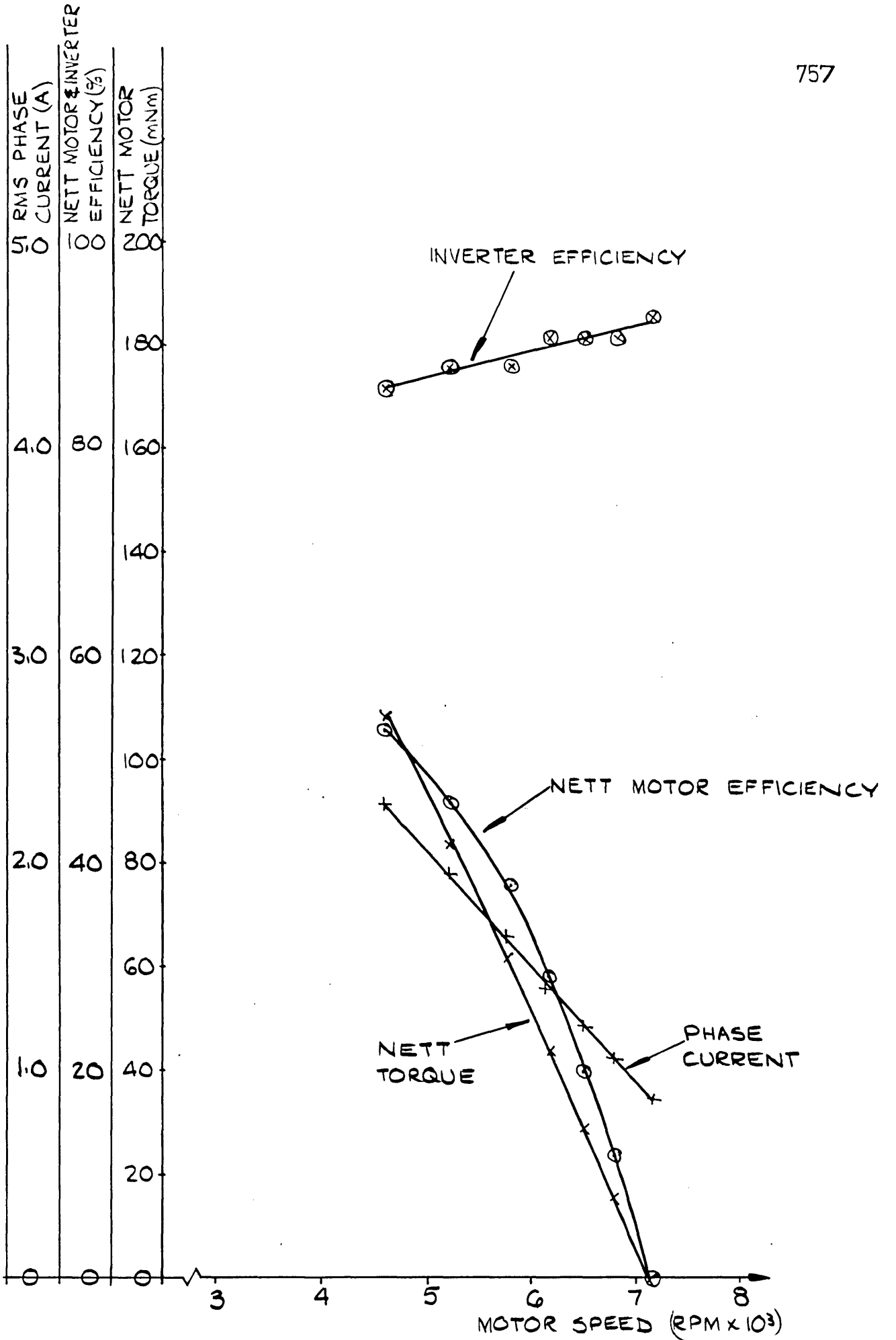


FIGURE 3.23 7 PHASE MOTOR: 0° LOAD ANGLE; ± 10 VOLT SUPPLY TO INVERTER - NETT TORQUE, NETT MOTOR EFFICIENCY, PHASE CURRENT, AND INVERTER EFFICIENCY.

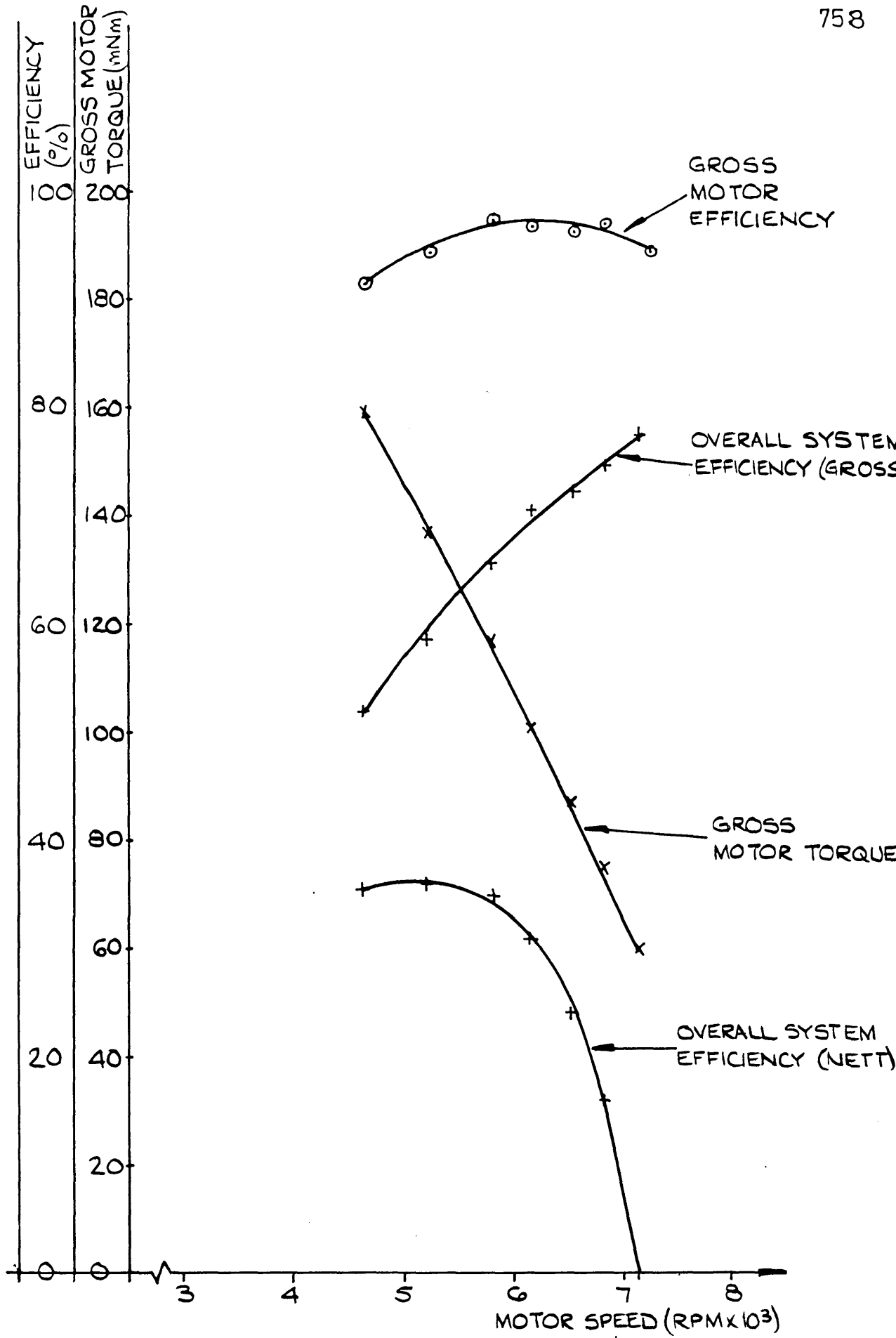


FIGURE 9.24 7 PHASE MOTOR: 0° LOAD ANGLE; ±10 VOLT SUPPLY TO INVERTER - GROSS MOTOR TORQUE, GROSS MOTOR EFFICIENCY, OVERALL SYSTEM EFFICIENCY (GROSS) AND OVERALL SYSTEM EFFICIENCY (NETT)

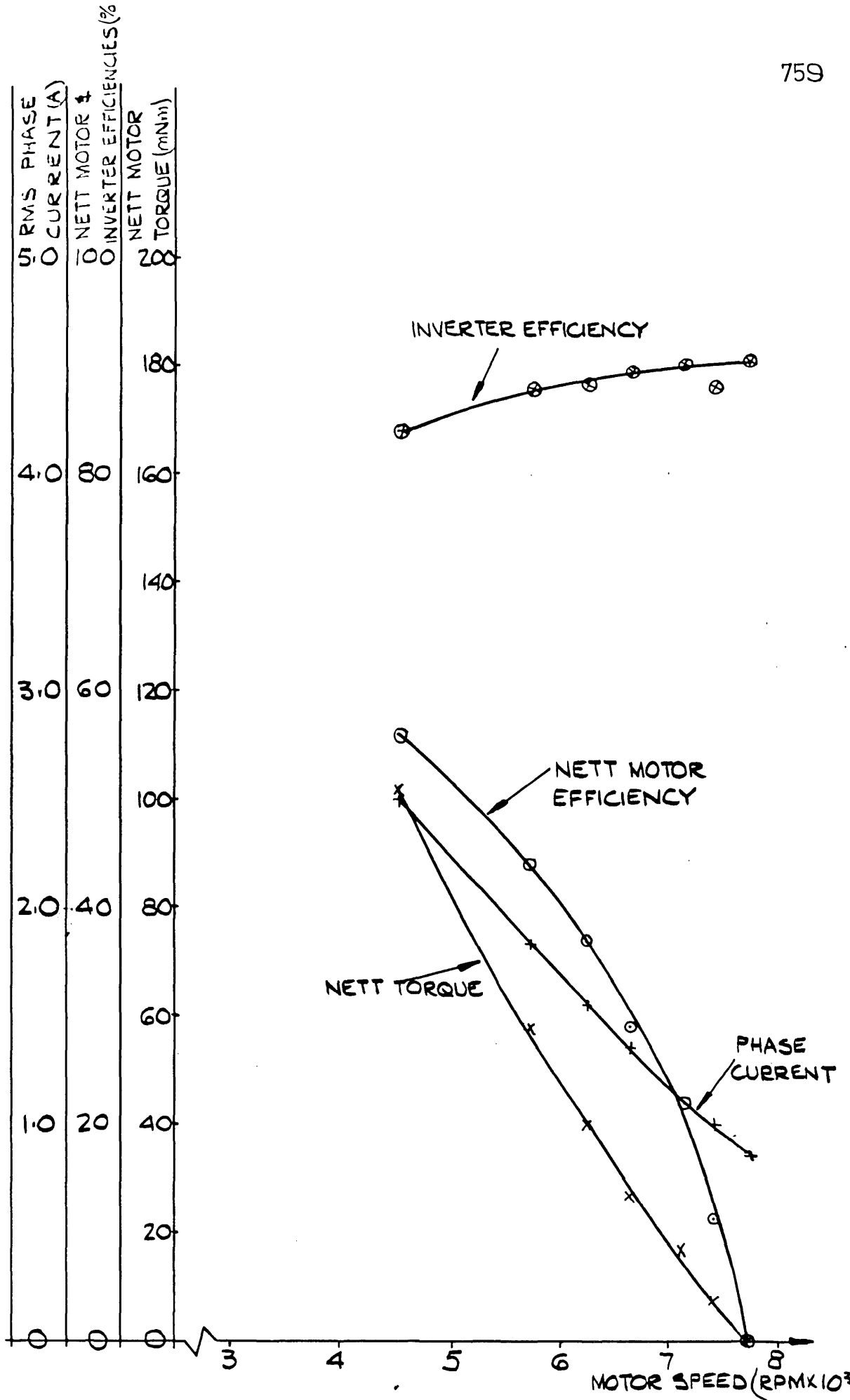


FIGURE 9.25 7 PHASE MOTOR: 6° LOAD ANGLE; ±10 VOLT SUPPLY TO INVERTER - NETT TORQUE, NETT MOTOR EFFICIENCY, PHASE CURRENT AND INVERTER EFFICIENCY

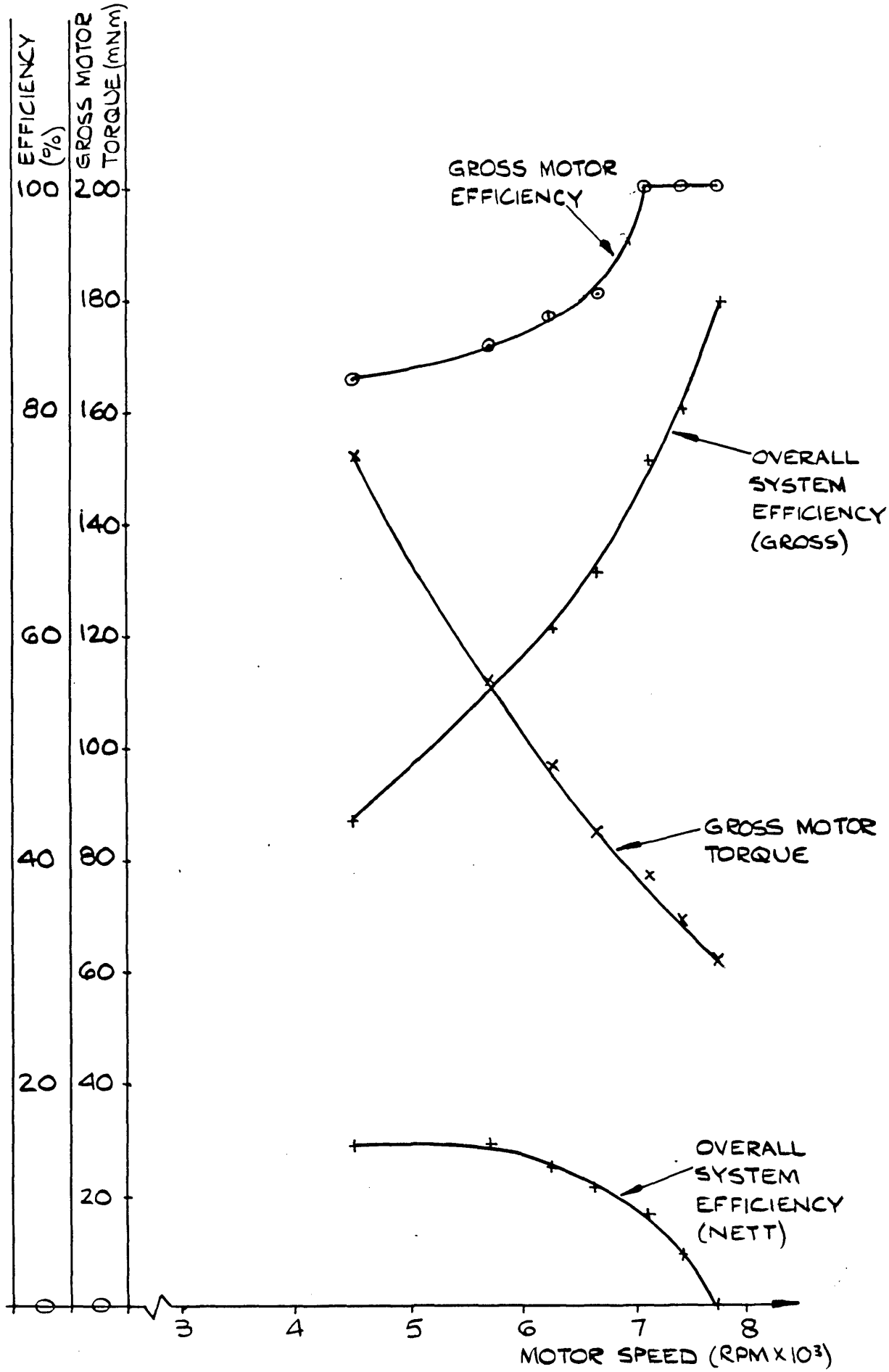


FIGURE 9.26 7 PHASE MOTOR: 6° LOAD ANGLE; ± 10 VOLT SUPPLY TO INVERTER - GROSS MOTOR TORQUE, GROSS MOTOR EFFICIENCY, OVERALL SYSTEM EFFICIENCY (GROSS) AND OVERALL SYSTEM EFFICIENCY (NETT)

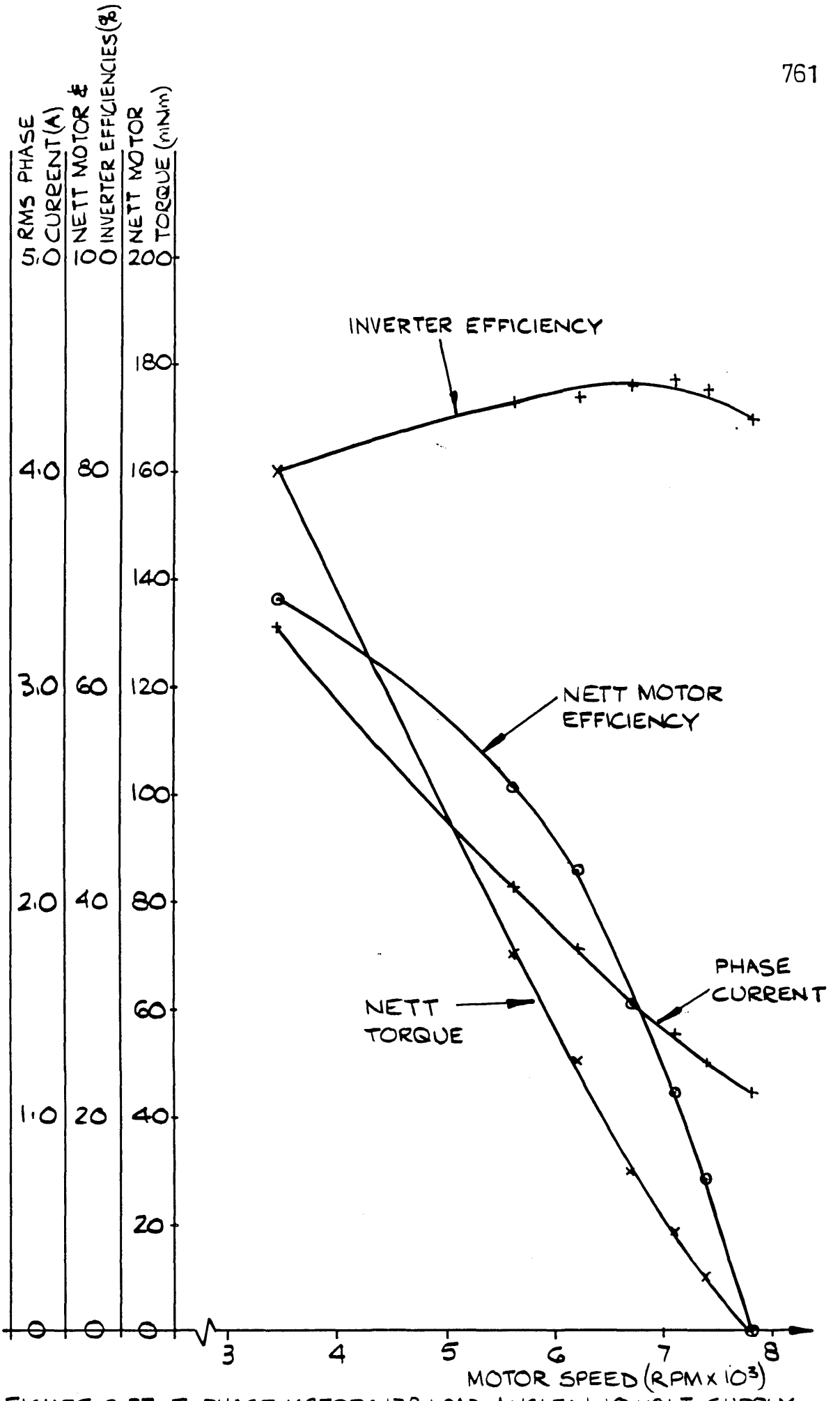


FIGURE 9.27 7 PHASE MOTOR: 13° LOAD ANGLE; ± 10 VOLT SUPPLY TO INVERTER - NETT TORQUE, NETT MOTOR EFFICIENCY, PHASE CURRENT AND INVERTER EFFICIENCY

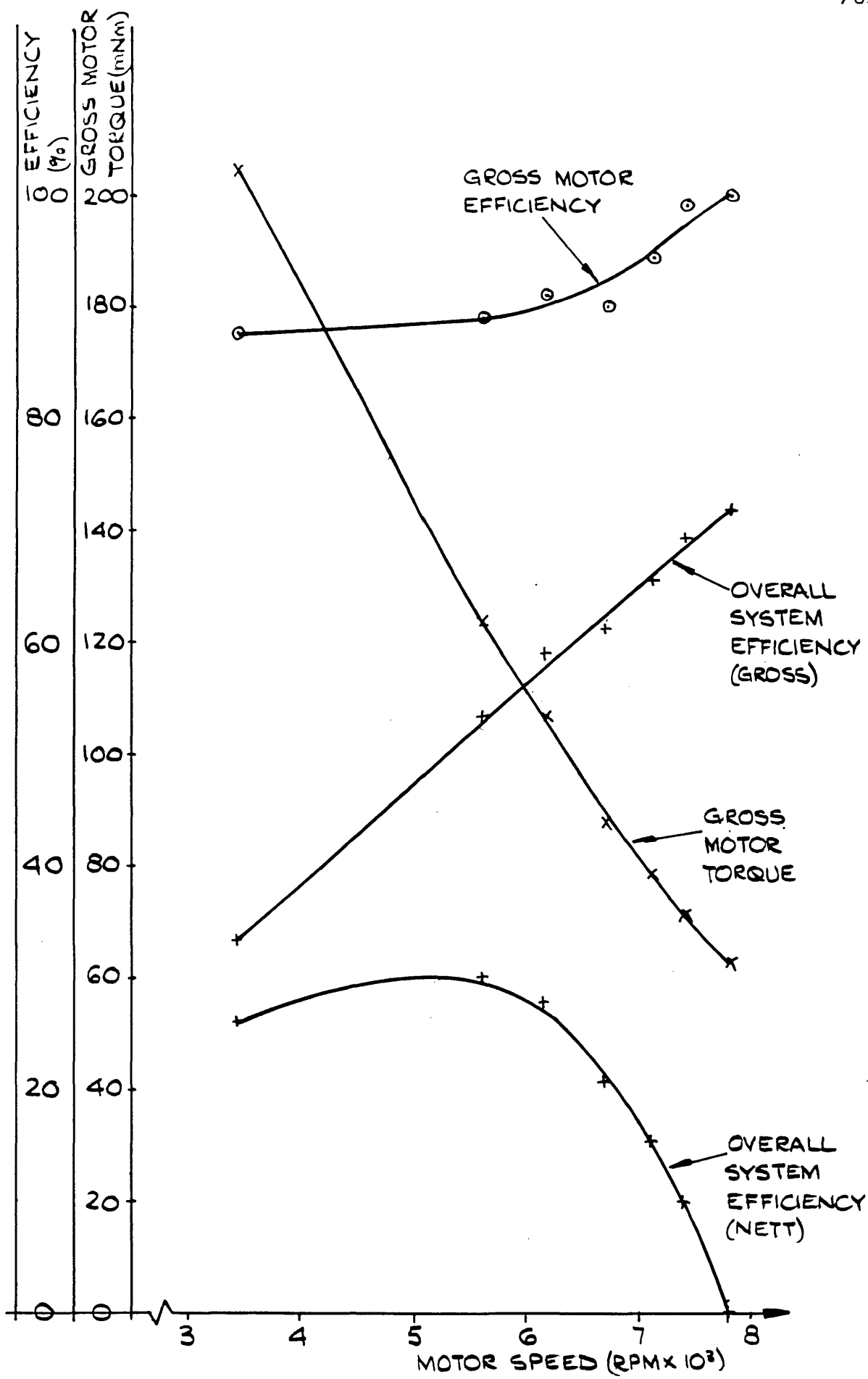


FIGURE 5.28 7 PHASE MOTOR: 13° LOAD ANGLE; ± 10 VOLT SUPPLY TO INVERTER - GROSS MOTOR TORQUE, GROSS MOTOR EFFICIENCY, OVERALL SYSTEM EFFICIENCY (GROSS) AND OVERALL SYSTEM EFFICIENCY (NETT)

efficiency it is reasonable to regard the inverter efficiency as extremely good. Further comments on the inverter efficiency are made in section 9.4.

Thirdly, the gross motor efficiency exceeds 100% by small amounts at some points. Whenever this has occurred the efficiency has been plotted as 100%. This "excess" efficiency is mainly due to errors in the loss-torque versus speed characteristics presented in figs. 9.13 and 9.14. The loss-torque seems to be slightly too large over the entire speed range, and it is thought that this is most probably due to the calculated value of the moment of inertia for the rotor being slightly too large. Certain approximations were made in the calculation of the moment of inertia and so the result may well be too big. The efficiency errors may also be due to some extent to the fact that the power input to the motor was calculated by multiplying the power input to phase 1 by a factor of 7. Hence any small variations in the power consumption of the phases are not accounted for and this could lead to some small errors in the calculated nett and gross motor efficiencies.

It is appropriate at this point to define what is meant by the terms nett (or gross) motor efficiency and nett (or gross) system efficiency. The motor efficiency refers to the energy conversion efficiency of the motor: the nett efficiency is calculated from the nett power available at the shaft divided by the total input power to the motor; the gross efficiency is calculated from the gross power available at the shaft (i.e. nett power plus I.W.F. loss power) divided by the total power input to the motor. The system efficiencies are likewise calculated using the nett and gross shaft powers but they are divided by the total power input to the inverter. Thus the system efficiencies include the effects of losses in the inverter, the protection components, and the motor.

Further performance characteristics and waveform

photographs obtained from the motor for inverter supply voltages of ±14, ±18, ±19, ±21, and ±22 volts are presented in figs. 9.29 to 9.59 inclusive. The characteristics and photographs are arranged as follows:

±14 volt results - figs. 9.29 to 9.38 inclusive;
±18 volt results - figs. 9.39 to 9.48 inclusive;
±19 volt results - figs. 9.49 to 9.51 inclusive;
±21 volt results - figs. 9.52 to 9.54 inclusive;
±22 volt results - figs. 9.55 to 9.59 inclusive.

The reasonably comprehensive selection of characteristics and waveforms over the operating voltage range have been included because of the novelty of the square-wave system. However, it is by no means necessary to study all the characteristics to appreciate the basic operating characteristics of the motor.

The results shown in figs. 9.29 to 9.59 inclusive are of the same form as the ±10 volt characteristics and waveforms shown in figs. 9.15 to 9.28 inclusive. It is not thought necessary to comment in detail on the results since the characteristics are smooth and consistent. However, it should be noted that in all cases the maximum torques shown by the characteristics are not the maximum available motor torques. The inadequate design of the eddy current brake prevented full torque/speed curves from being obtained. The differences between the nett and gross motor efficiencies show that the main source of loss in the motor/inverter system is the I.W.F. loss torque. It would be necessary to reduce the I.W.F. loss torque substantially in order to obtain a reasonable nett motor efficiency.

The relationship between the 0° load angle nett torque versus speed characteristics for inverter supply voltages of ±10, ±14, ±19, and ±22 volts is shown in fig. 9.60. Fig. 9.61 shows the relationship between the gross torque versus speed characteristics for a 0° load angle and the

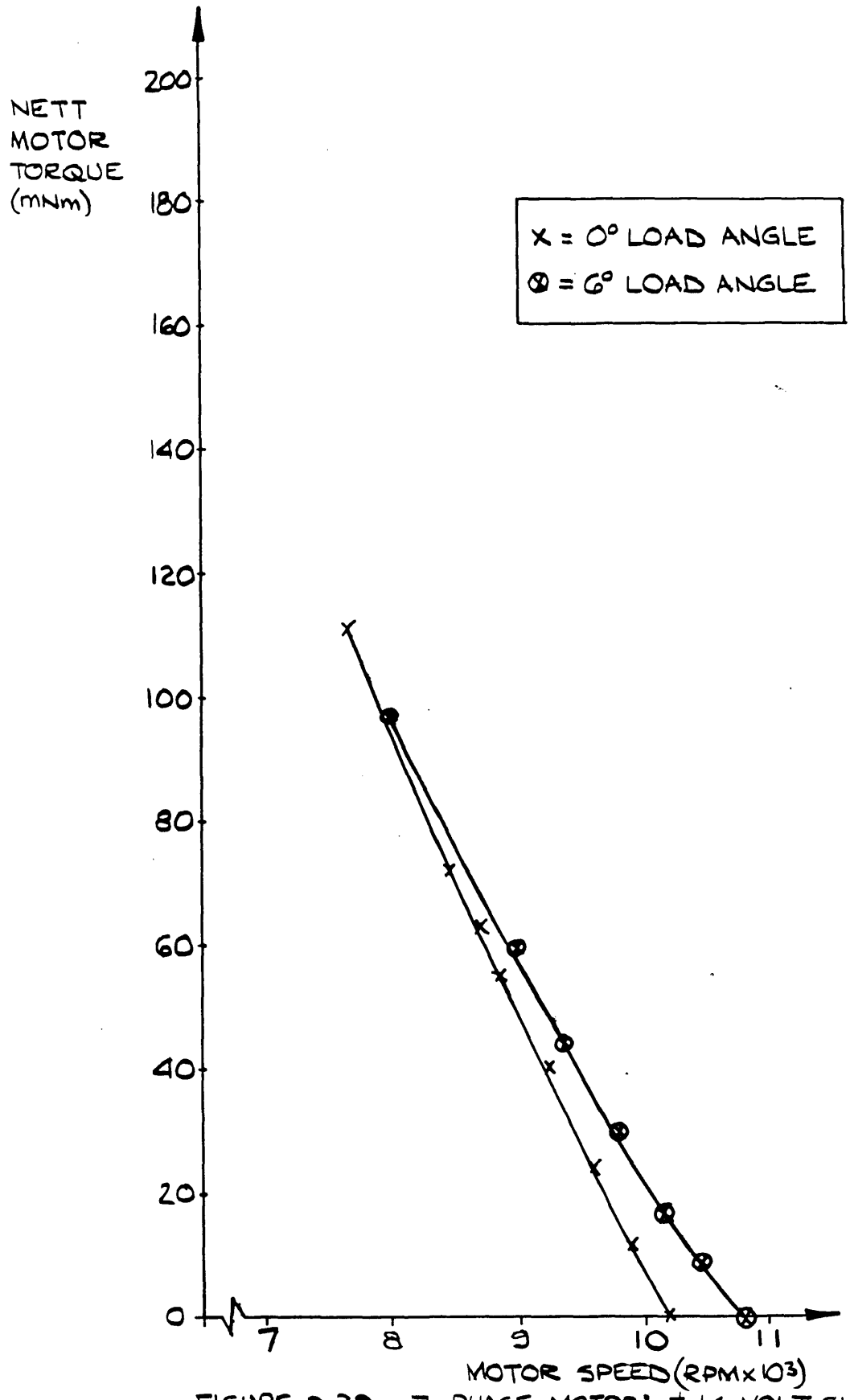


FIGURE 9.23. 7 PHASE MOTOR: ± 14 VOLT SUPPLY TO INVERTER - NETT TORQUE FOR LOAD ANGLES OF 0° AND 6°

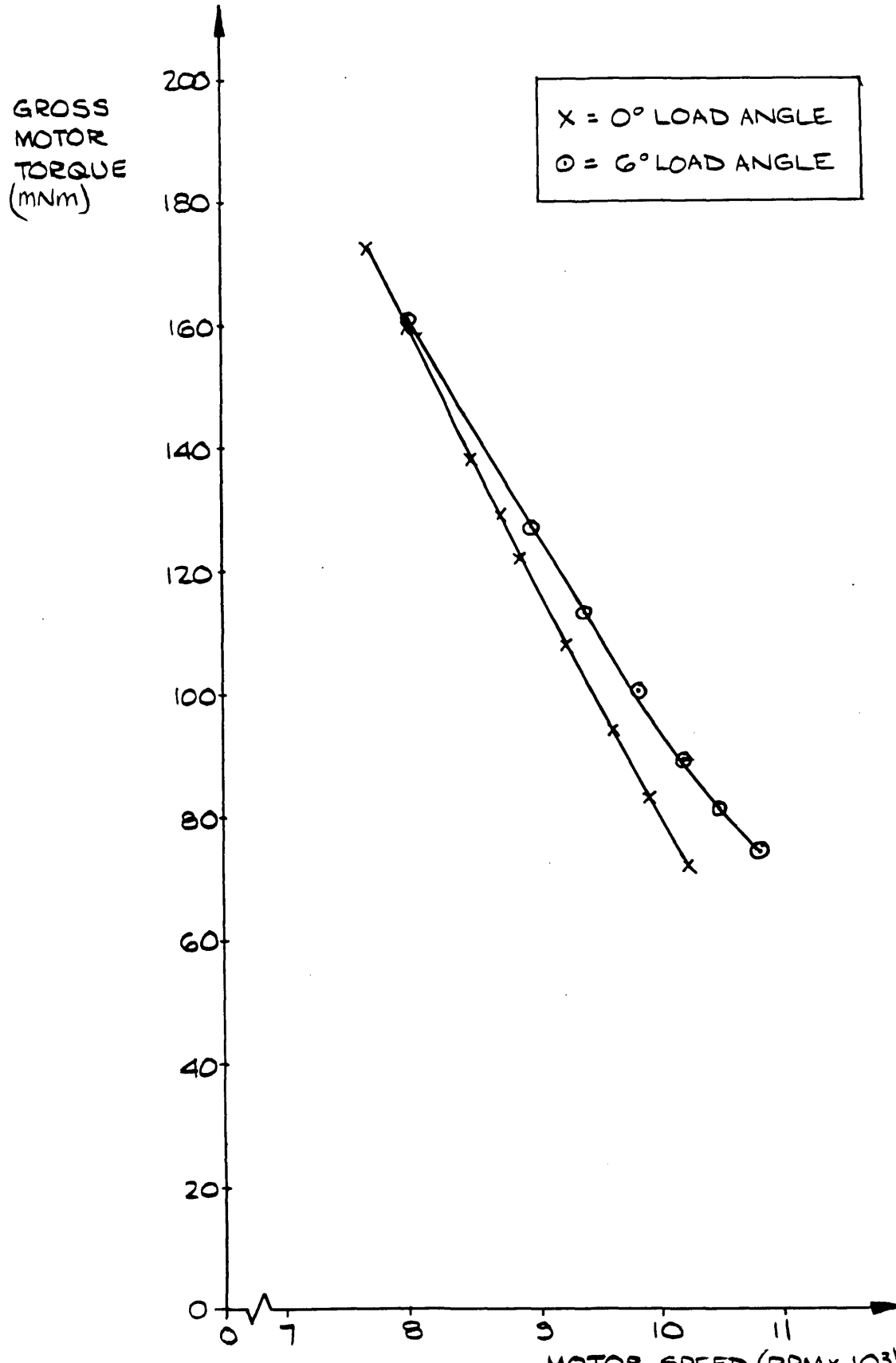


FIGURE 9.30. 7 PHASE MOTOR: ±14 VOLT SUPPLY TO INVERTER - GROSS TORQUE FOR LOAD ANGLES OF 0° AND 6°

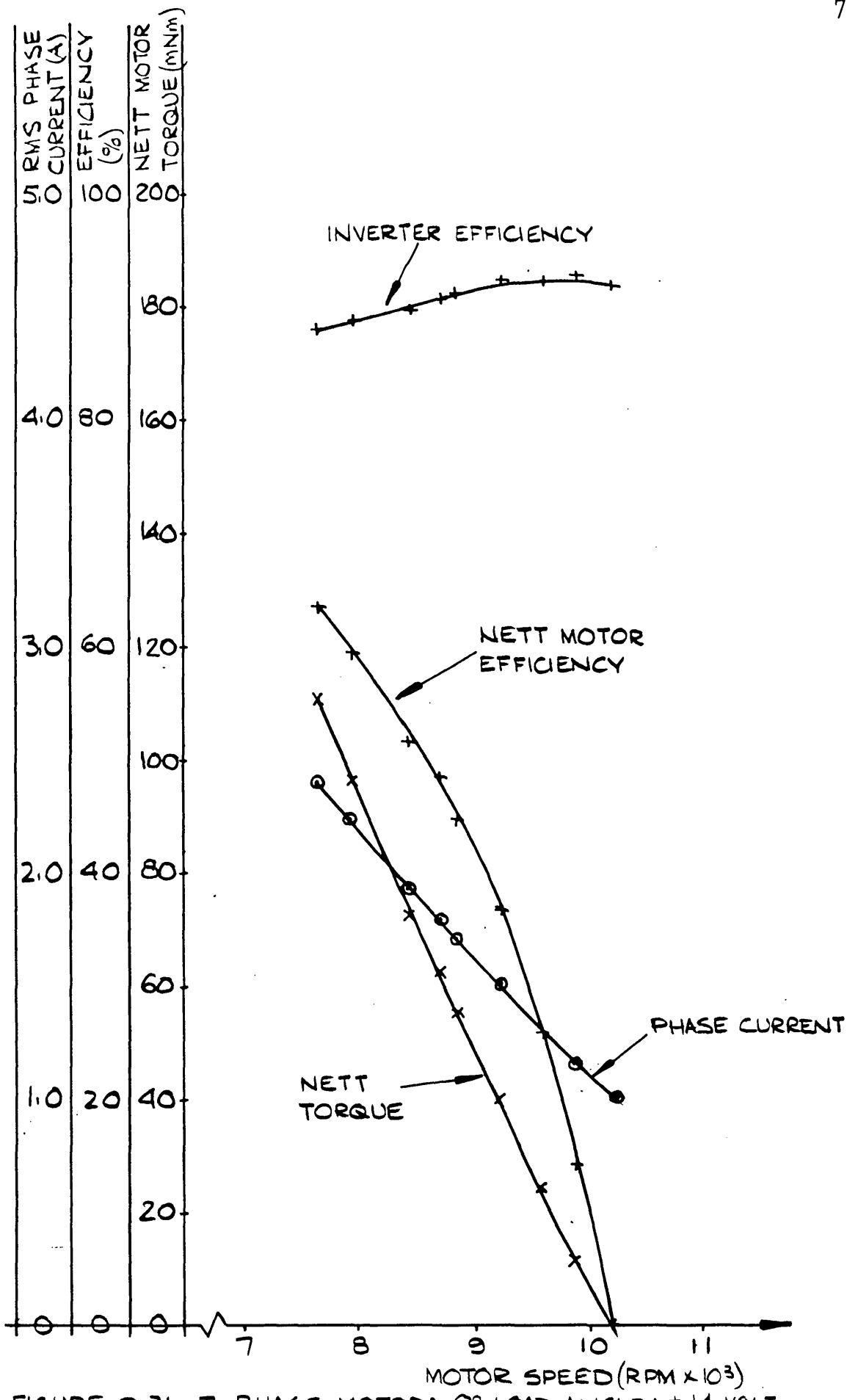


FIGURE 9.31. 7 PHASE MOTOR: 0° LOAD ANGLE; ±14 VOLT SUPPLY TO INVERTER - NETT TORQUE, NETT MOTOR EFFICIENCY, PHASE CURRENT, AND INVERTER EFFICIENCY

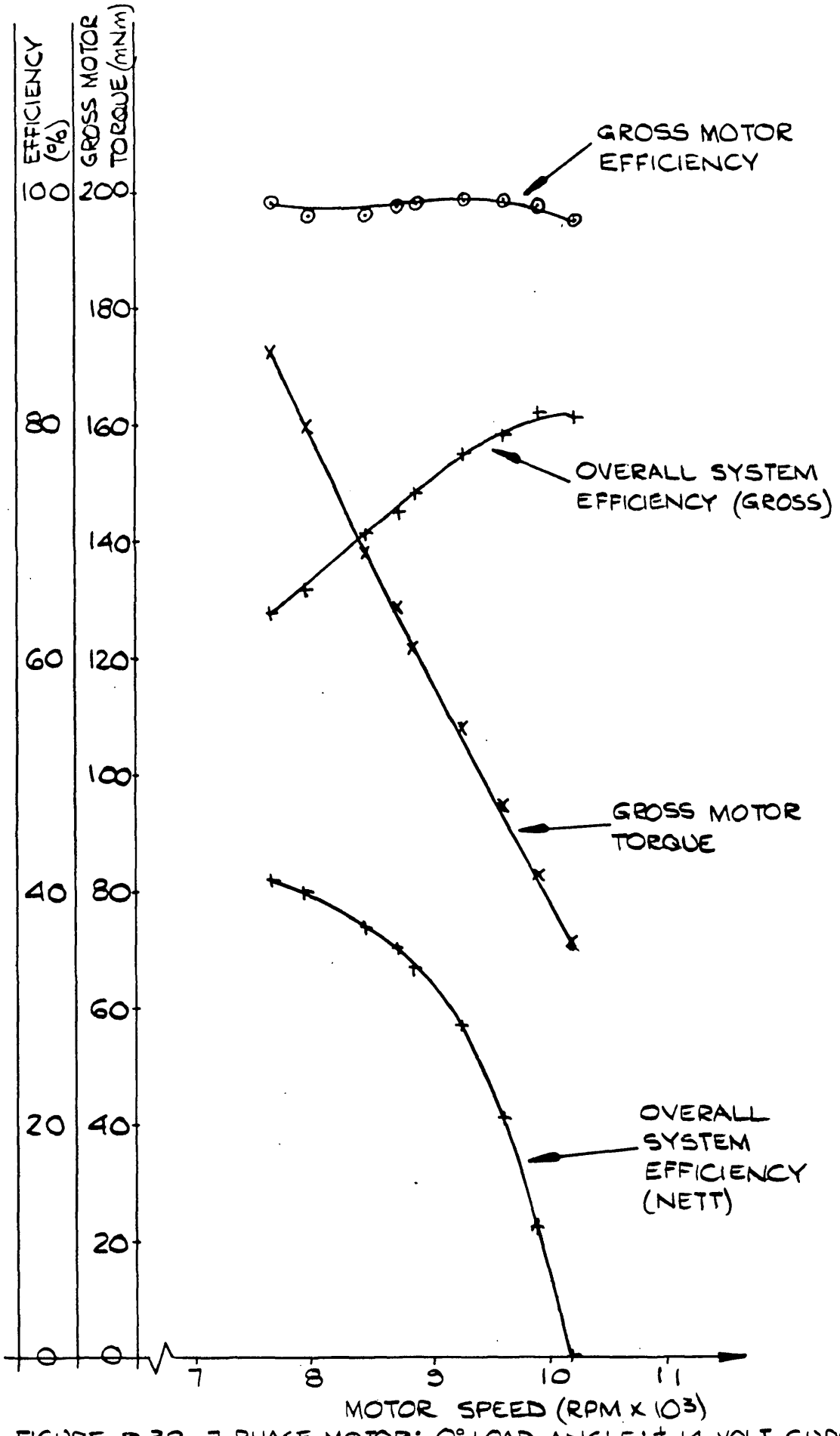


FIGURE 9.32. 7 PHASE MOTOR: 0° LOAD ANGLE; ± 14 VOLT SUPPLY TO INVERTER - GROSS MOTOR TORQUE, GROSS MOTOR EFFICIENCY, OVERALL SYSTEM EFFICIENCY (GROSS), AND OVERALL SYSTEM EFFICIENCY (NETT)

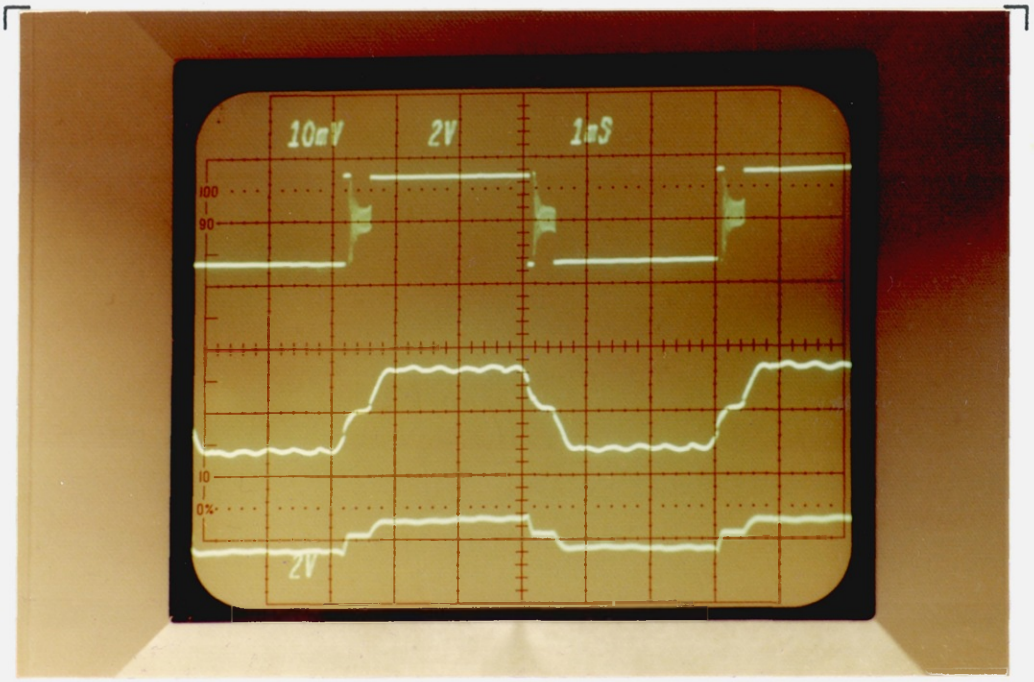


FIG. 9.33. PHASE ONE WAVEFORMS : 0° LOAD ANGLE ; ±14 VOLT INVERTER SUPPLY ; NO LOAD (ZERO NETT TORQUE), 10180 RPM — TOP : INVERTER OUTPUT VOLTAGE (20V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (20V/cm) — BOTTOM : PHASE CURRENT (5A/cm) — 1ms/cm.

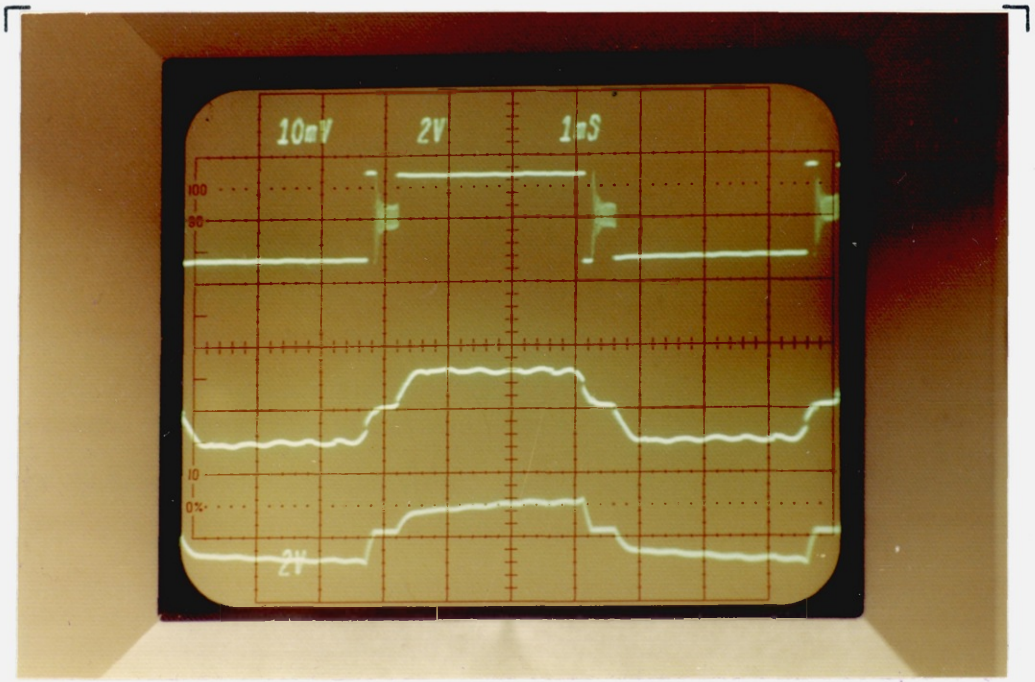


FIG. 9.34. PHASE ONE WAVEFORMS : 0° LOAD ANGLE ; ±14 VOLT INVERTER SUPPLY ; 63mNm NETT TORQUE, 8700 RPM — TOP : INVERTER OUTPUT VOLTAGE (20V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (20V/cm) — BOTTOM : PHASE CURRENT (5A/cm) — 1ms/cm.

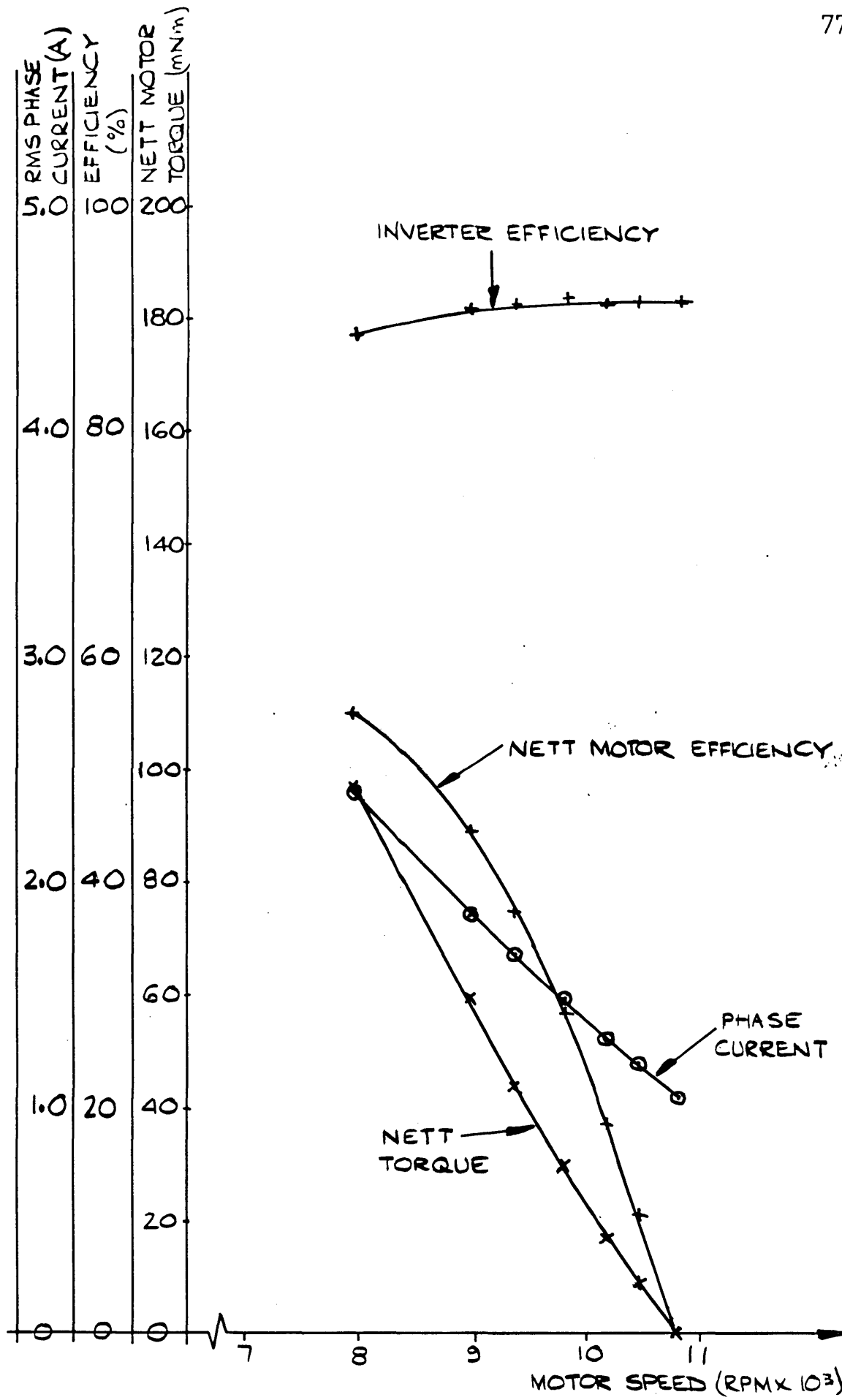


FIGURE 5.35 7 PHASE MOTOR: 6° LOAD ANGLE; ± 14 VOLT SUPPLY TO INVERTER - NETT TORQUE, NETT MOTOR EFFICIENCY, PHASE CURRENT, AND INVERTER EFFICIENCY

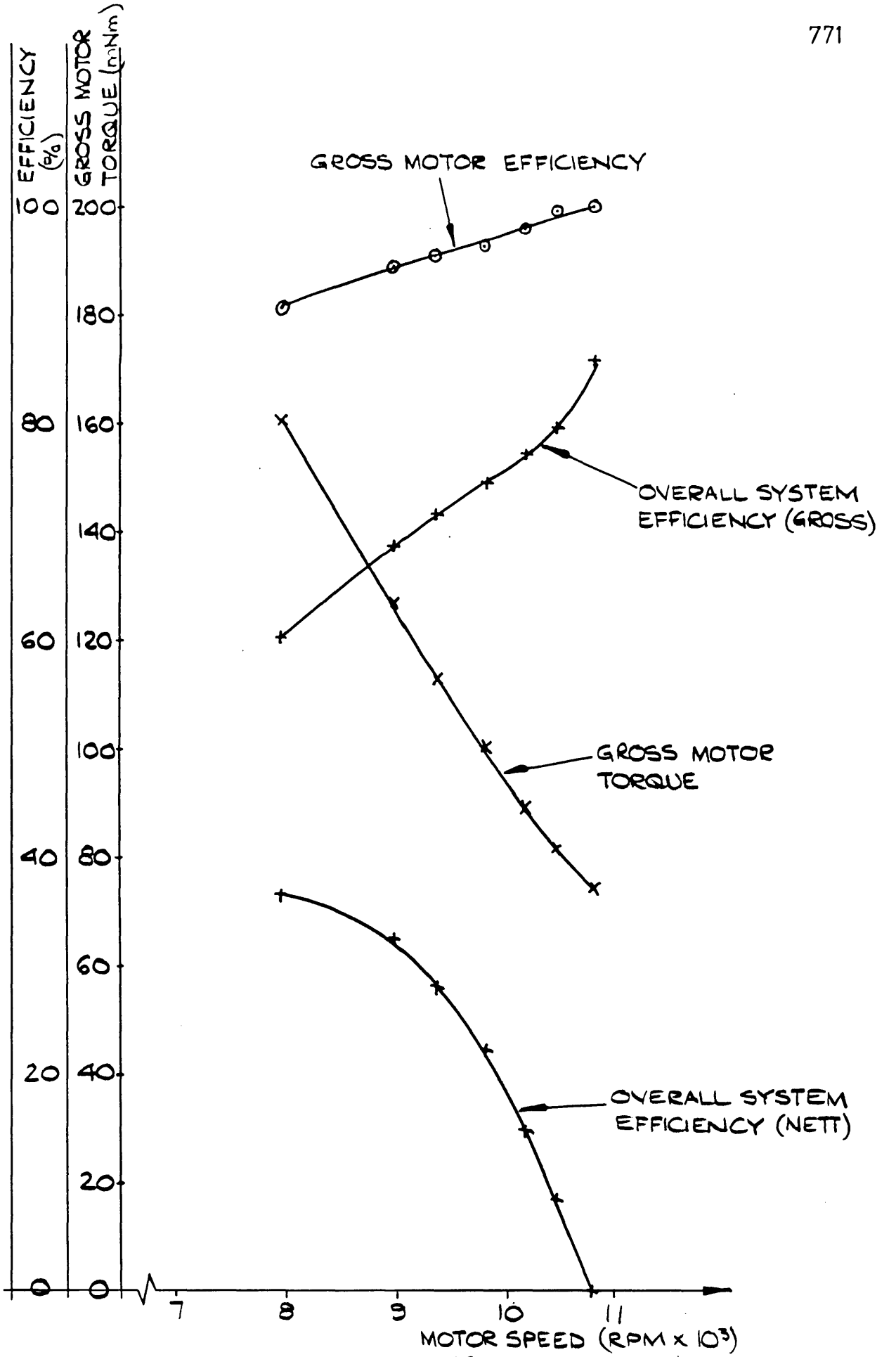


FIGURE 9.36 7 PHASE MOTOR: 6° LOAD ANGLE; ± 14 VOLT SUPPLY TO INVERTER - GROSS MOTOR TORQUE, GROSS MOTOR EFFICIENCY, OVERALL SYSTEM EFFICIENCY (GROSS) AND OVERALL SYSTEM EFFICIENCY (NETT)

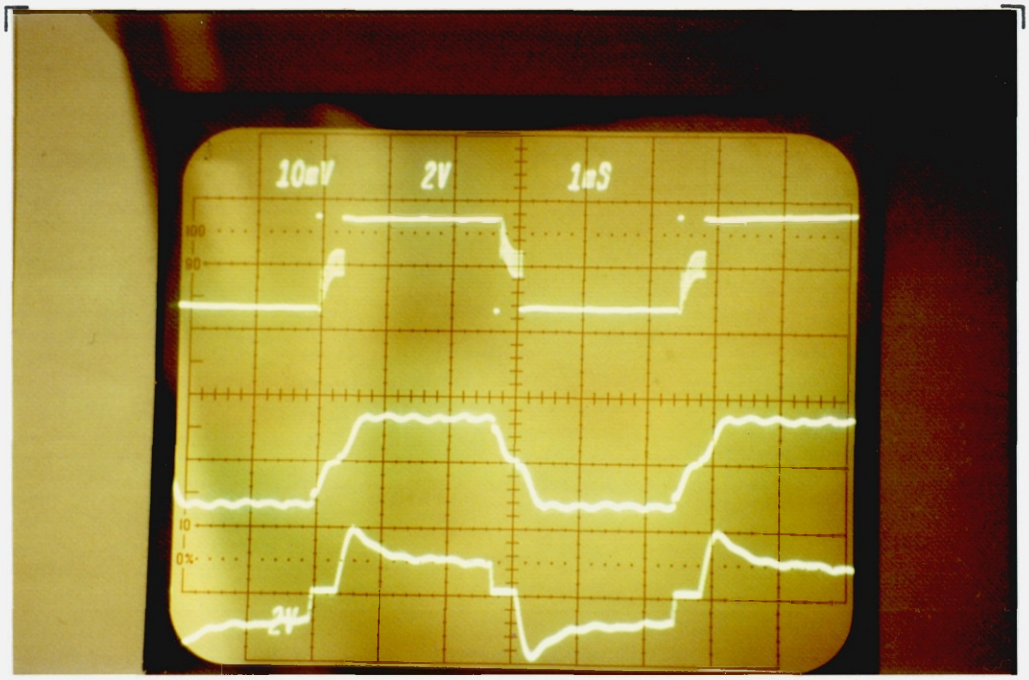


FIG. 9.37. PHASE ONE WAVEFORMS: 6° LOAD ANGLE; ±14 VOLT INVERTER SUPPLY; NO LOAD (ZERO NETT TORQUE), 10800 RPM — TOP: INVERTER OUTPUT VOLTAGE (20V/cm) — MIDDLE: MOTOR PHASE VOLTAGE (20V/cm) — BOTTOM: PHASE CURRENT (2A/cm) — 1ms/cm.

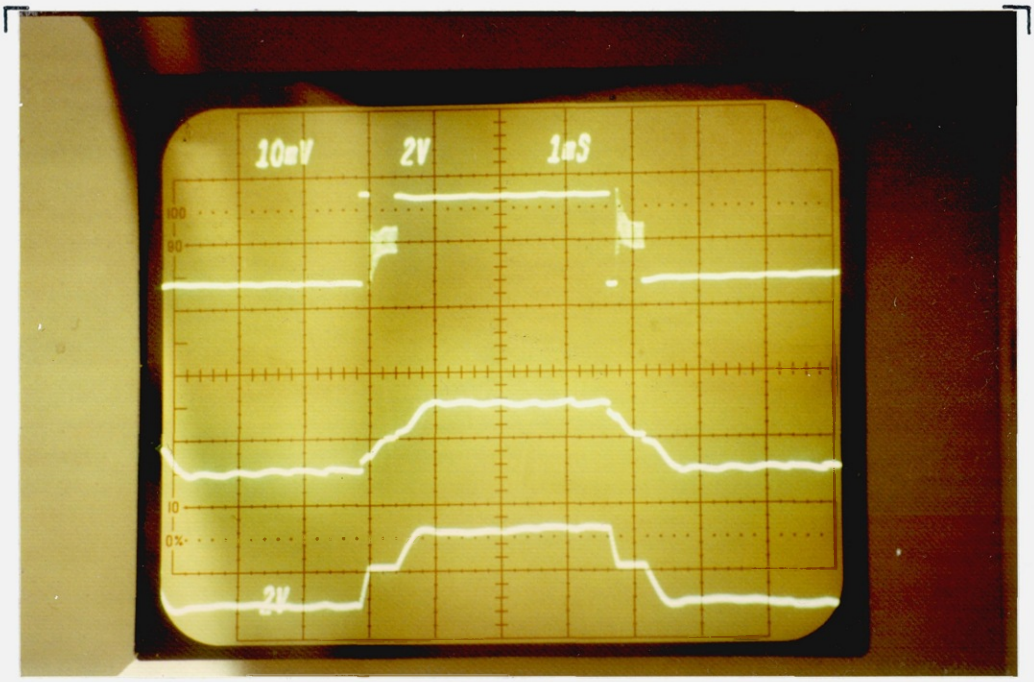


FIG. 9.38. PHASE ONE WAVEFORMS: 6° LOAD ANGLE; ±14 VOLT INVERTER SUPPLY; 97 mNm NETT TORQUE, 7950 RPM — TOP: INVERTER OUTPUT VOLTAGE (20V/cm) — MIDDLE: MOTOR PHASE VOLTAGE (20V/cm) — BOTTOM: PHASE CURRENT (5A/cm) — 1ms/cm.

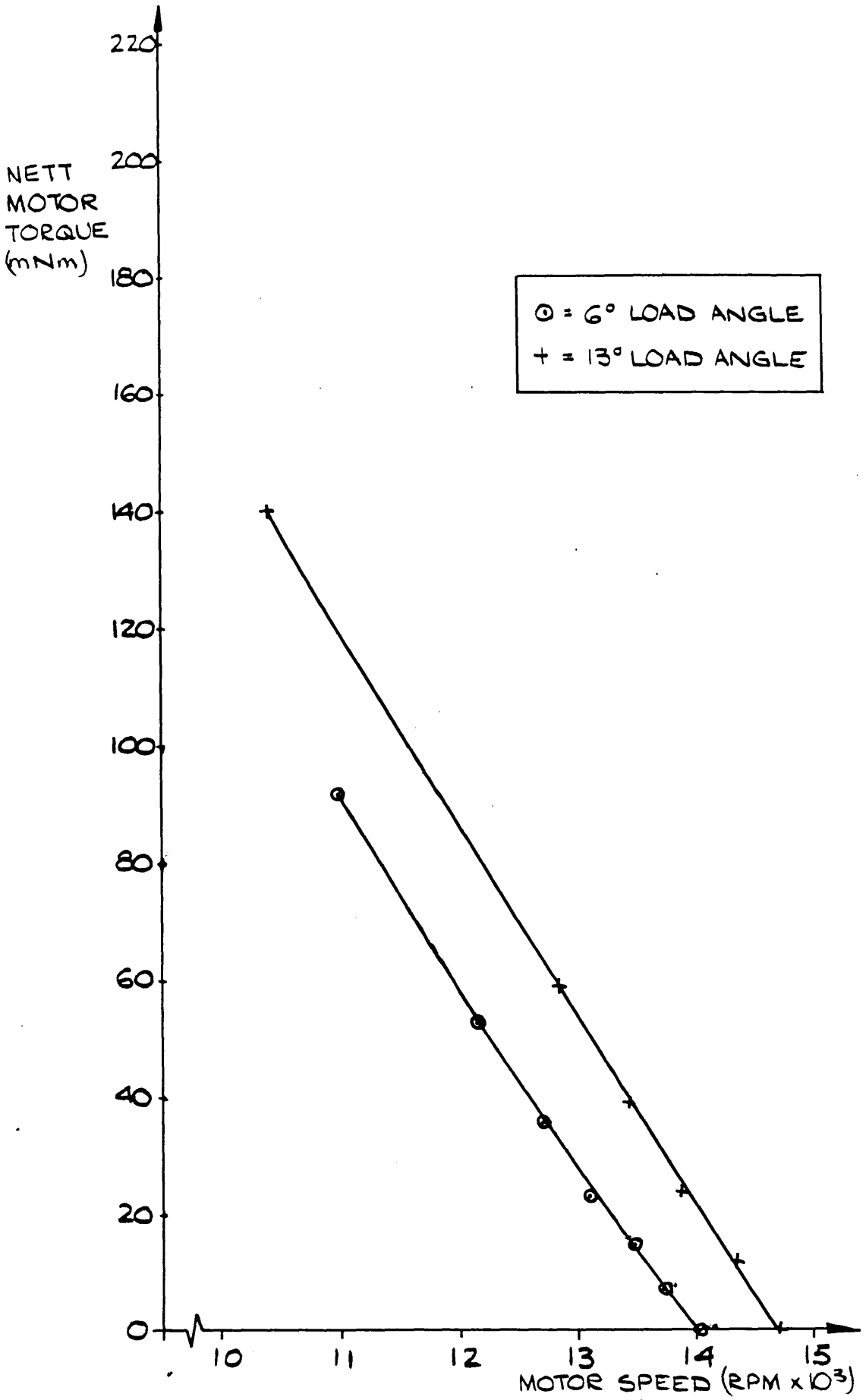


FIGURE 9.39 7 PHASE MOTOR: ± 18 VOLT SUPPLY TO INVERTER - NETT TORQUE FOR LOAD ANGLES OF 6° AND 13°

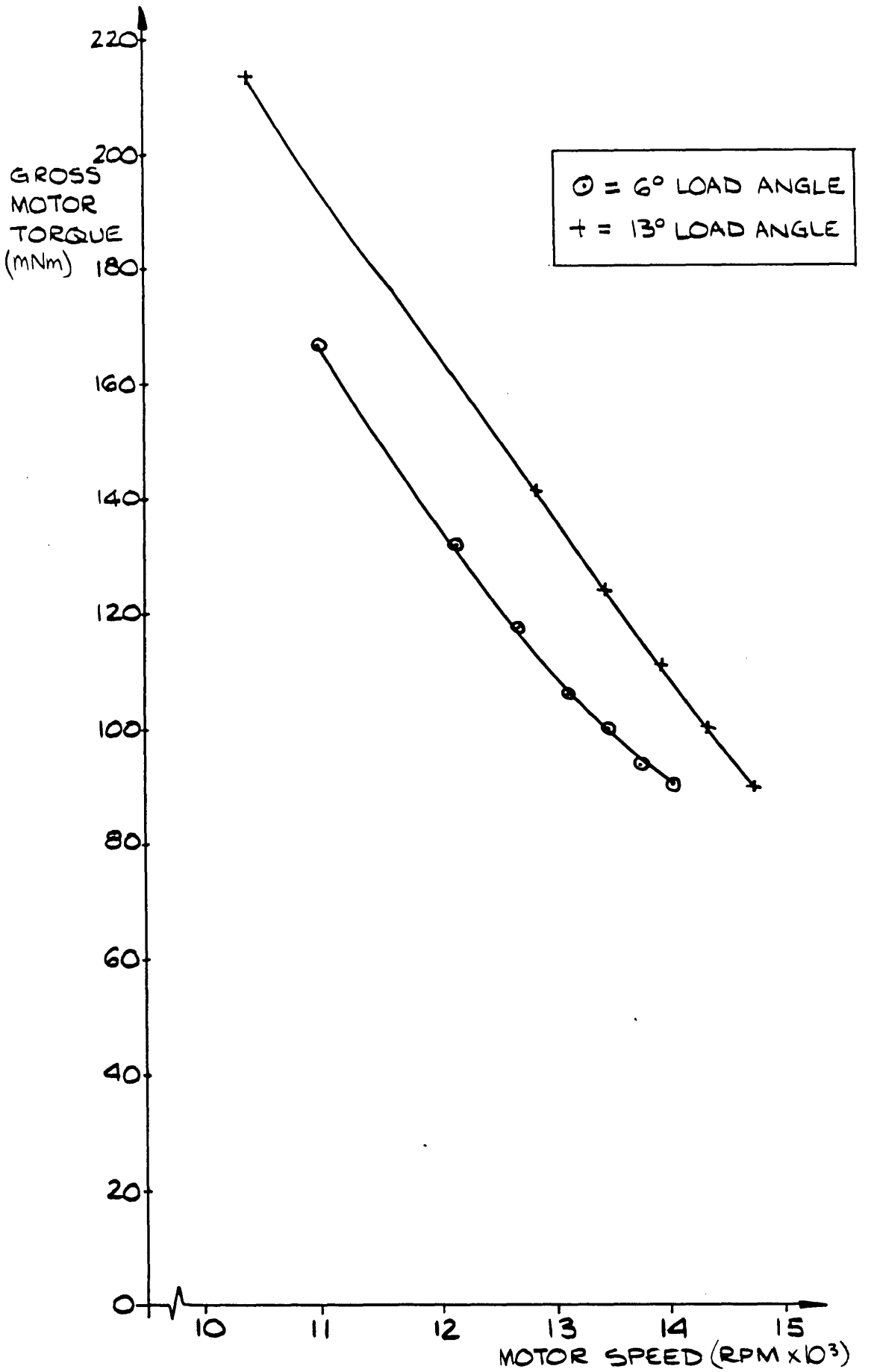


FIGURE 9.40 7 PHASE MOTOR: ± 18 VOLT SUPPLY TO INVERTER - GROSS TORQUE FOR LOAD ANGLES OF 6° AND 13°

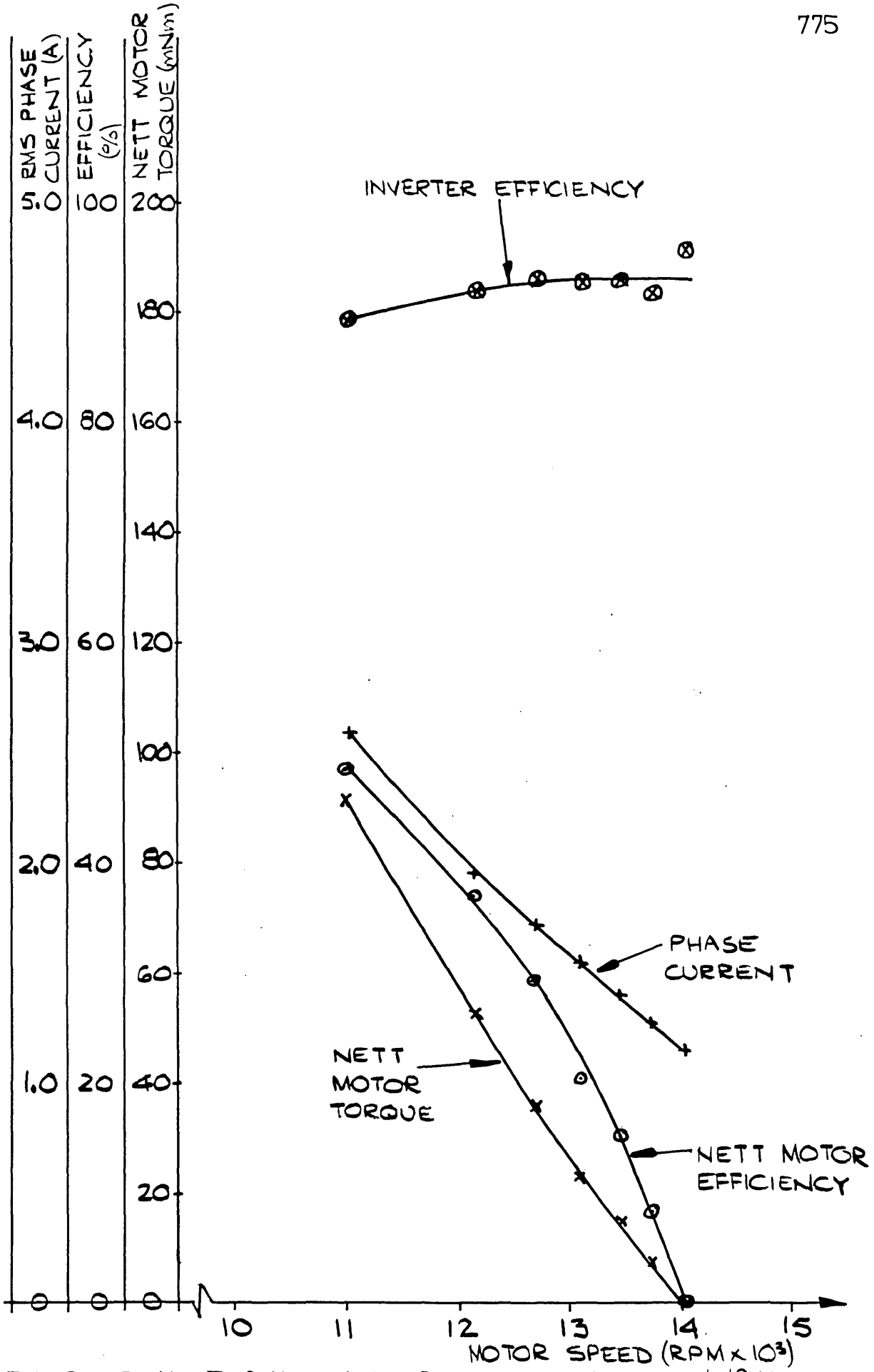


FIGURE 9.41 7 PHASE MOTOR: 6° LOAD ANGLE; ± 18 VOLTS SUPPLIED TO INVERTER - NETT TORQUE, NETT MOTOR EFFICIENCY, PHASE CURRENT AND INVERTER EFFICIENCY

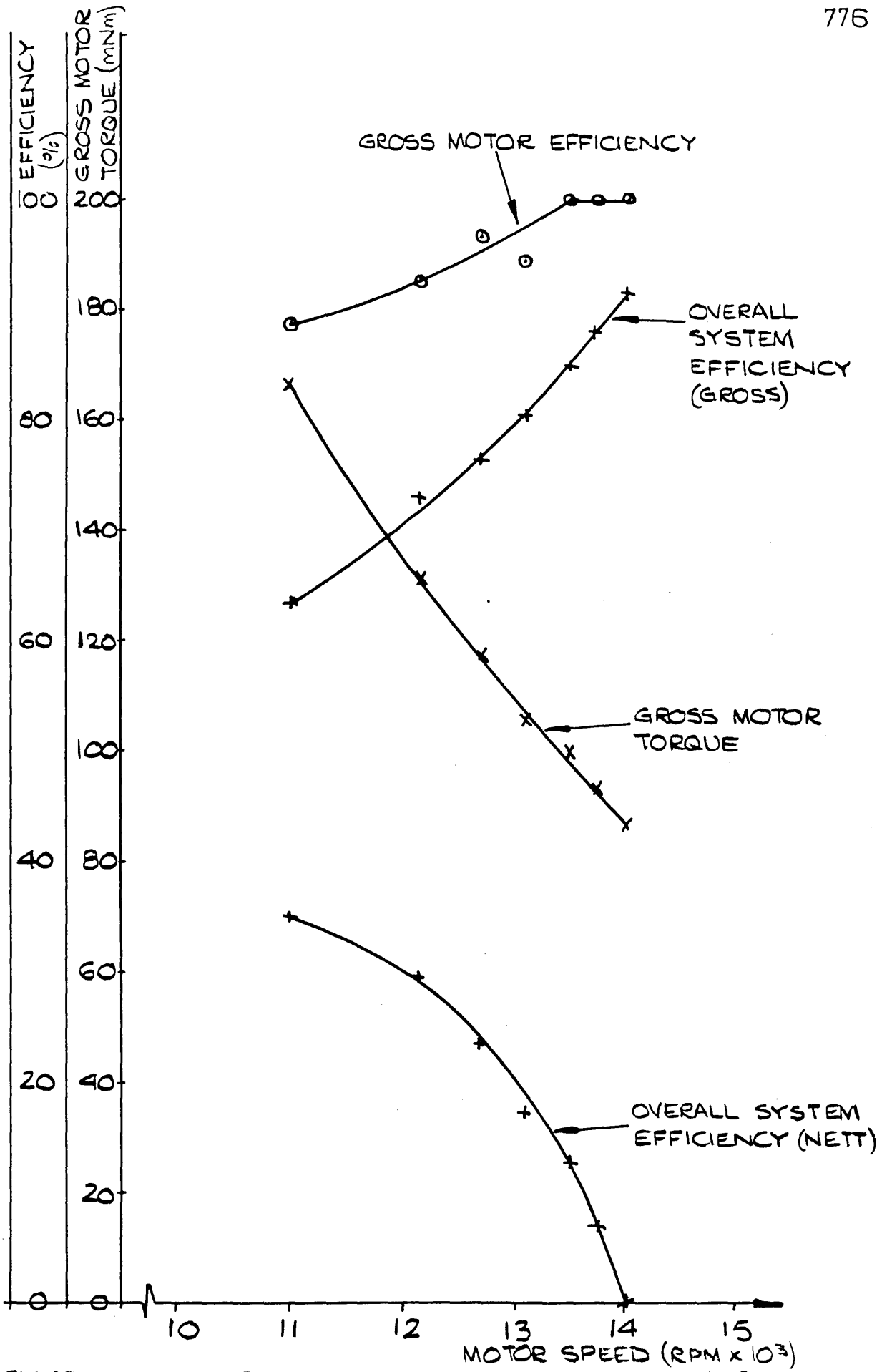


FIGURE 5.42 7 PHASE MOTOR: 6° LOAD ANGLE; ± 18 VOLT SUPPLY TO INVERTER - GROSS MOTOR TORQUE, GROSS MOTOR EFFICIENCY, OVERALL SYSTEM EFFICIENCY (GROSS) AND OVERALL SYSTEM EFFICIENCY (NETT)

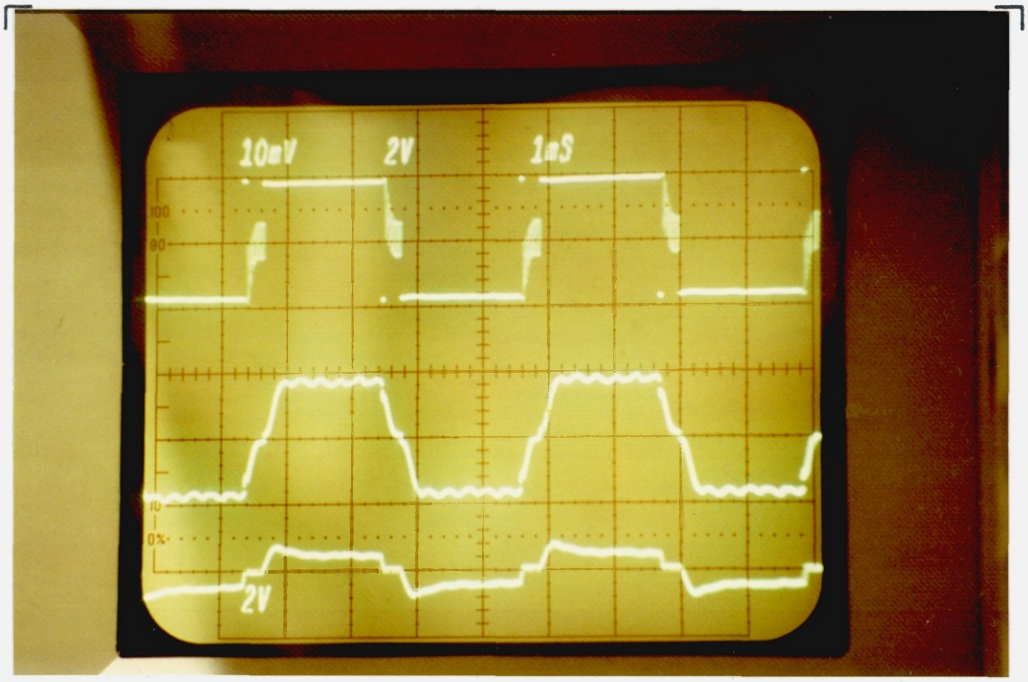


FIG. 9.43. PHASE ONE WAVEFORMS : 6° LOAD ANGLE ; ±18 VOLT INVERTER SUPPLY ; NO LOAD (ZERO NETT TORQUE), 14025 RPM — TOP : INVERTER OUTPUT VOLTAGE (20V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (20V/cm) — BOTTOM : PHASE CURRENT (5A/cm) — 1ms/cm.

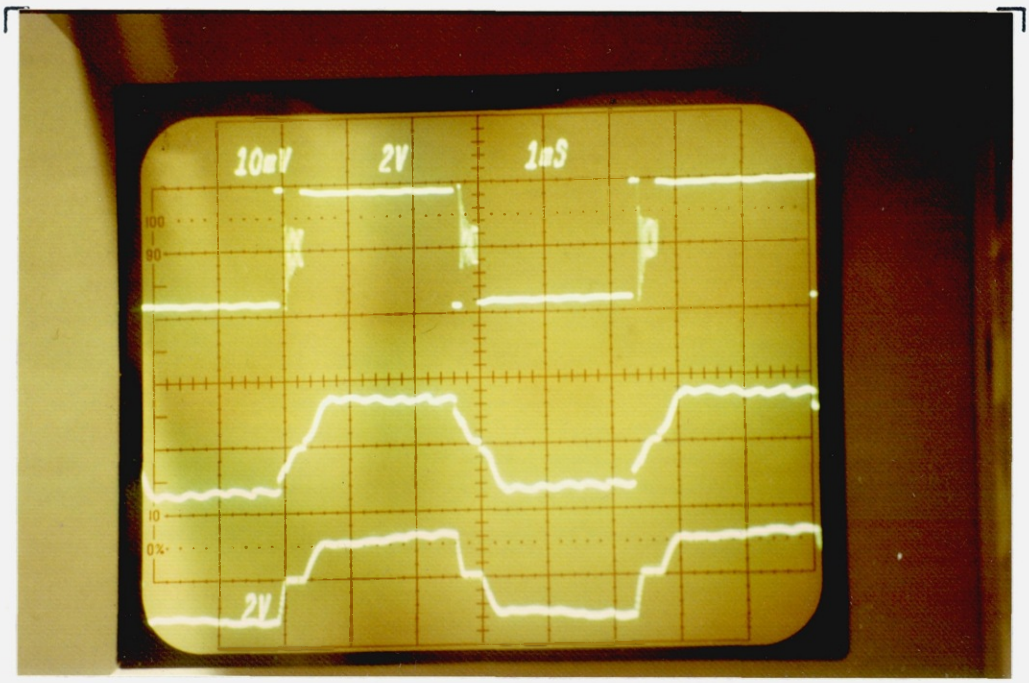


FIG. 9.44. PHASE ONE WAVEFORMS : 6° LOAD ANGLE ; ±18 VOLT INVERTER SUPPLY ; 92 mNm NETT TORQUE, 10980 RPM — TOP : INVERTER OUTPUT VOLTAGE (20V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (20V/cm) — BOTTOM : PHASE CURRENT (5A/cm) — 1ms/cm.

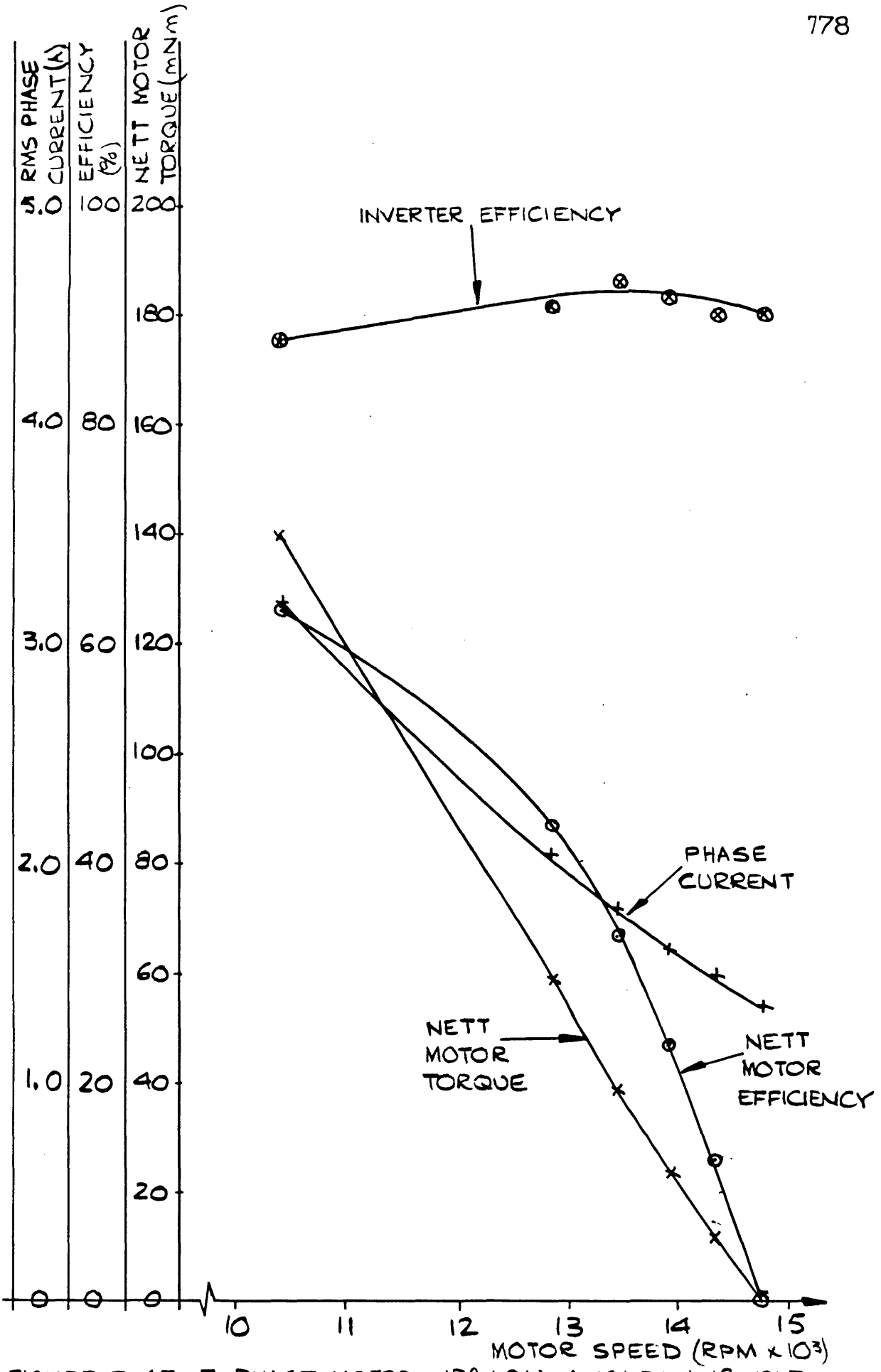


FIGURE 3.45 7 PHASE MOTOR: 13° LOAD ANGLE; ± 18 VOLT SUPPLY TO INVERTER - NETT TORQUE, NETT MOTOR EFFICIENCY, PHASE CURRENT AND INVERTER EFFICIENCY

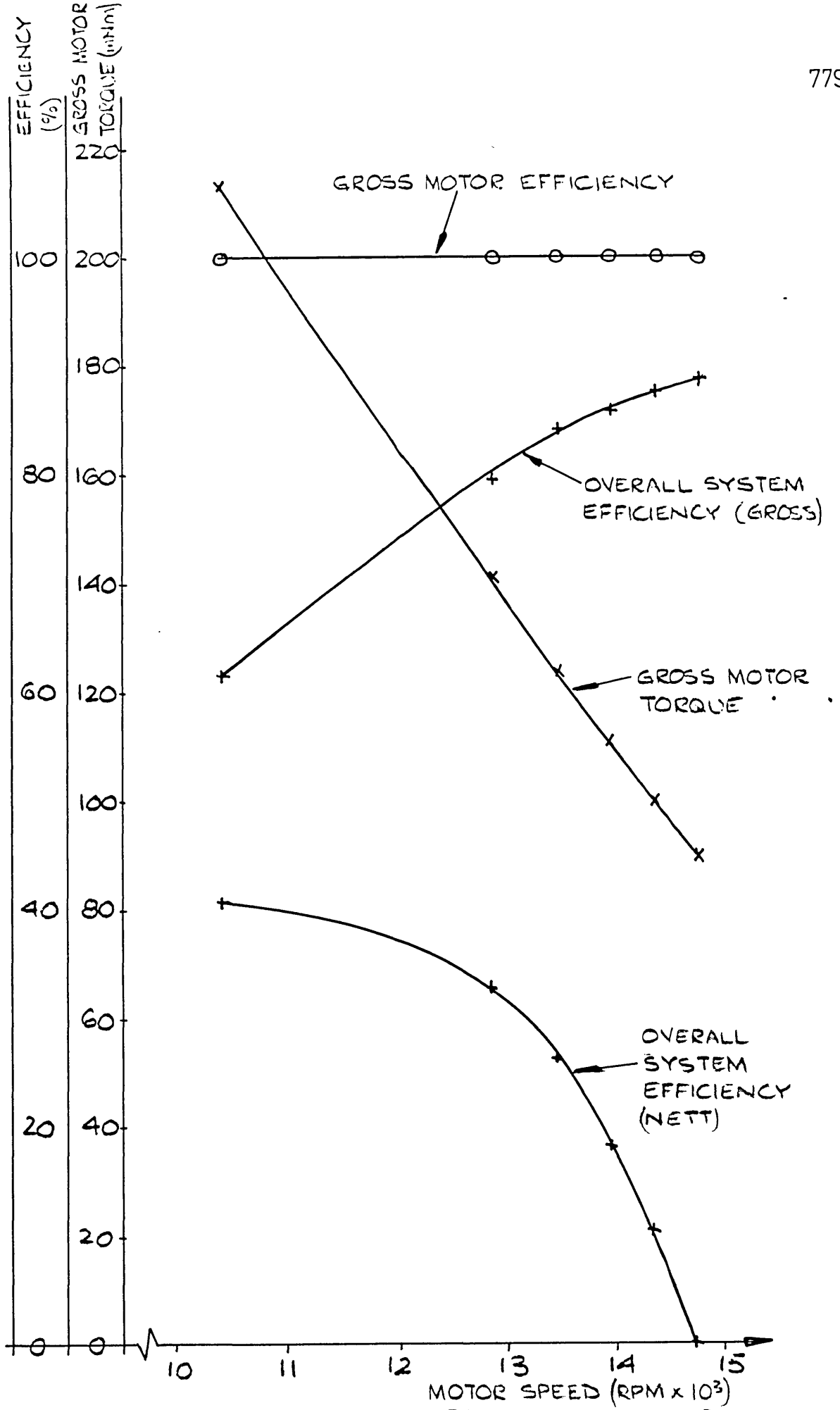


FIGURE 9.46 7 PHASE MOTOR: 13° LOAD ANGLE; ± 18 VOLT SUPPLY TO INVERTER - GROSS MOTOR TORQUE, GROSS MOTOR EFFICIENCY, OVERALL SYSTEM EFFICIENCY (GROSS) AND OVERALL SYSTEM EFFICIENCY (NETT)

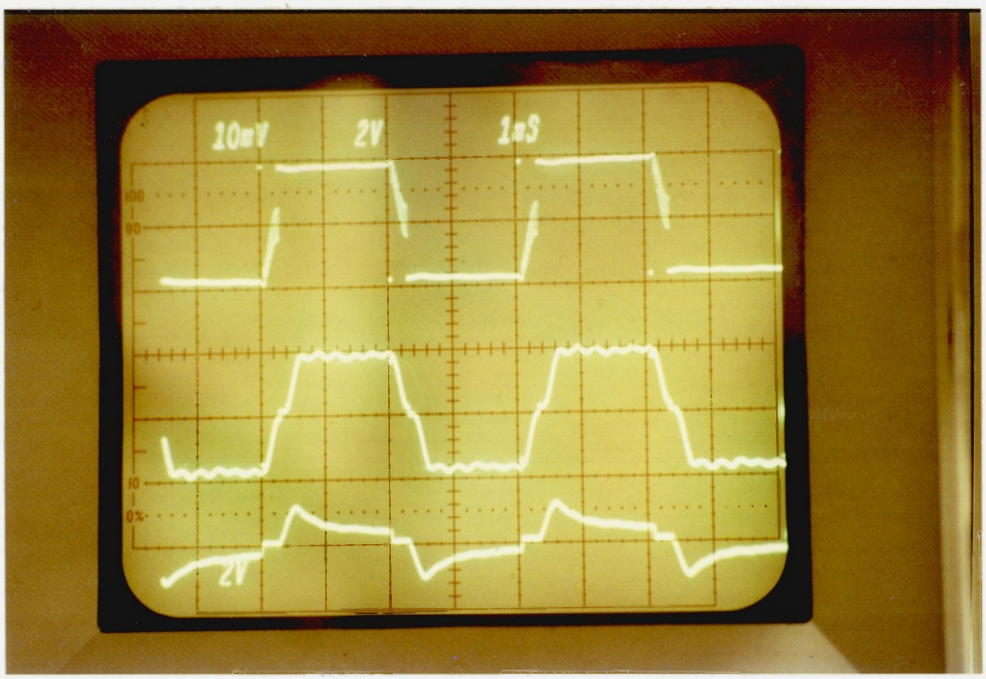


FIG. 9.47. PHASE ONE WAVEFORMS : 13° LOAD ANGLE ; ±18 VOLT INVERTER SUPPLY ; NO LOAD (ZERO NETT TORQUE), 14730 RPM — TOP : INVERTER OUTPUT VOLTAGE (20V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (20V/cm) — BOTTOM : PHASE CURRENT (5A/cm) — 1ms/cm.

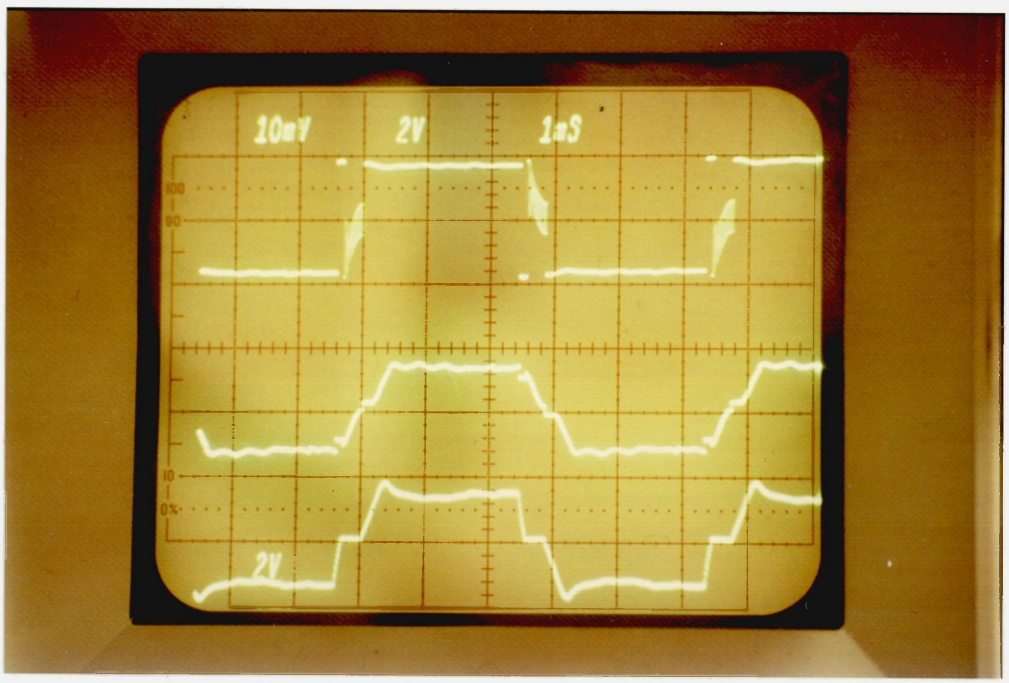


FIG. 9.48. PHASE ONE WAVEFORMS : 13° LOAD ANGLE ; ±18 VOLT INVERTER SUPPLY ; 140 mNm NETT TORQUE, 10380 RPM — TOP : INVERTER OUTPUT VOLTAGE (20V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (20V/cm) — BOTTOM : PHASE CURRENT (5A/cm) — 1ms/cm.

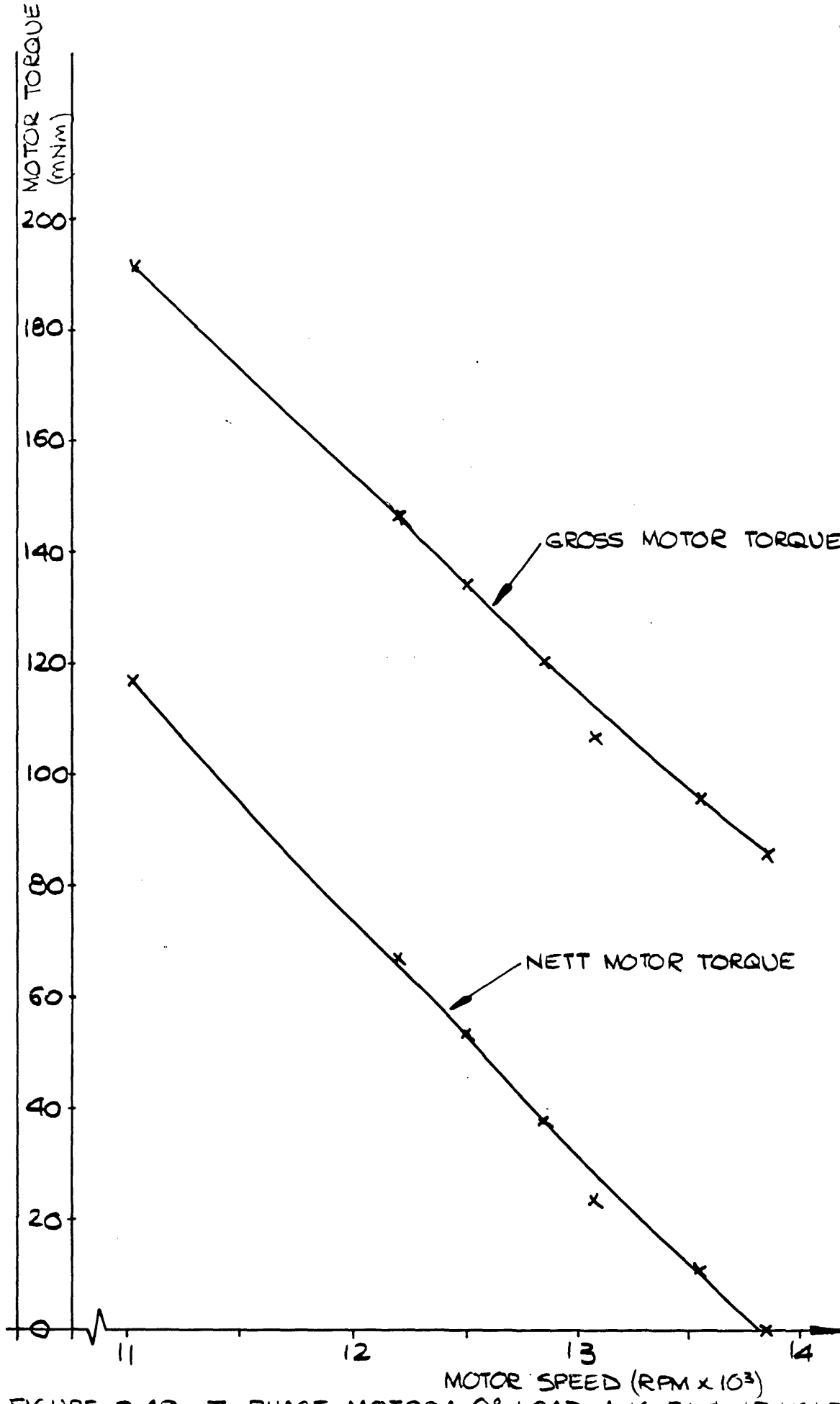


FIGURE 3.49 7 PHASE MOTOR: 0° LOAD ANGLE; 1 15 VOLT SUPPLY TO INVERTER - GROSS TORQUE AND NETT TORQUE

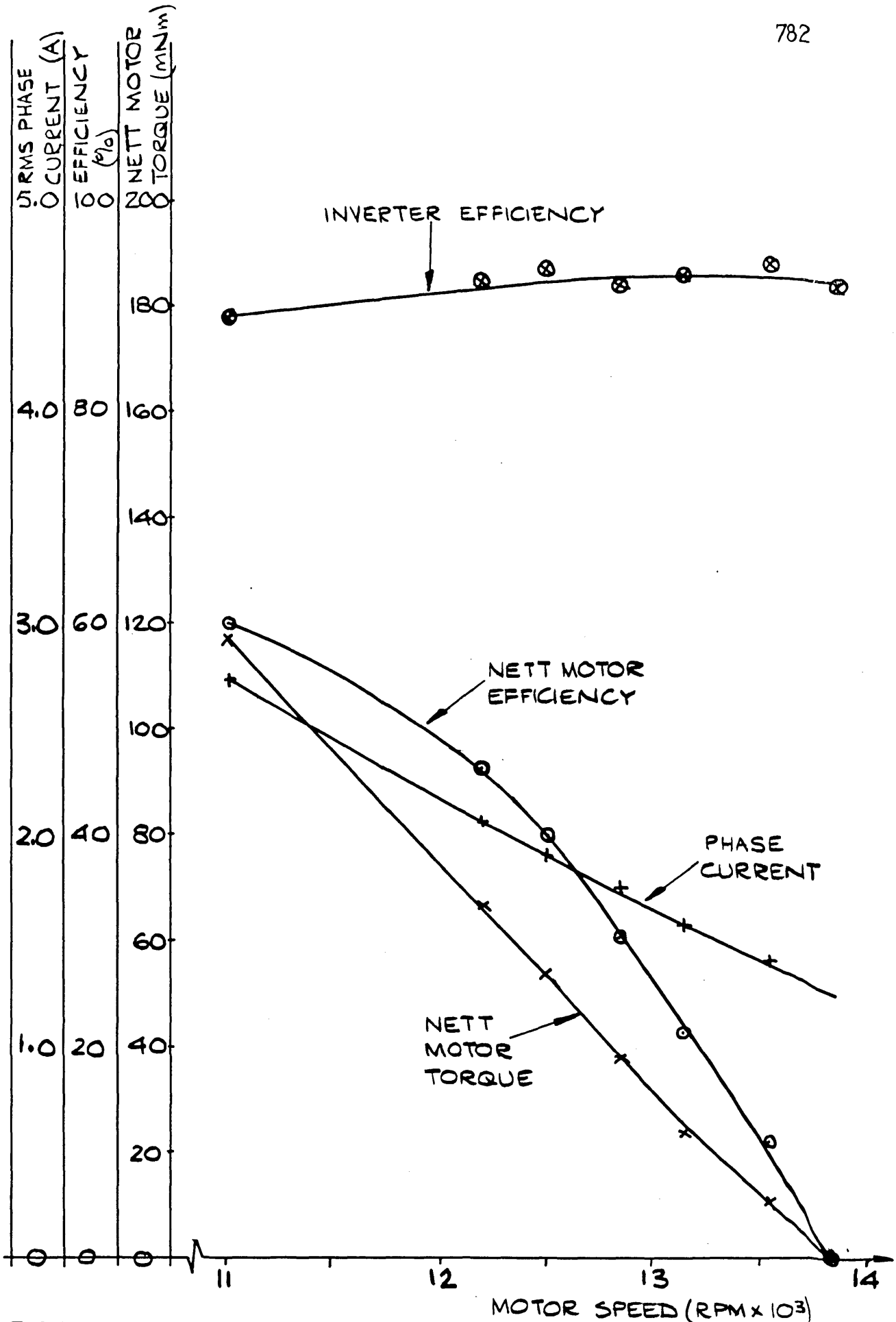


FIGURE 3.50 7 PHASE MOTOR: 0° LOAD ANGLE; ± 13 VOLT SUPPLY TO INVERTER - NETT TORQUE, NETT MOTOR EFFICIENCY, PHASE CURRENT, AND INVERTER EFFICIENCY

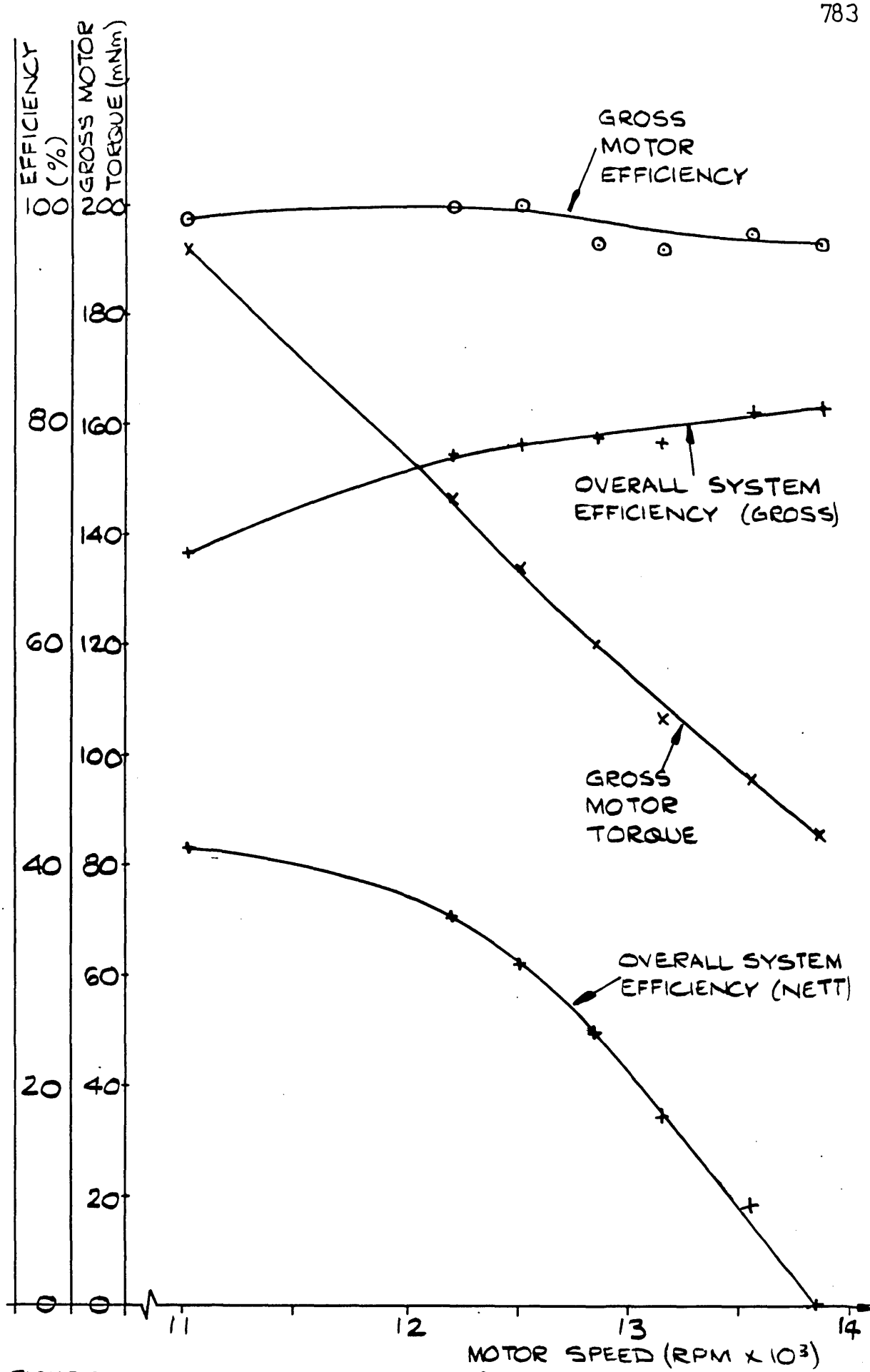


FIGURE 9.51 7 PHASE MOTOR: 0° LOAD ANGLE; ± 19 VOLT SUPPLY TO INVERTER - GROSS MOTOR TORQUE, GROSS MOTOR EFFICIENCY, OVERALL SYSTEM EFFICIENCY (GROSS) AND OVERALL SYSTEM EFFICIENCY (NETT)

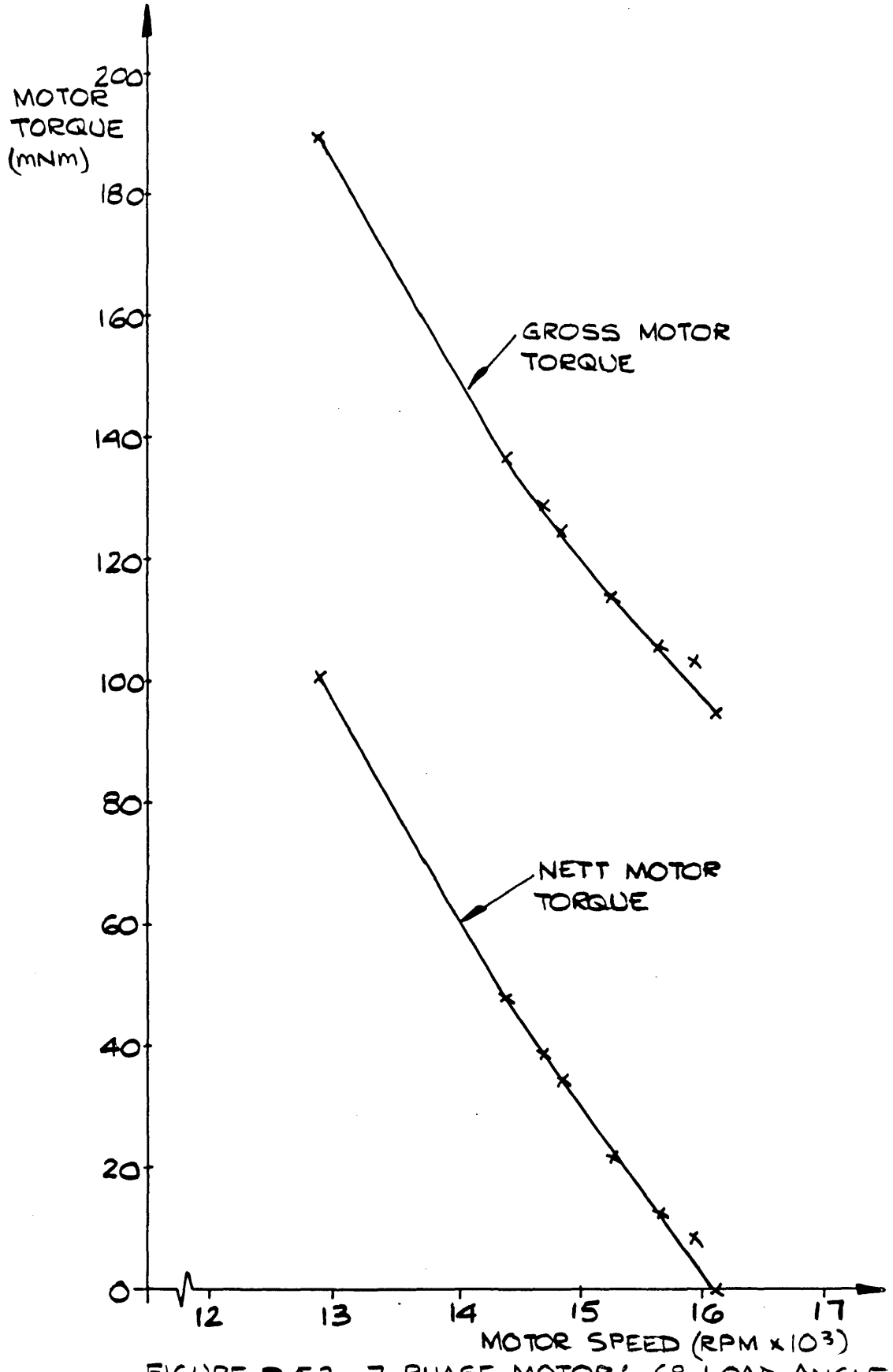


FIGURE 3.52 7 PHASE MOTOR: 6° LOAD ANGLE; ± 21 VOLT SUPPLY TO INVERTER - GROSS TORQUE AND NETT TORQUE

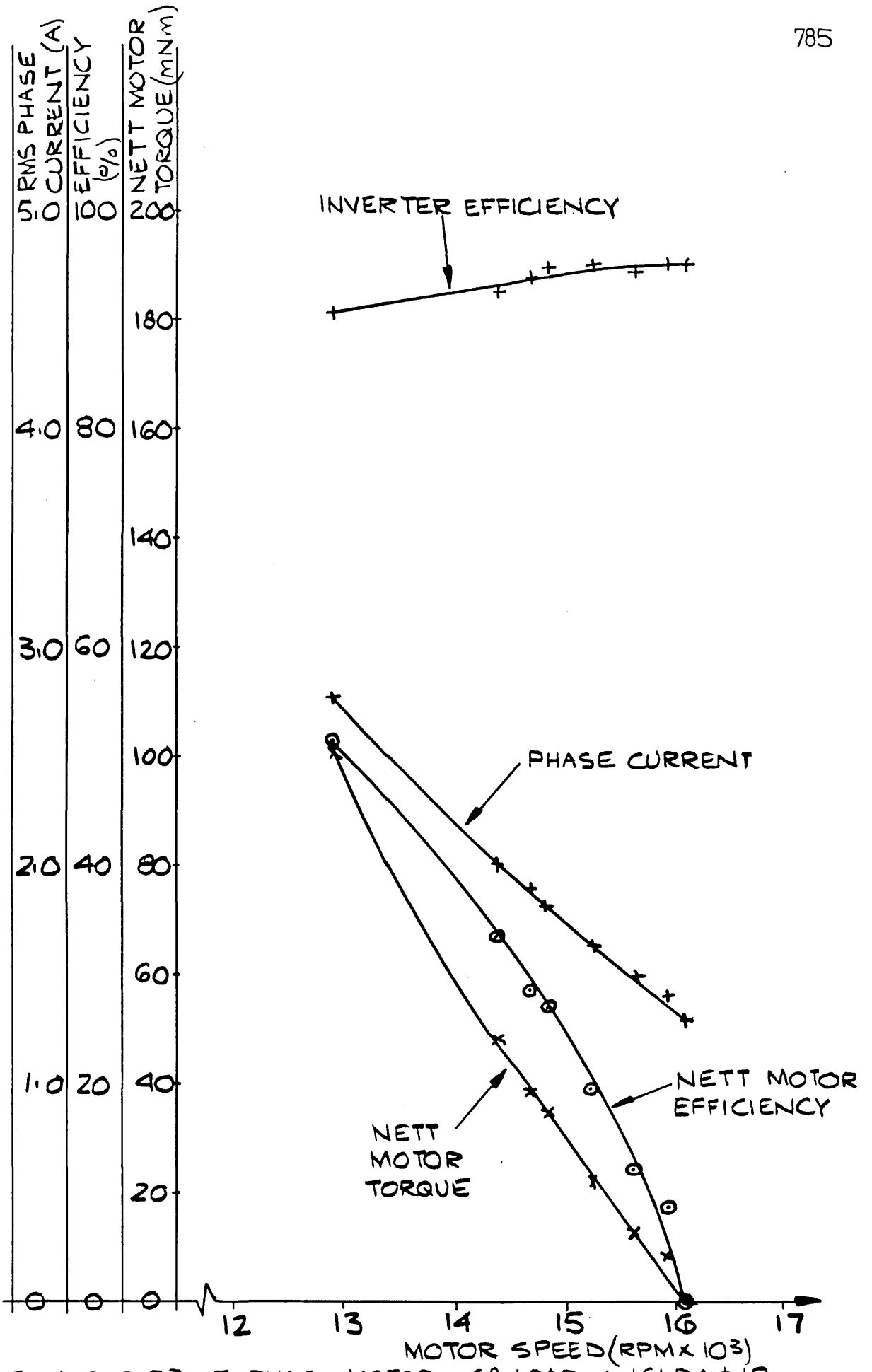


FIGURE 9.53 7 PHASE MOTOR: 6° LOAD ANGLE; ±12 VOLT SUPPLY TO INVERTER - NETT TORQUE, NETT MOTOR EFFICIENCY, PHASE CURRENT, AND INVERTER EFFICIENCY

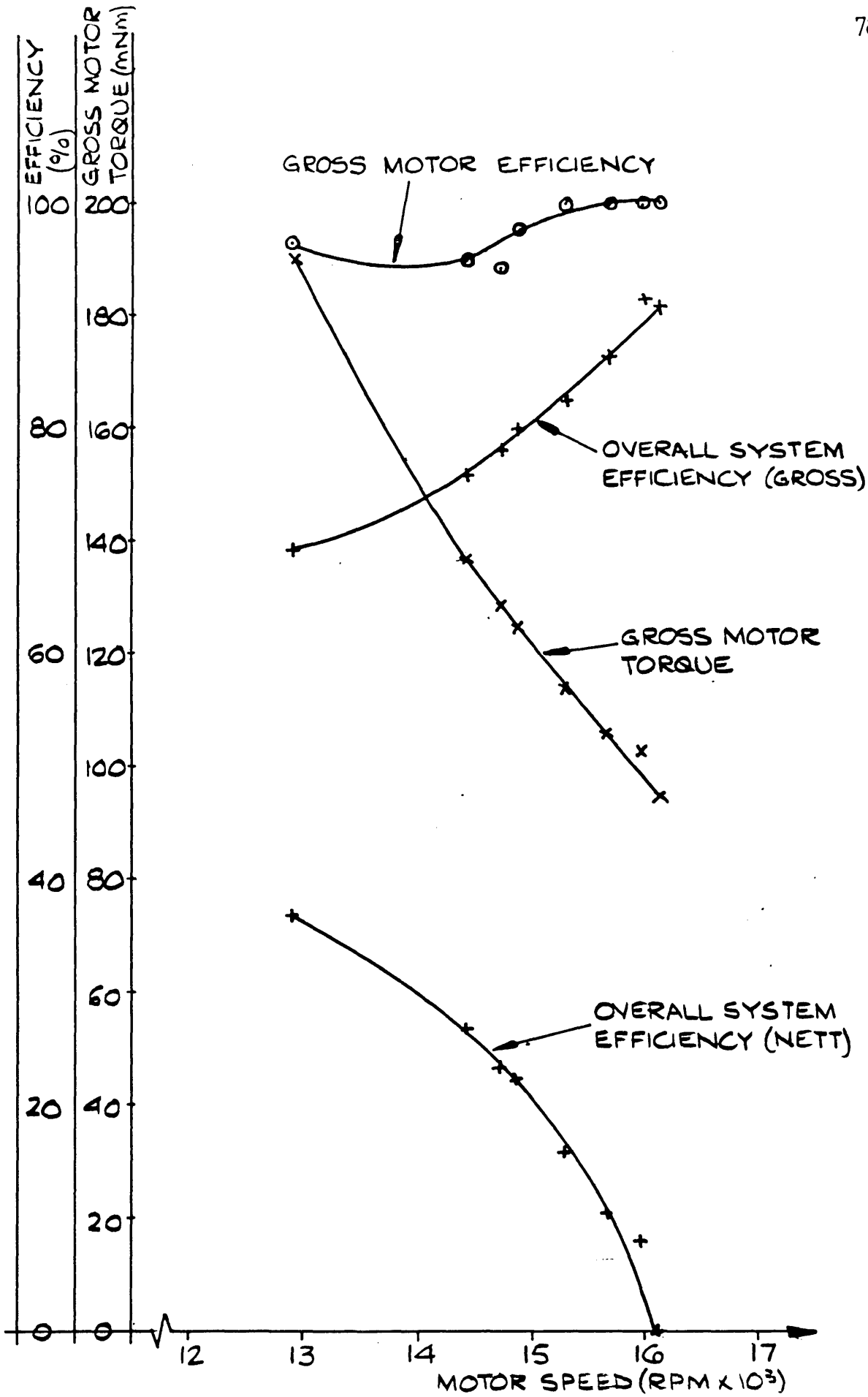


FIGURE 3.54 7 PHASE MOTOR: 6° LOAD ANGLE; ±21 VOLT SUPPLY TO INVERTER - GROSS MOTOR TORQUE, GROSS MOTOR EFFICIENCY, OVERALL SYSTEM EFFICIENCY (GROSS) AND OVERALL SYSTEM EFFICIENCY (NETT)

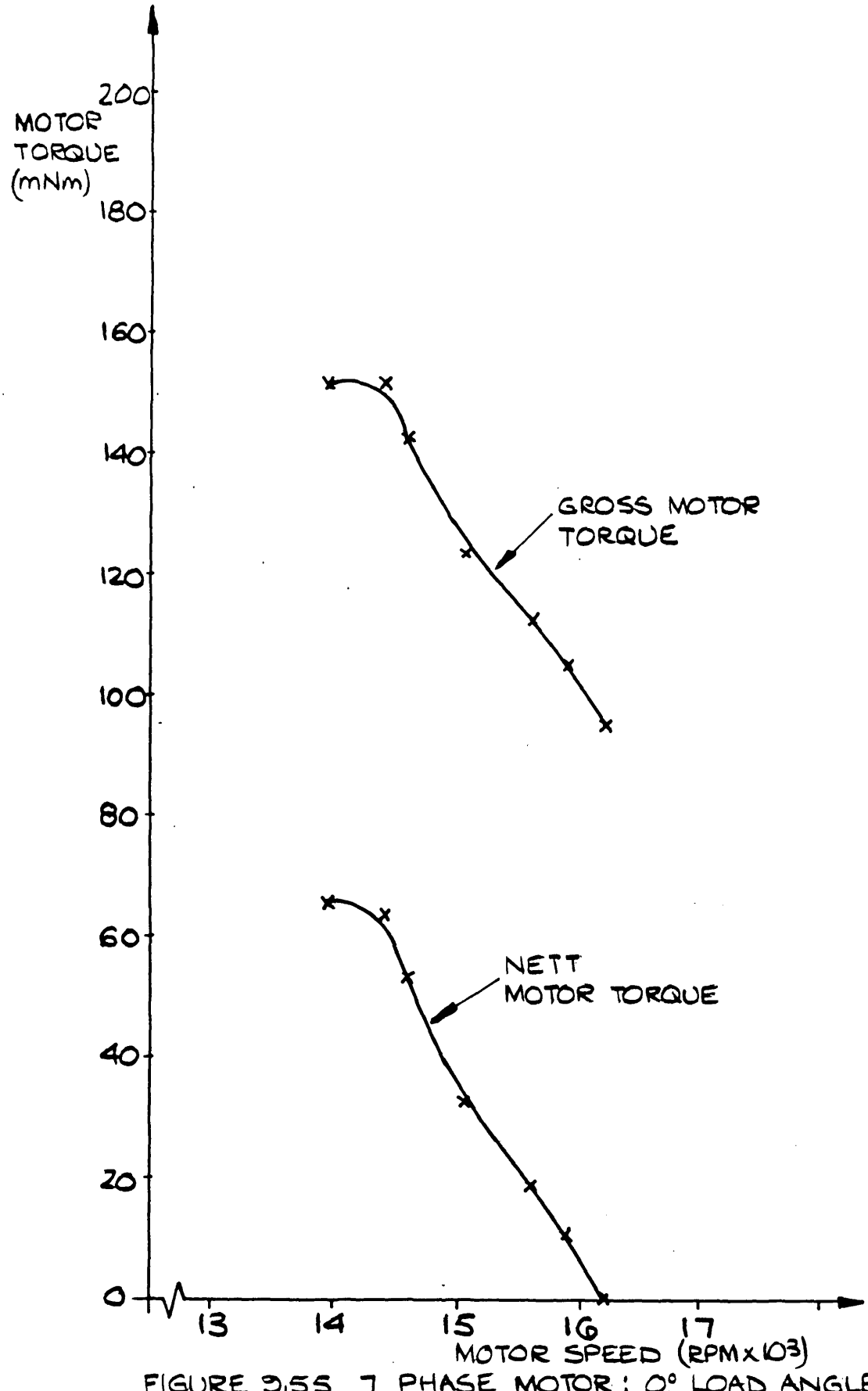


FIGURE 3.55 7 PHASE MOTOR; 0° LOAD ANGLE; ± 22 VOLT SUPPLY TO INVERTER - GROSS TORQUE AND NETT TORQUE

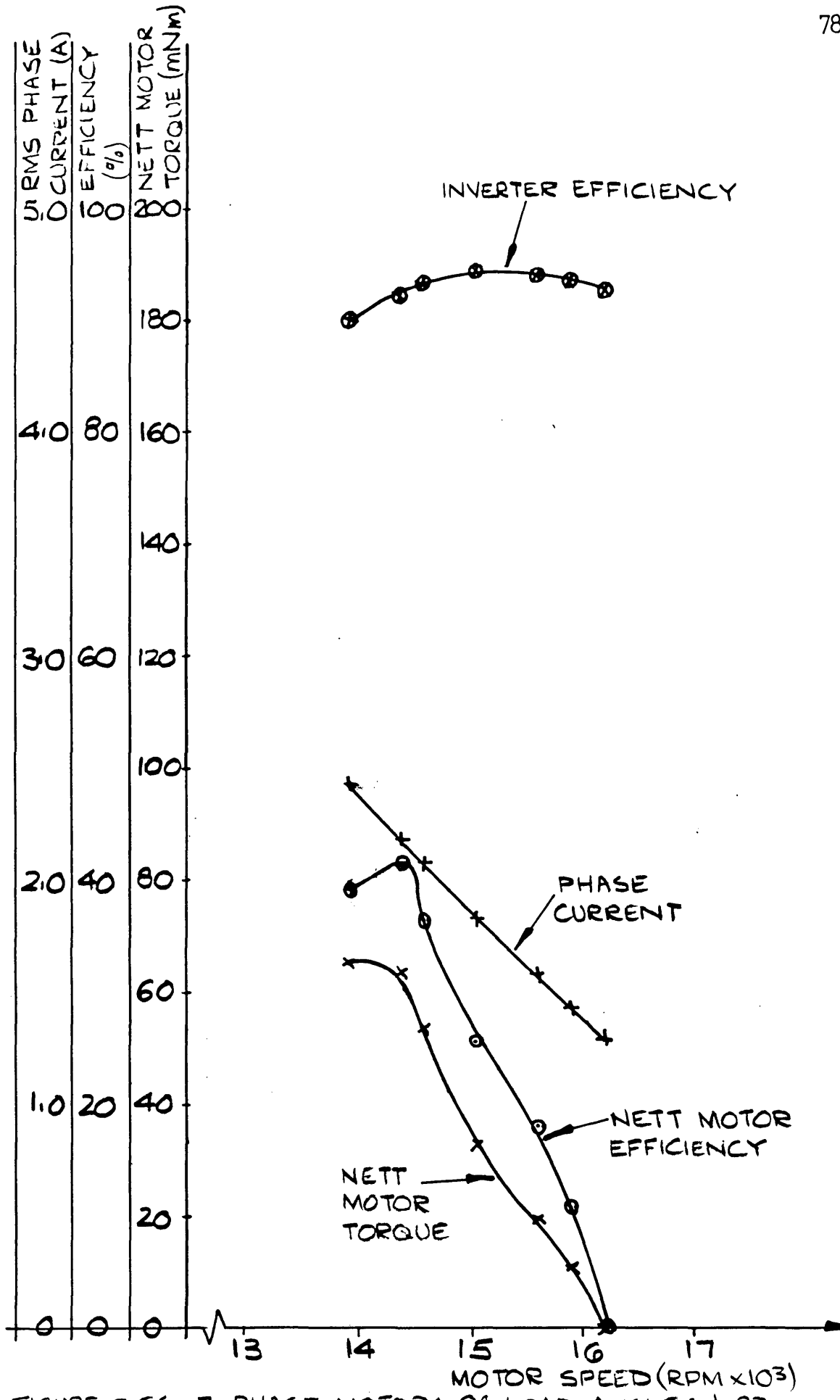


FIGURE 3.56 7 PHASE MOTOR: 0° LOAD ANGLE; ± 22 VOLT SUPPLY TO INVERTER - NETT TORQUE, NETT MOTOR EFFICIENCY, PHASE CURRENT AND INVERTER EFFICIENCY

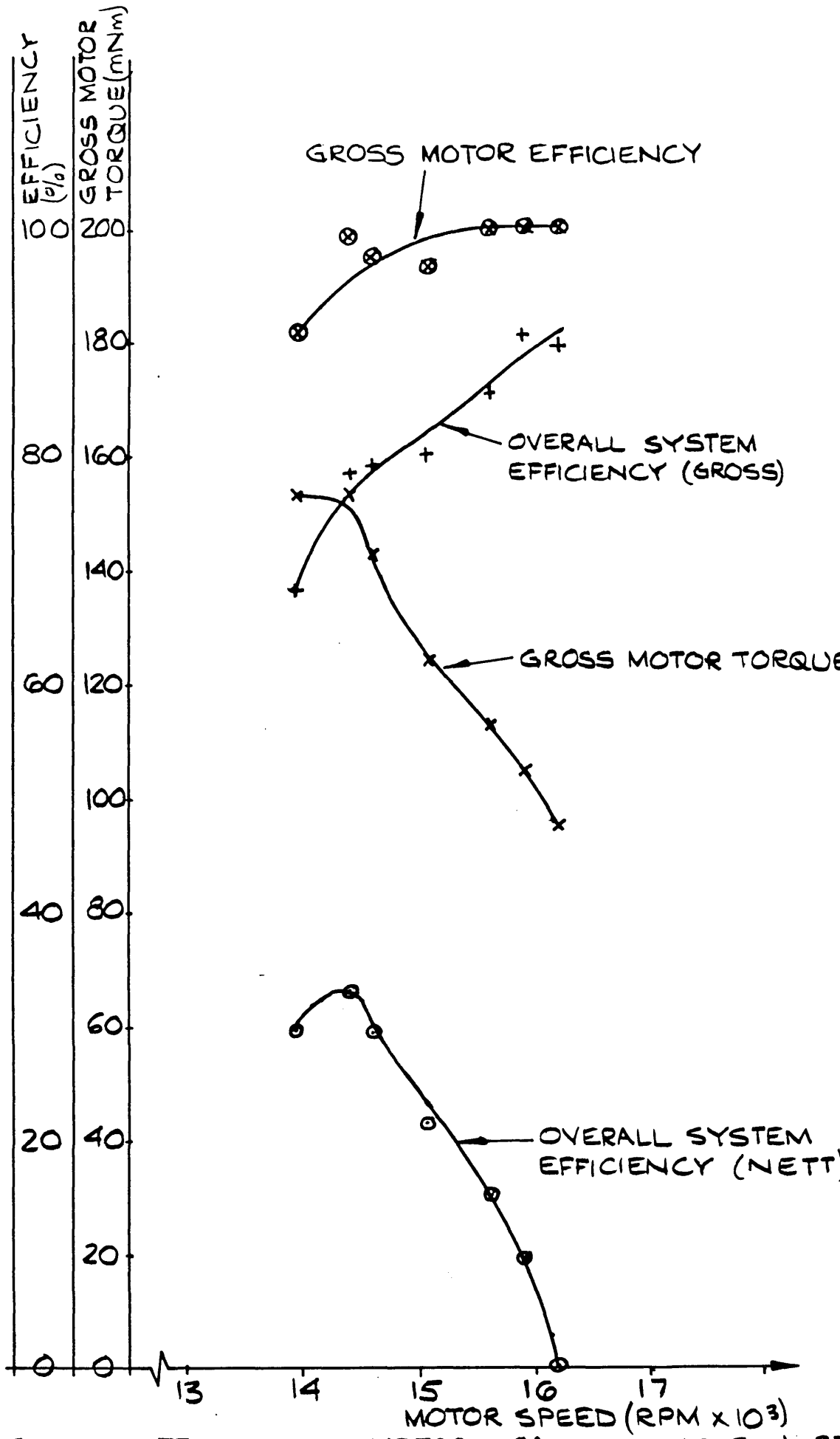


FIGURE 9.57 7 PHASE MOTOR: 0° LOAD ANGLE; ± 22 VOLT SUPPLY TO INVERTER - GROSS MOTOR TORQUE, GROSS MOTOR EFFICIENCY, OVERALL SYSTEM EFFICIENCY (GROSS) AND OVERALL SYSTEM EFFICIENCY (NETT)

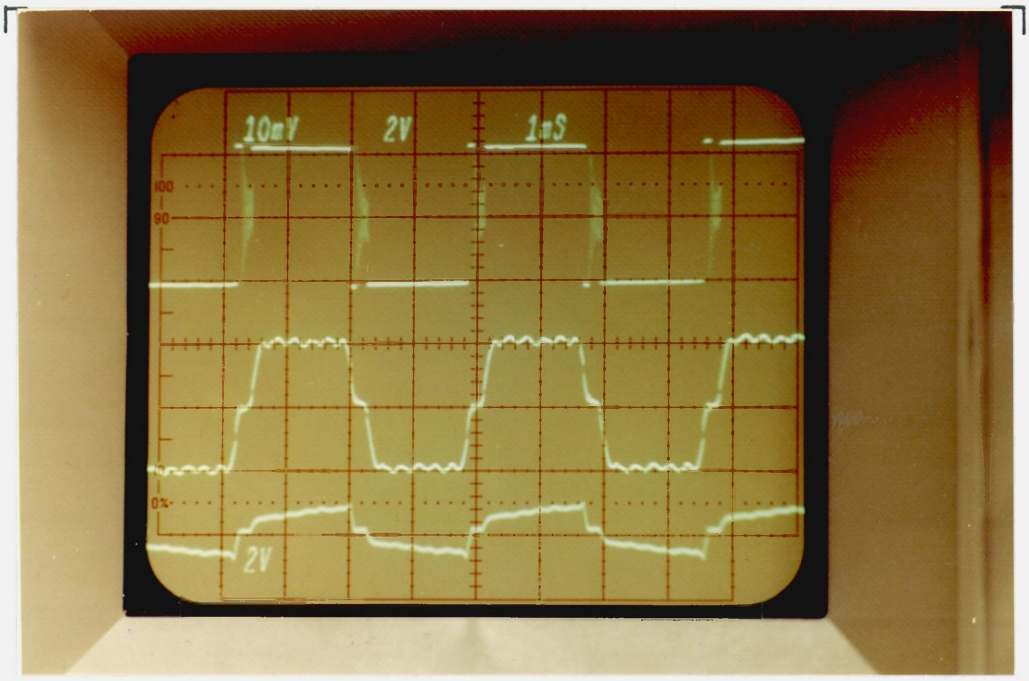


FIG. 9.58. PHASE ONE WAVEFORMS : 0° LOAD ANGLE ; ±22 VOLT INVERTER SUPPLY ; NO LOAD (ZERO NETT TORQUE), 16190 RPM — TOP : INVERTER OUTPUT VOLTAGE (20V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (20V/cm) — BOTTOM : PHASE CURRENT (5A/cm) — 1ms/cm.

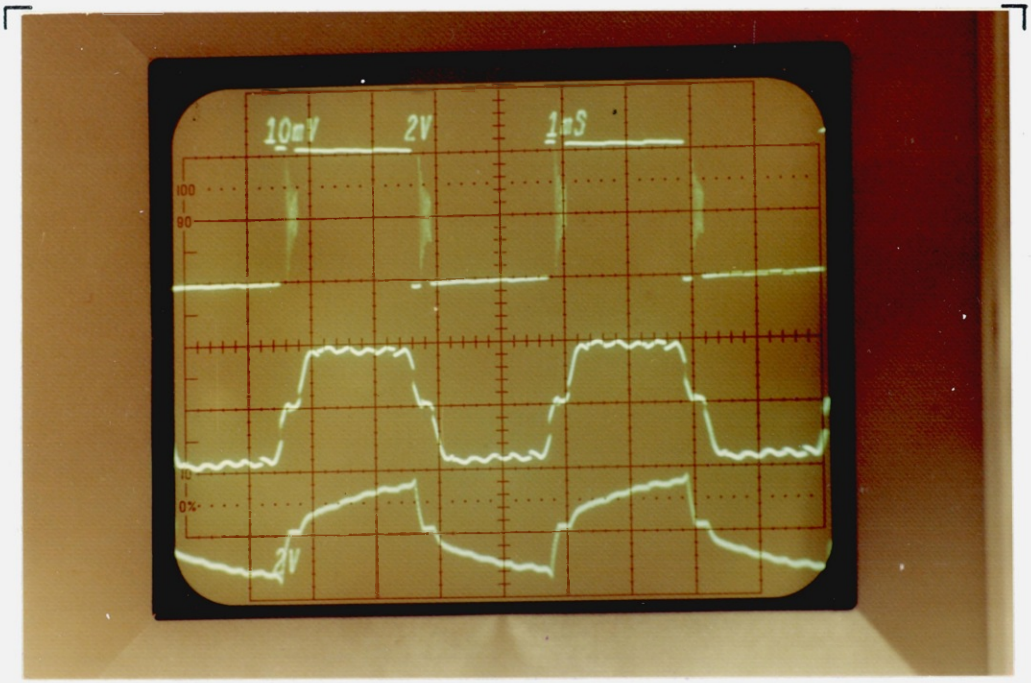


FIG. 9.59. PHASE ONE WAVEFORMS : 0° LOAD ANGLE ; ±22 VOLT INVERTER SUPPLY ; 66mNm NETT TORQUE, 13960 RPM — TOP : INVERTER OUTPUT VOLTAGE (20V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (20V/cm) — BOTTOM : PHASE CURRENT (5A/cm) — 1ms/cm.

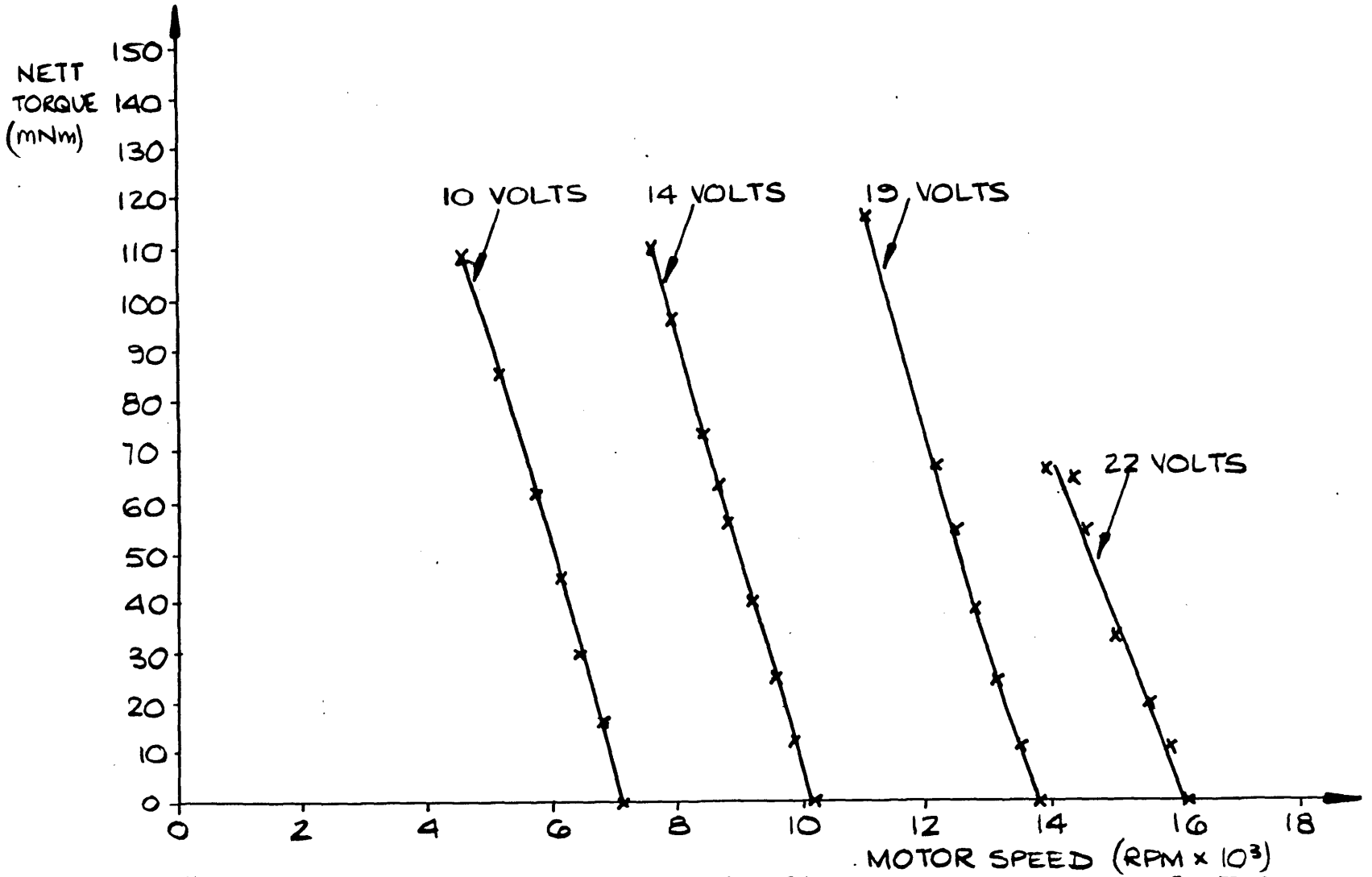


FIGURE 3.60 7 PHASE MOTOR: 0° LOAD ANGLE; NETT MOTOR TORQUE VERSUS SPEED FOR SUPPLY VOLTAGES TO INVERTER OF ±10, ±14, ±19 AND ±22 VOLTS

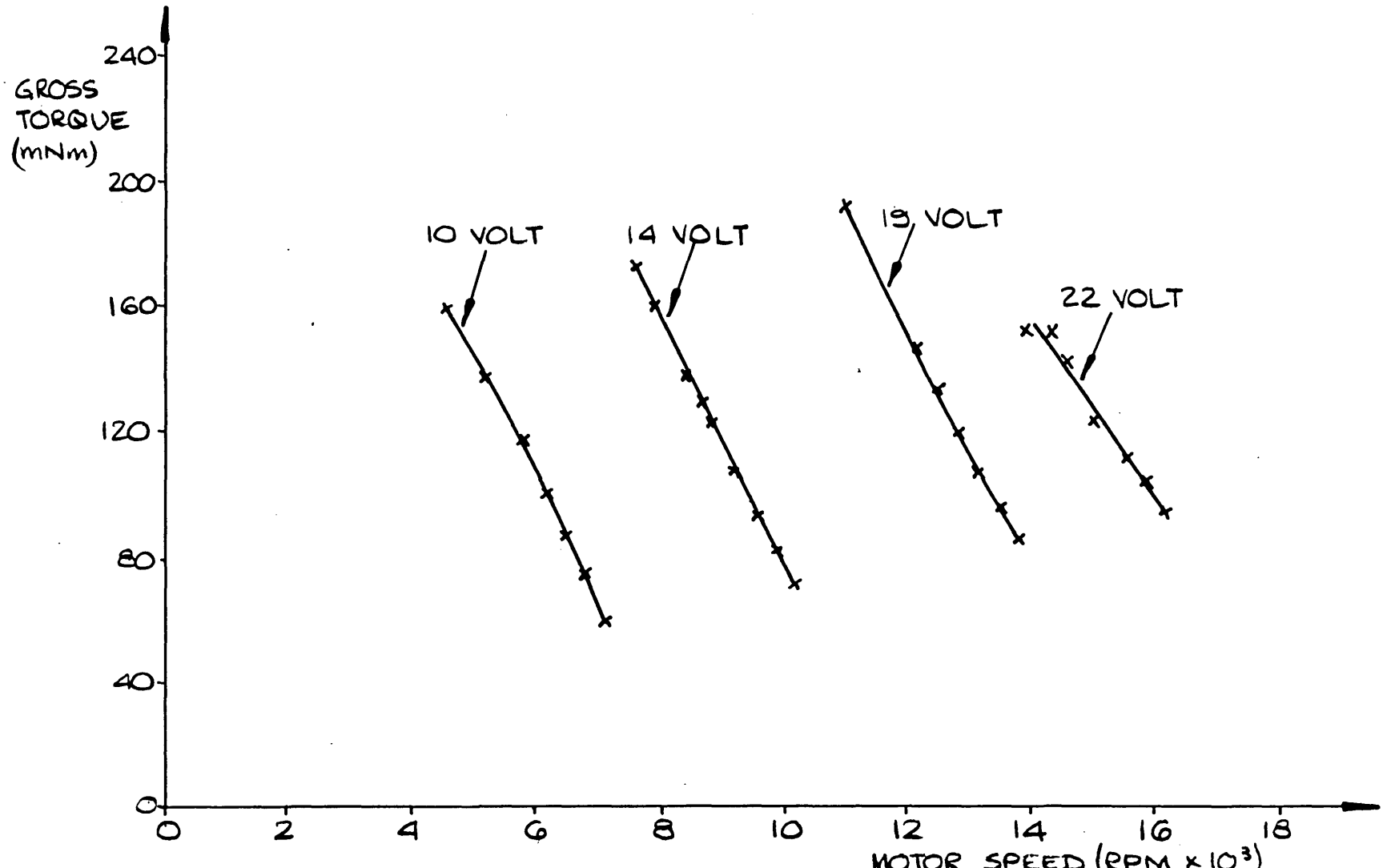


FIGURE 3.61 7 PHASE MOTOR: 0° LOAD ANGLE; GROSS MOTOR TORQUE VERSUS SPEED FOR SUPPLY VOLTAGES TO INVERTER OF ±10, ±14, ±19 AND ± 22 VOLTS

same set of inverter supply voltages as used for fig. 9.60. The relationship between the nett torque versus speed characteristics for a 6° load angle and inverter supply voltages of ±10, ±14, ±18, and ±21 volts is shown in fig. 9.62: the corresponding relationship between the gross torque versus speed characteristics is shown in fig. 9.63. The nett and gross torque versus speed characteristics for a 13° load angle and inverter supply voltages of ±10 and ±18 volts are similarly presented in figs. 9.64 and 9.65.

The relationships shown in figs. 9.60 to 9.65 inclusive, indicate that for a given load angle the motor torque/speed characteristic has the same "shape" and slope irrespective of the inverter supply voltage. The inverter supply voltage appears to set the point along the speed axis at which the characteristic intersects. The intersect speed seems to be proportional to the inverter supply voltage. Reference back to earlier graphs, such as those presented in figs. 9.15 and 9.16, shows that there is some change in the torque versus speed characteristic when the load angle is changed under constant inverter supply voltage conditions. For the range of load angles used in the tests (0° to 13°), the changes in the torque/speed characteristics are small, and so the setting of the load angle is not critical in this respect. However, the load angle setting does affect the phase current waveform substantially. Hence it is possible to alter the load angle so that the current waveform is essentially quasi-square, whilst not affecting the motor torque to any great degree.

In general it is reasonable to state that the power factor of the motor is good and close to unity. It is possible to trim the power factor to unity by making small adjustments to the operating load angle, the required adjustment being larger at higher motor speeds. The 0°, 6°, and 13° load angles used in the tests were sufficient to trim the current waveform into an approximately unity power factor condition, although it should be noted that no

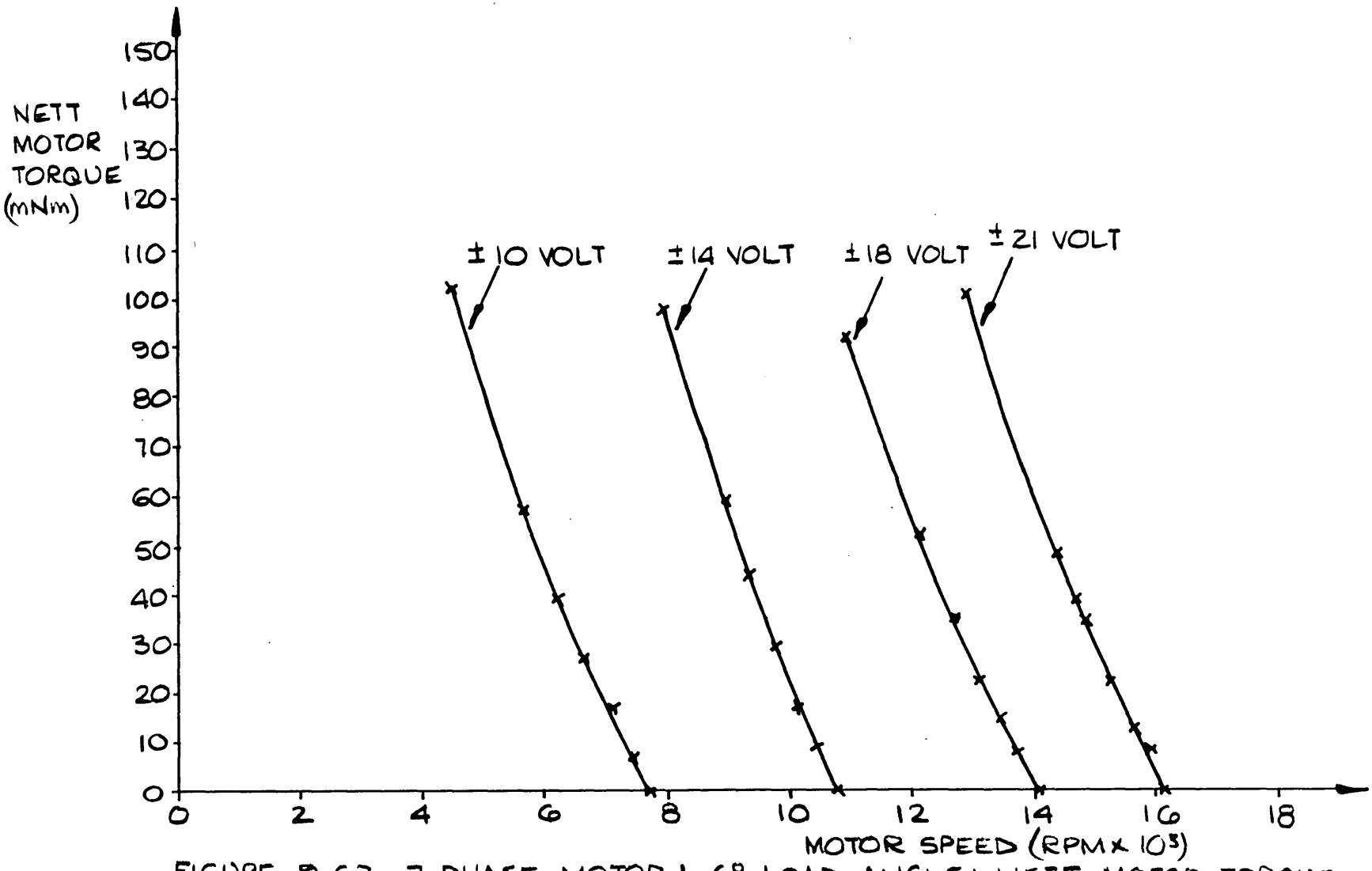


FIGURE 9.62 7 PHASE MOTOR: 6° LOAD ANGLE; NETT MOTOR TORQUE VERSUS SPEED FOR SUPPLY VOLTAGES TO INVERTER OF ±10, ±14, ±18 AND ±21 VOLTS

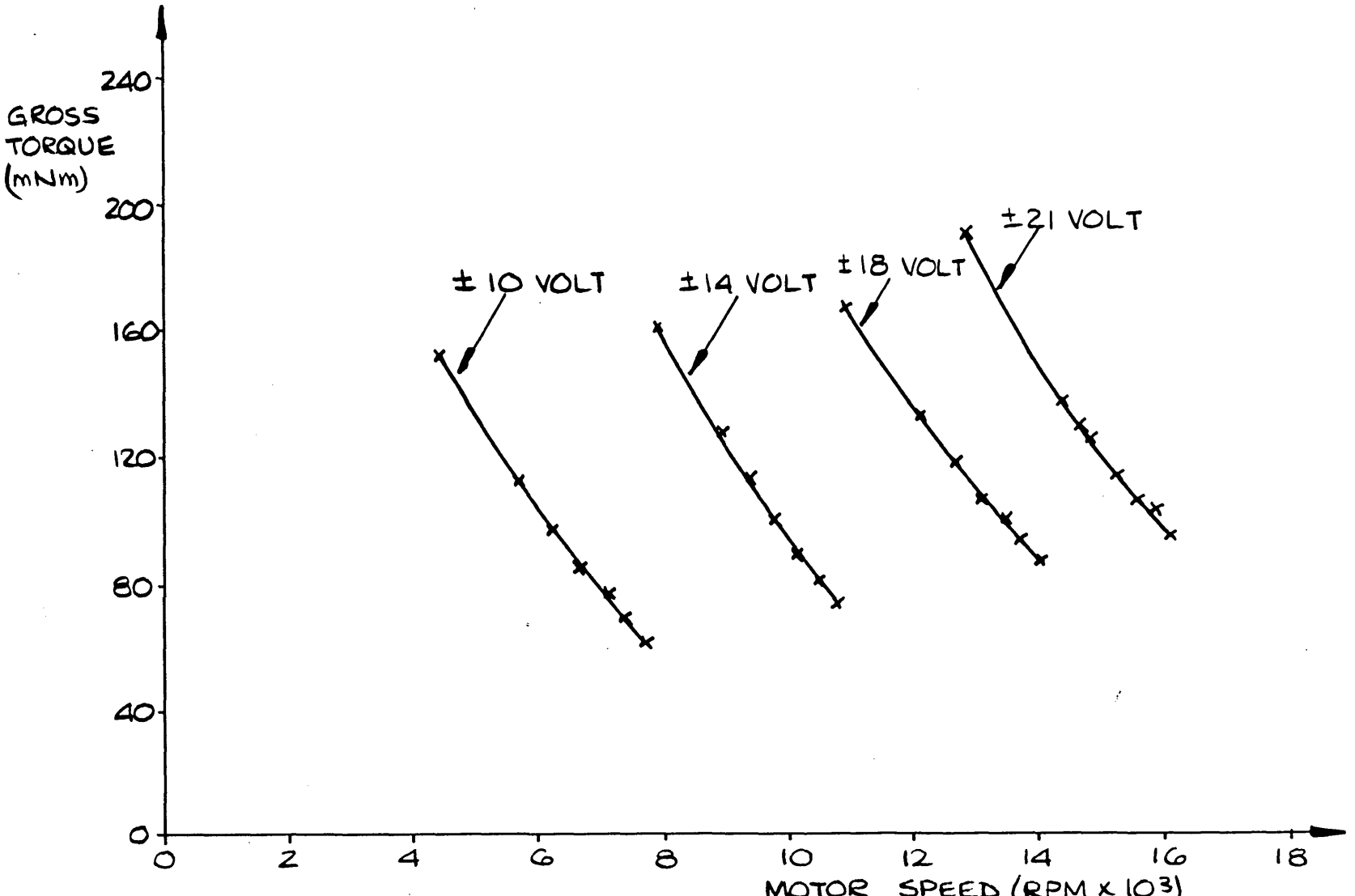


FIGURE 9.63 7 PHASE MOTOR: 6° LOAD ANGLE; GROSS TORQUE VERSUS SPEED FOR SUPPLY VOLTAGES TO INVERTER OF ±10, ±14, ±18 AND ±21 VOLTS

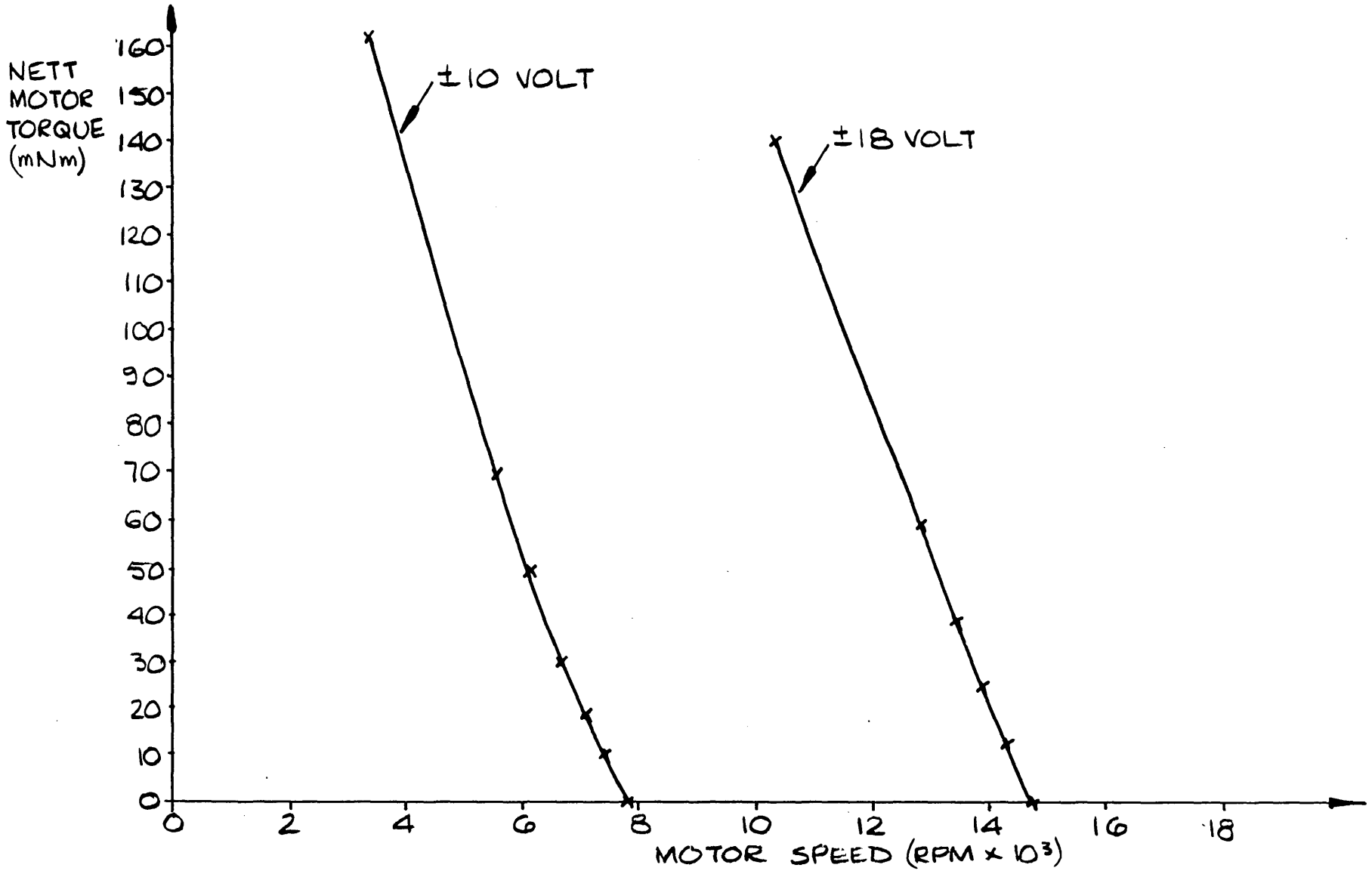


FIGURE 9.64 7 PHASE MOTOR: 13° LOAD ANGLE; NETT MOTOR TORQUE VERSUS SPEED FOR SUPPLY VOLTAGES TO INVERTER OF ±10 AND ±18 VOLTS

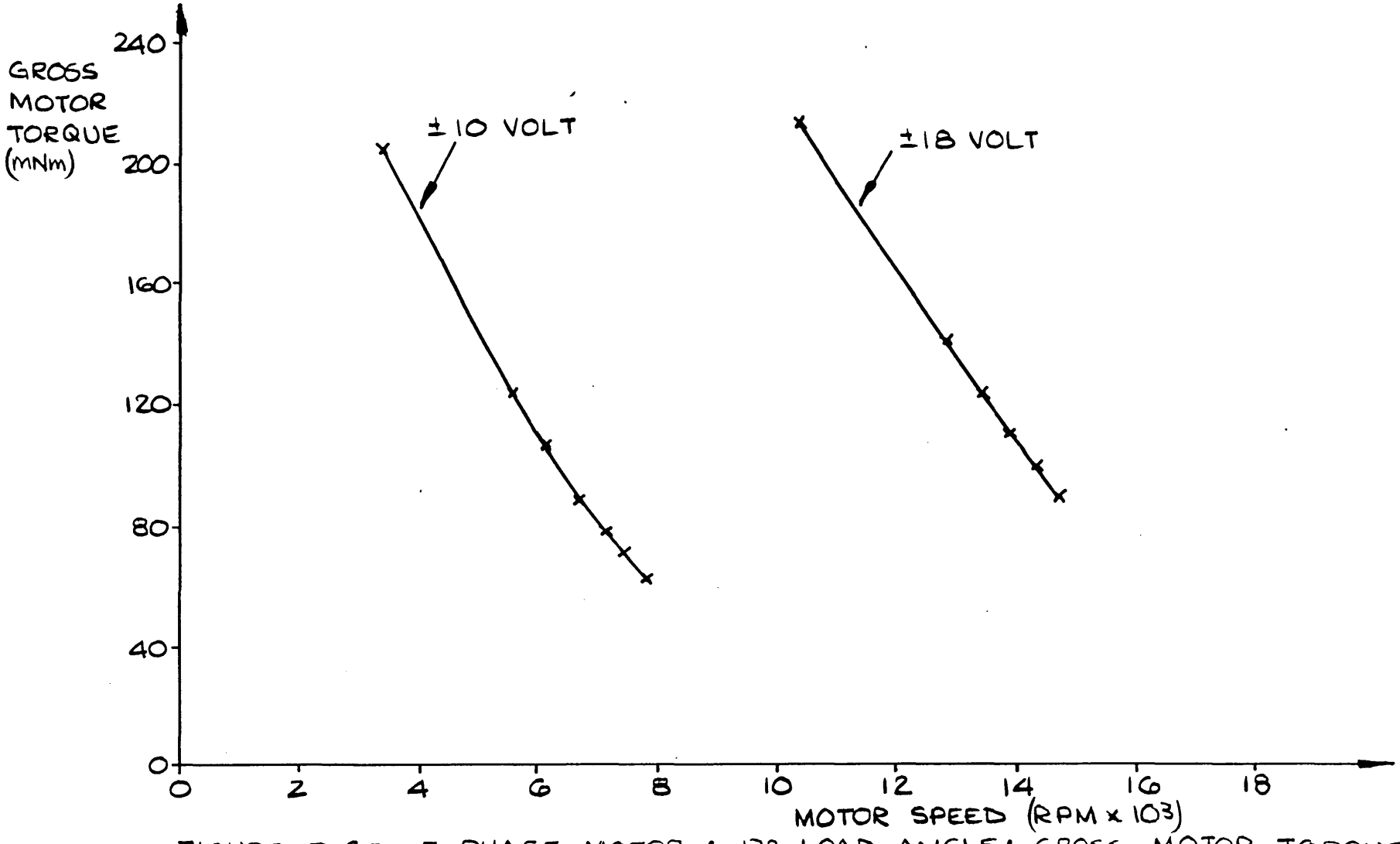


FIGURE 3.65 7 PHASE MOTOR : 13° LOAD ANGLE; GROSS MOTOR TORQUE VERSUS SPEED FOR SUPPLY VOLTAGES TO INVERTER OF ±10 AND ±18 VOLTS

accurate power factor measurements were made, (the power factor being assessed visually by means of oscilloscope waveforms).

A breakdown of the power dissipation in the inverter/motor system for an inverter supply voltage of ±10 volts and 0° load angle is shown in fig. 9.66. A similar breakdown for conditions of ±19 volts and 0° load angle is shown in fig. 9.67. Taking into account the fact that the gross motor efficiency is not necessarily totally accurate (due to inaccuracies in the loss-torque measurement), it is nevertheless possible to conclude from the results that there are no stray losses within the motor. Allowing for measurement errors it is reasonable to state that the power balance in the system is good; that is, all the input power can be accounted for. The bulk of the motor losses are due to the I.W.F. loss-torque suffered by the rotor. The vast majority of the loss-torque is due to eddy currents induced in the stator by the rotating rotor flux; the excessive stator heating during motor operation was clear evidence of this. There are, of course, some heating losses due to the motor phase currents flowing through the resistance of the phase windings. The phase currents also generate eddy currents in the stator-iron. However, both of these phase current associated losses are small: the resistance loss is small because the phase winding resistance is very low; the stator eddy current losses due to the phase current are small because the stator was deliberately designed to minimise stator flux. The phase winding resistance losses are in fact negligible in comparison to the losses in the series protection components and the inverter hexfet components.

The motor winding losses are not shown on fig. 9.67 because inaccuracies in the I.W.F. losses meant that the winding losses were "surplus to requirements". This is not a serious omission though because the winding losses are so small by comparison with the other losses in the

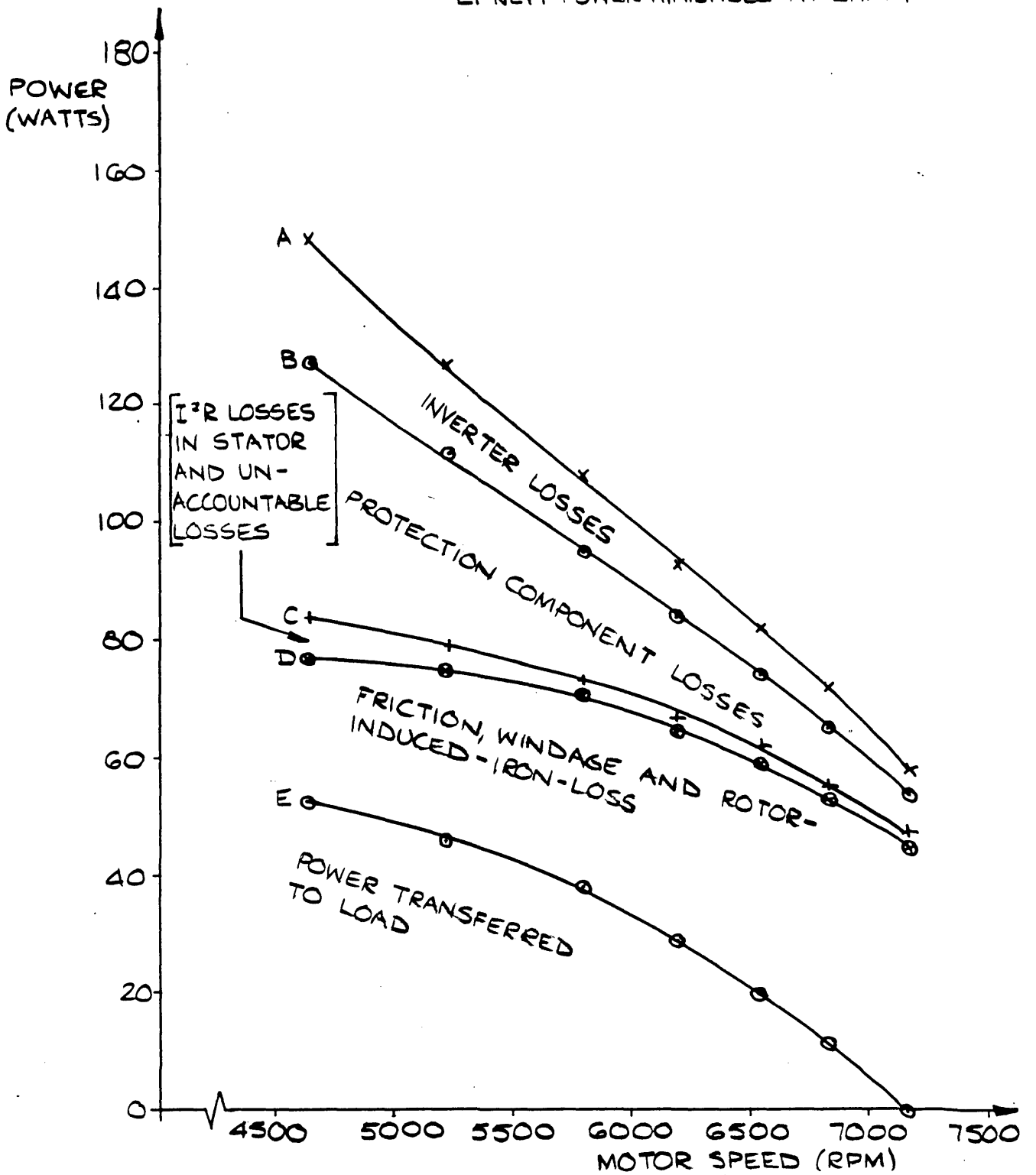


FIGURE 9.66 POWER DISSIPATION IN THE 7 PHASE INVERTER/MOTOR SYSTEM WITH ±10 VOLT SUPPLY TO INVERTER; 0° LOAD ANGLE

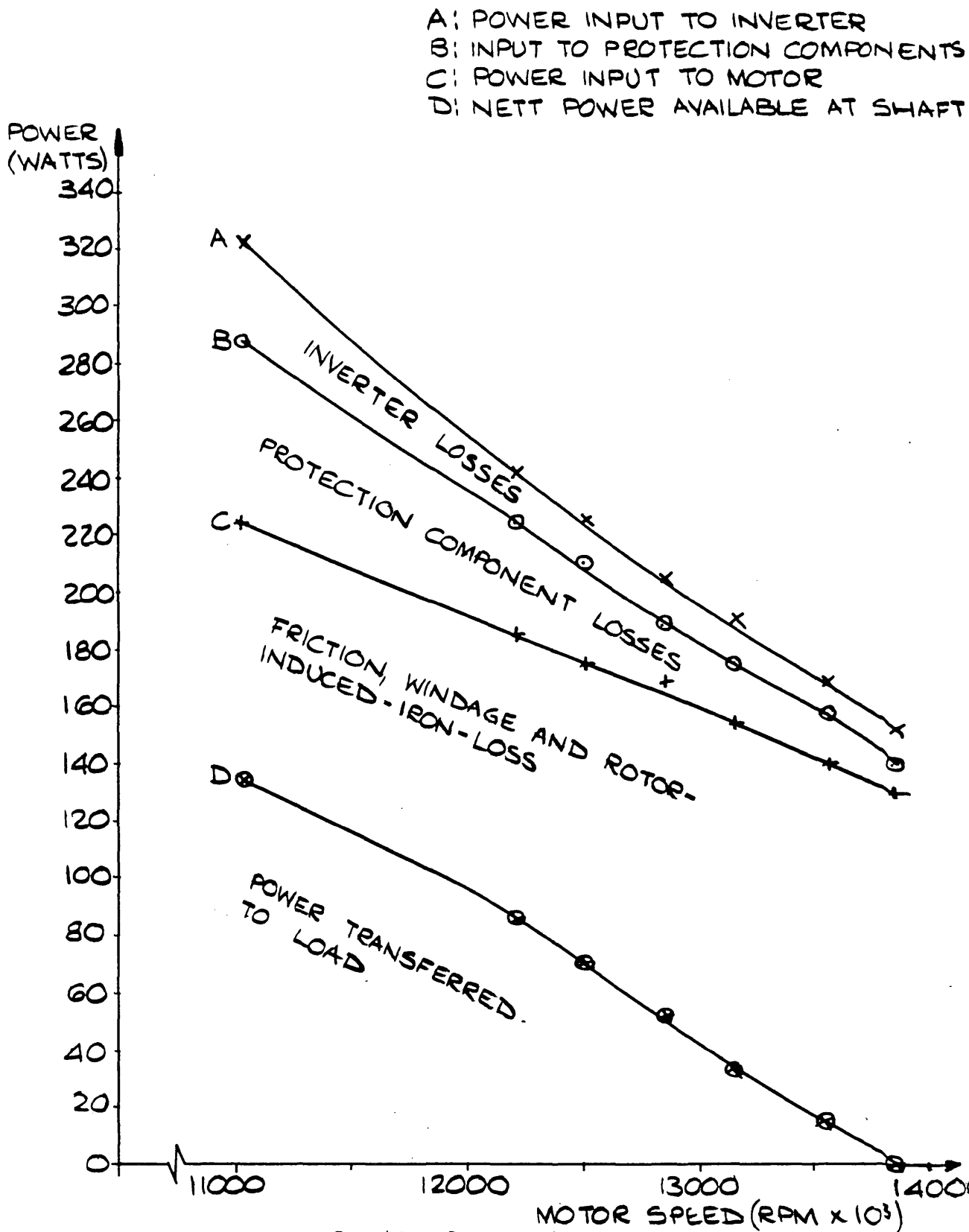


FIGURE 9.67 POWER DISSIPATION IN THE 7 PHASE INVERTER/MOTOR SYSTEM WITH ± 19 VOLT SUPPLY TO INVERTER; 0° LOAD ANGLE

whole system.

Examination of figs. 9.66 and 9.67 shows that the losses within the motor dominate all other losses at present. The I.W.F. rotor-borne losses could be significantly reduced by appropriate stator design: that is to say, by the use of higher quality electrical steel and by proper treatment of the laminations. A magnetic stator material like ferrite where eddy current losses are zero could even be considered. If the motor losses were reduced to conventional levels, the main losses in the system would then be in the protection components. This is quite acceptable, since the protection components are passive devices which can be located in suitably ventilated positions.

Figs. 9.66 and 9.67 show that the inverter loss rises as the motor speed falls under load: that is, as the phase current rises. A rise in inverter losses with increasing phase current is consistent with the resistive nature of the hexfets. Further comments on the inverter efficiency are made in section 9.4.

9.3.4 The 7 Phase Motor Performance Characteristics for Constant Applied Phase Voltages

Torque/speed characteristics were obtained from the 7 phase motor system with the motor terminal phase voltages held at various constant r.m.s. values. For a given inverter supply voltage, the motor phase voltages depend on the resistive voltage drops in the protection components and the hexfets. Hence, under fixed inverter supply voltage conditions, the phase voltages vary whenever the phase currents change. In order to maintain the motor phase voltages at constant r.m.s. values, it is necessary to monitor the phase voltages and adjust the inverter supply voltage accordingly to maintain the desired values.

Fine voltage control over the ±60 volt variable power

supplies was required in practice and this was not possible via the keypad and its associated D/A circuitry. Therefore, the programming currents for the ±60 volt power supplies were supplied from a pair of adjustable stabilised power supplies via appropriate series resistors.

Torque/speed characteristics for a load angle of 0° with r.m.s. phase voltages of 10 volts and 15 volts are shown respectively in figs. 9.68 and 9.69. Both sets of characteristics show the nett and gross motor torque, the nett motor efficiency, and the inverter efficiency. The inverter efficiency was in excess of 90% for both the 10 volt and 15 volt tests. There was very little speed droop in either test when load torque was applied. This is because the motor phase voltages are deliberately held at a constant r.m.s. value, and so the motor phase winding back-emfs are virtually forced to be constant owing to the negligible phase winding resistances. Since the back-emfs are proportional to speed, there is very little change in speed as the load is applied. The small droop in speed is due to the phase winding resistance and could be reduced by lowering the phase winding resistance. The fact that the motor phase voltage waveform is a stable quasi-square shape means that an approximately constant-speed system can be easily implemented: a simple analogue circuit can be used to measure the motor phase voltage and so control the inverter supply voltages to maintain the required phase voltage value. By this means the motor speed can be held more or less constant without requiring an actual knowledge of the rotor speed.

The ratio of the no-load speeds for the 15 volt and 10 volt tests is 1.49 and so the motor speed is virtually proportional to the motor phase voltage.

The power losses in the inverter/motor system for the 10 volt and 15 volt tests are shown respectively in figs. 9.70 and 9.71. The loss characteristics are of the same

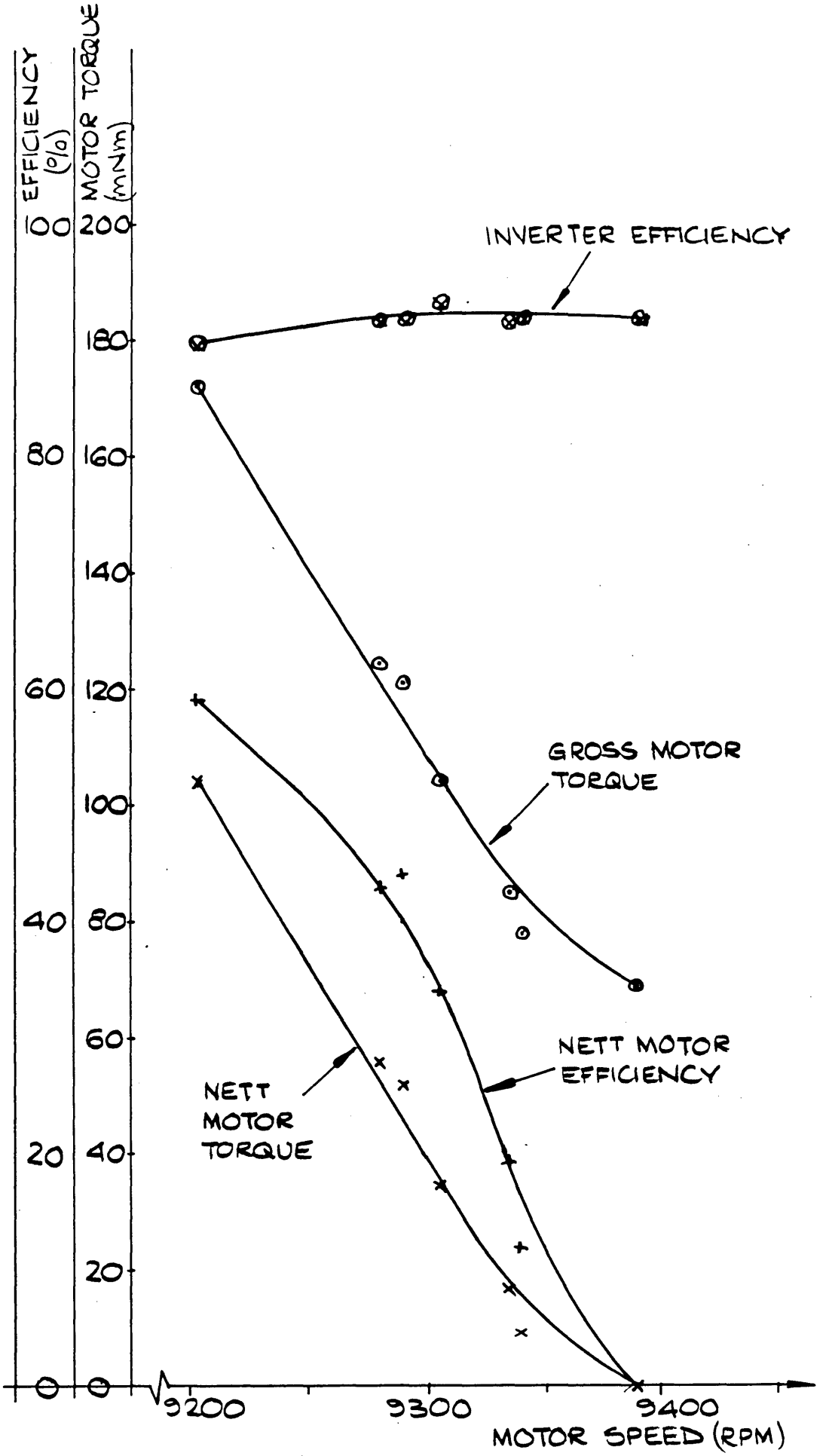


FIGURE 9.68 - 7 PHASE MOTOR: 0° LOAD ANGLE; TORQUE VERSUS SPEED (GROSS AND NETT), NETT MOTOR EFFICIENCY AND INVERTER EFFICIENCY FOR MOTOR PHASE VOLTAGE MAINTAINED AT 10 VOLTS RMS

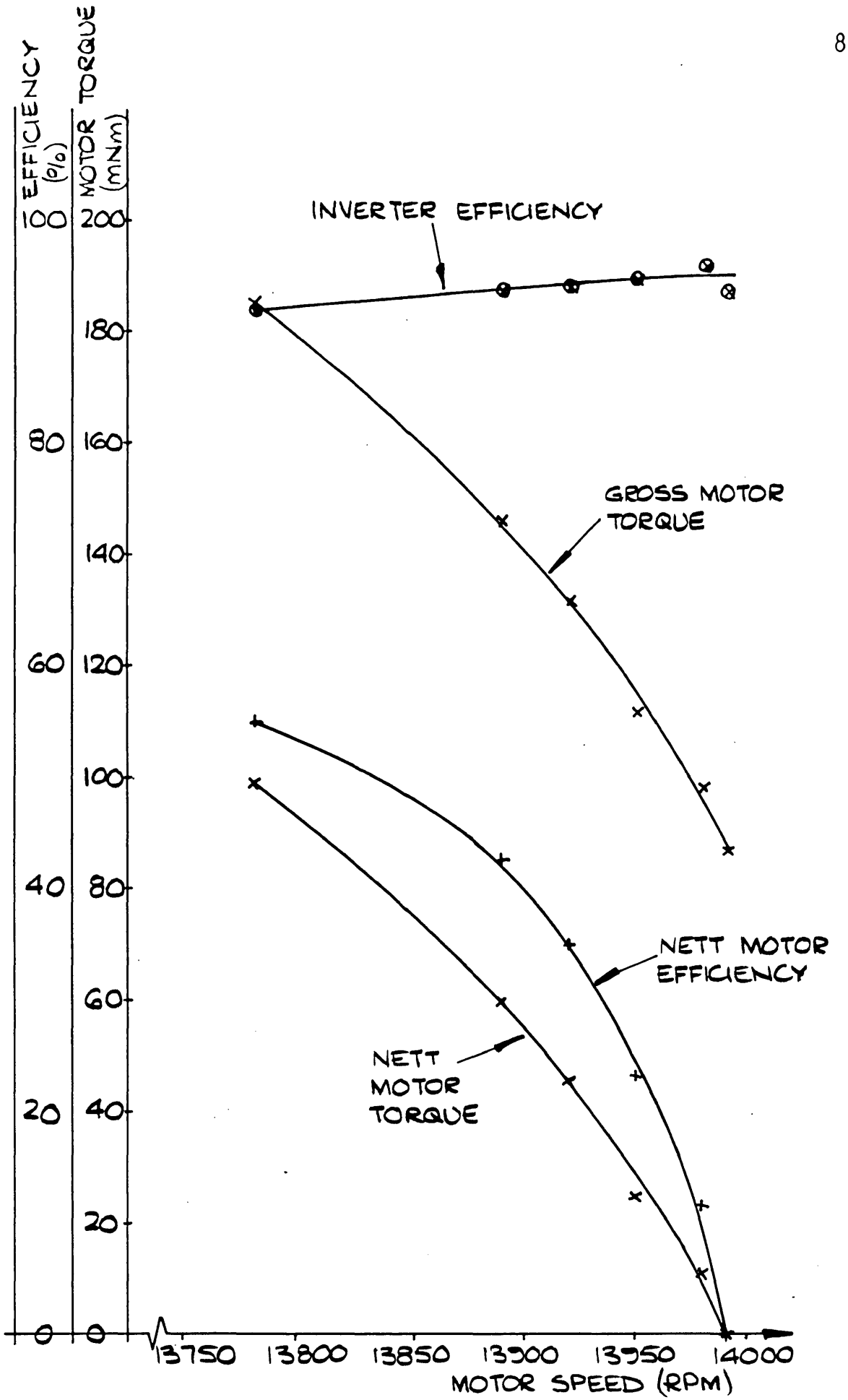


FIGURE 9.69 7 PHASE MOTOR: 0° LOAD ANGLE; TORQUE VERSUS SPEED (GROSS AND NETT), NETT MOTOR EFFICIENCY AND INVERTER EFFICIENCY FOR MOTOR PHASE VOLTAGE MAINTAINED AT 15 VOLTS RMS

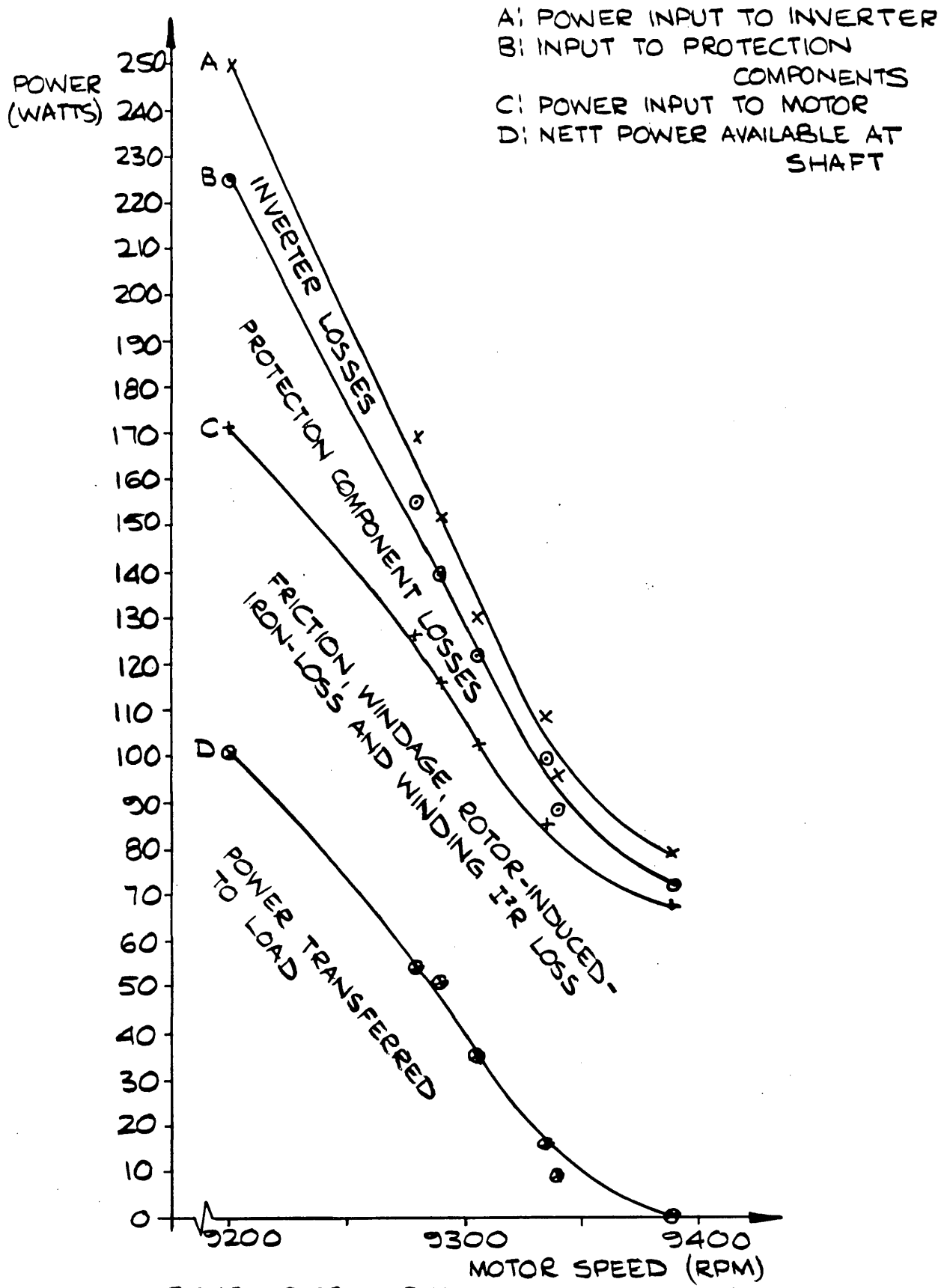


FIGURE 3.70 7 PHASE MOTOR: 0° LOAD ANGLE; CONSTANT 10 VOLT RMS PHASE VOLTAGES - POWER LOSSES THROUGHOUT SYSTEM

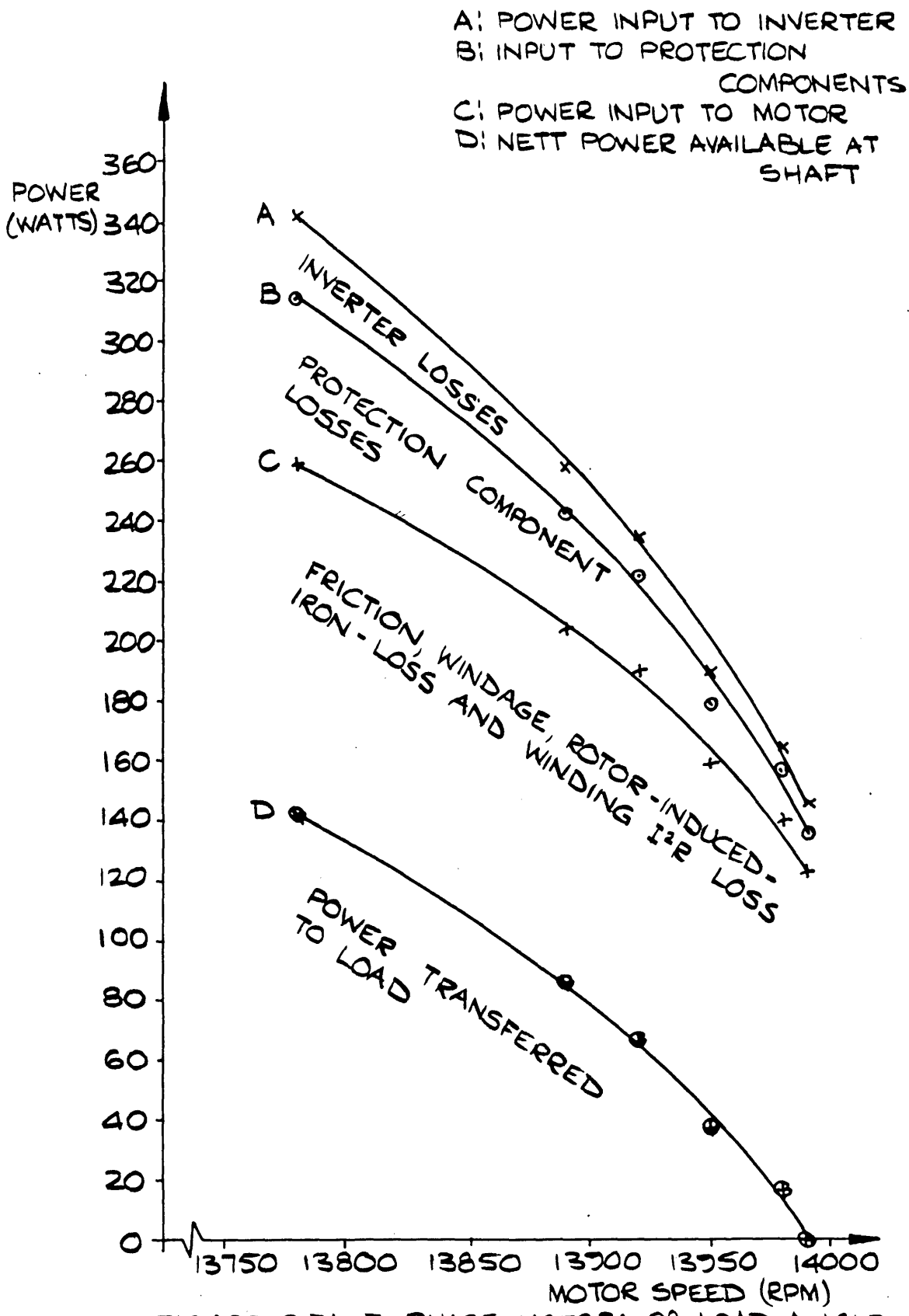


FIGURE 3.71 7 PHASE MOTOR: 0° LOAD ANGLE; CONSTANT IS VOLT RMS PHASE VOLTAGES - POWER LOSSES THROUGHOUT SYSTEM

form as those shown earlier in sub-section 9.3.3 for the constant inverter supply voltage tests, and so no further comments will be made here.

9.3.5 Torque Versus Phase Current Characteristic

In order to determine the maximum torque capability of the motor at the rated inverter phase current, it was necessary to employ the alternative loading technique using string and spring balances as described in section 9.2.

The system was run with ± 15 volts supplied to the inverter and a load angle of 0°. The nett torque versus r.m.s. phase current characteristic obtained from the system is shown in fig. 9.72. The characteristic is remarkably linear, and so it is reasonable to assume that no saturation occurs at higher phase currents. Projecting the characteristic back to intercept the current axis shows that the zero-torque phase current is about 0.85 amps r.m.s. This value agrees with the practically measured zero nett-torque phase current.

The phase current was not allowed to exceed 7 amps r.m.s. during the test. This was to prevent excessively large peak phase currents. The IRF230 hexfets can withstand peak currents of 15 amps but their continuous rating is only 7 amps. The peak current occurring for 7.0 amp r.m.s. conditions is about 7.56 amps (assuming the current waveform is the desired quasi-square shape), and this was thought to be a reasonable limit to observe.

The speed at each torque measurement was noted to enable a value of I.W.F. loss-torque to be estimated. It was assumed that the windage losses, due to the 10.4 cm diameter pulley used for the torque measurement, were the same as those suffered by the dural disc used in the eddy current brake configuration. Hence, the I.W.F. loss-torque characteristic shown in fig. 9.13 was used to estimate the

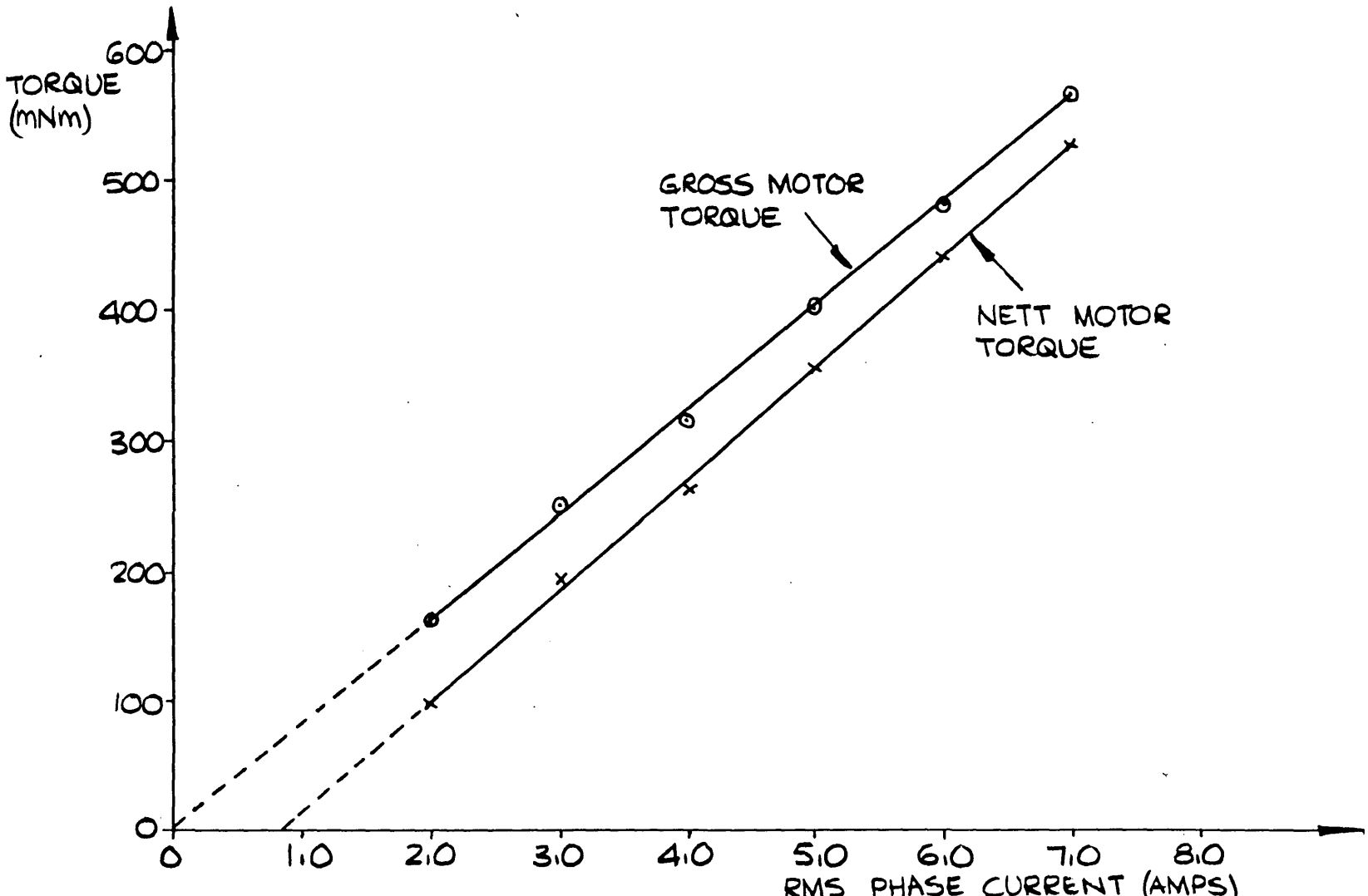


FIGURE 9.72 7 PHASE MOTOR: 0° LOAD ANGLE; SYSTEM SUPPLIED WITH 15 VOLTS - NETT AND GROSS MOTOR TORQUE VERSUS RMS PHASE CURRENT

loss-torque.

The gross-torque versus r.m.s. phase current characteristic obtained by combining the nett and loss-torques is shown in fig. 9.72. The gross-torque characteristic is linear and intercepts the axes at the origin. The 7 phase motor therefore produces a gross torque that is directly proportional to the r.m.s. phase current.

The equation describing the linear gross torque characteristic is:

$$T_{gross} = (0.080\ I_{rms} + 0.004)\ Nm \qquad 9.11$$

If the very small constant term is neglected, the motor torque per amp constant is simply 80mNm per r.m.s. amp.

The nett torque characteristic can be defined similarly:

$$T_{nett} = (0.085\ I_{rms} - 0.071)\ Nm \qquad 9.12$$

A predicted gross-torque versus r.m.s. phase current characteristic can be drawn by using equation 6.71:

$$T = 12.\sqrt{7/6}.\ B_{ag}.N_{ph}.I_{rms}.l_s.r_{gm} \qquad (6.71)$$

Substitution of the appropriate values for B_{ag}, N_{ph}, l_s, and r_{gm} yields:

$$T = 7.645 \times 10^{-2}.\ I_{rms}\ Nm \qquad 9.13$$

(where the value of B_{ag} is 0.298T).

The predicted motor torque characteristic is plotted with the practically determined nett and gross torque characteristics in fig. 9.73. It can be seen that the predicted torque curve is quite close to the measured gross torque curve. Hence the "Bil" torque formula is quite

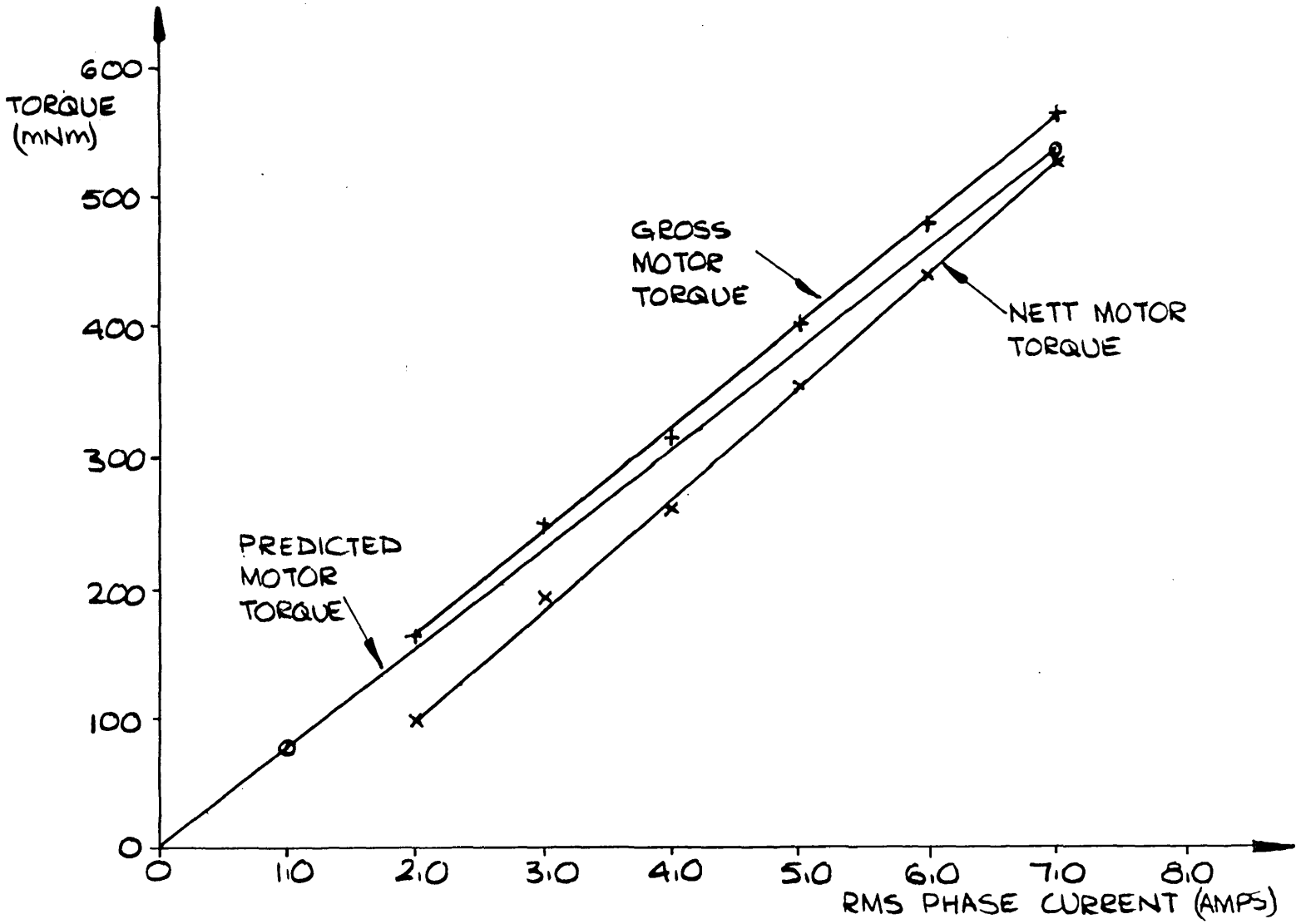


FIGURE 9.73 7 PHASE MOTOR; 0° LOAD ANGLE - COMPARISON BETWEEN MEASURED GROSS AND NETT TORQUE VERSUS PHASE CURRENT CHARACTERISTICS AND THE PREDICTED TORQUE CHARACTERISTIC

sufficient for the prediction of the gross torque output of the 7 phase motor.

It is possible to predict the maximum power output of the motor by assuming that the measured torque at a phase current of 7 amps r.m.s. would be available at the top speed of the motor (i.e. at 30000 rpm). The gross power output P_{gmax} is then:

$$P_{gmax} = (T_{gmax} \times \omega) \quad W \qquad 9.14$$

$$P_{gmax} = 0.565 \times \frac{2\pi}{60} \times 30000 \quad W$$

$$P_{gmax} = 1775 \quad W$$

Unfortunately, the large I.W.F. loss-torque would ensure that the majority of this power would be dissipated in the stator laminations. Nevertheless, a rewound motor with an improved "low-loss" stator would undoubtedly perform much better.

The current waveform was extremely good during the high torque tests. Fig. 9.74 is a photograph showing a typical set of waveforms taken under the following conditions: speed approximately 1250 rpm; nett torque 0.51Nm; phase current 6.8 amp r.m.s. The jitter on the photograph is due to the difficulty in holding the speed exactly constant when loading the motor by the string and spring balance method. The current waveform has the desired quasi-square shape, although it has to be admitted that it might not be so good at much higher motor speeds.

9.3.6 The 7 Phase Motor Starting Characteristic

In order to start, the motor has to develop sufficient torque to overcome the combined effects of the cogging torque and any externally applied load torque. No load torque was applied during the start procedure in the tests,

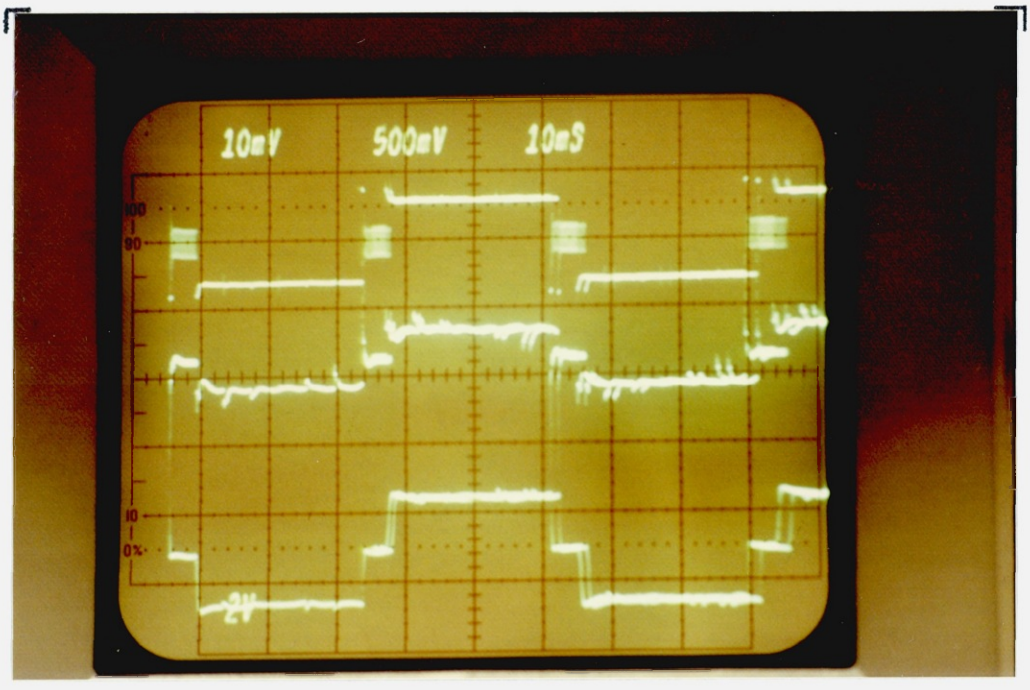


FIG. 9.74. 7 PHASE MOTOR : 0° LOAD ANGLE — APPROXIMATELY 1250 RPM ; 0·51 Nm NETT TORQUE ; 6·8 A RMS PHASE CURRENT — TOP : INVERTER OUTPUT VOLTAGE (20v/cm) — MIDDLE : MOTOR PHASE VOLTAGE (5v/cm) — BOTTOM : PHASE CURRENT (10A/cm) — 10ms/cm.

and so only cogging torque had to be overcome.

A test was conducted to determine the conditions required for the motor to overcome the cogging (load angle = 0°). It was found that the motor started every time when the inverter supply voltage was ±8 volts. It started virtually every time with an inverter supply of ±7 volts, but it refused to start on ±6 volts. These "starting" voltages (V_{start}), can be converted to equivalent starting phase currents (I_{start}), by dividing them by the total series resistance (R_{tot}) present in a phase circuit. That is, under d.c. conditions:

$$I_{start} = (V_{start}/R_{tot}) \quad \text{Amp} \qquad 9.15$$

The total series resistance per phase is 1.585Ω, made up of the resistances of a hexfet (0.4Ω), a set of protection components (1.085Ω), and a phase winding (0.1Ω).

By use of equation 9.15 it can therefore be stated that the motor always started with phase currents of 5.05 amps, it nearly always started with currents of 4.42 amps, but it failed to start with currents of 3.79 amps.

Equation 6.67 enables the actual starting torques to be calculated:

$$T_{start} = 12.B_{ag}.N_{ph}.I_{ph}.l_s.r_{gm} \qquad (6.67)$$

Substituting $B_{ag} = 0.298T$, $N_{ph} = 14.5$, $l_s = 0.05m$, $r_{gm} = 0.0273m$ yields:

$$T_{start} = 7.078 \times 10^{-2} I_{ph} \qquad 9.16$$

Combining equations 9.15 and 9.16 gives an equation for the starting torque in terms of the inverter supply voltage:

$$T_{start} = 4.466 \times 10^{-2}.V_{start} \qquad 9.17$$

The torques corresponding to the inverter supply voltages of ±8, ±7, and ±6 volts are found to be 0.357Nm, 0.313Nm, and 0.268Nm respectively. Since the measured value of cogging torque was found to be 0.271Nm (see Chapter 6), it is not surprising that the motor failed to start with ±6 volts supplied to the inverter. The starting torque available with a ±8 volt supply was obviously large enough to overcome the cogging and provide more than sufficient acceleration to the rotor.

A few general comments on the movement of the rotor during a start are appropriate in this sub-section. The rotor does not move a great deal until the stator current more or less comes into the correct spacial orientation with the rotor flux. The cogging torque is so large that correct current/flux alignment, in combination with phase currents in excess of about 5.0 amps, is required. During the time that the current pattern is stepping round to the correct axis, the rotor merely vibrates. When alignment is correct, the cogging torque is overcome and the rotor flicks round. If a SYNC interrupt occurs during the rotor movement the system becomes synchronised (i.e. the current/flux alignment is set to its correct position), and the motor runs. However, in the absence of a SYNC interrupt, the current pattern continues stepping round and eventually the rotor flicks on again. During one of these subsequent rotor movements a SYNC interrupt is detected, and normal motor operation begins.

A significant reduction in the cogging torque of the motor would be necessary before the motor could start with a load connected to it.

An attempt to record the stepping movement of the rotor during a start procedure was made. The dural disc was removed from the motor shaft to prevent the disc's inertia modifying the starting characteristic of the motor. A 10 turn potentiometer was coupled to the motor shaft by

a suitable piece of plasticine. The potentiometer was fed from a 10 volt stabilised power supply, and the slider of the potentiometer was connected to the y axis input of a y:t plotter. Any movement of the rotor was therefore recorded as a displacement in the y axis of the plotter. The timebase on the t axis was chosen so that the plotter could record all the rotor movements up to the point at which the rotor accelerates up to running speed.

Each start characteristic was obtained by holding the potentiometer in line with the motor shaft, and coupling the two with plasticine. The start procedure was then initiated (with the inverter supply voltages set to ±8 volts), and the timebase of the y:t plotter was simultaneously triggered. When the motor synchronised and accelerated, the potentiometer continued to revolve until it reached its limit of rotation, at which point the plasticine coupling sheared. An example starting characteristic is shown in fig. 9.75. The y axis (indicating rotor rotation) is not calibrated. However, the initial rotor movement and the oscillations (caused by the cogging) at the end of it, can be clearly seen. The rapid rotor acceleration is also clearly evident, as is the point at which the plasticine coupling shears.

The motor starting characteristic recorded by the y:t plotter, depends on the initial position of the SYNC line marked on the perspex disc in relation to the SYNC optical fibre. Four starting characteristics were obtained: fig. 9.76 shows the characteristic when the SYNC line is initially in line with the SYNC optical fibre - it can be seen that the motor accelerates immediately from rest; fig. 9.77 shows the characteristic when the SYNC line is initially 90° from the SYNC optical fibre - the plot shows that the rotor moved backwards and then accelerated to speed; fig. 9.78 shows the characteristic when the SYNC line is initially opposite to the SYNC optical fibre - the rotor steps forward once prior to accelerating to speed;

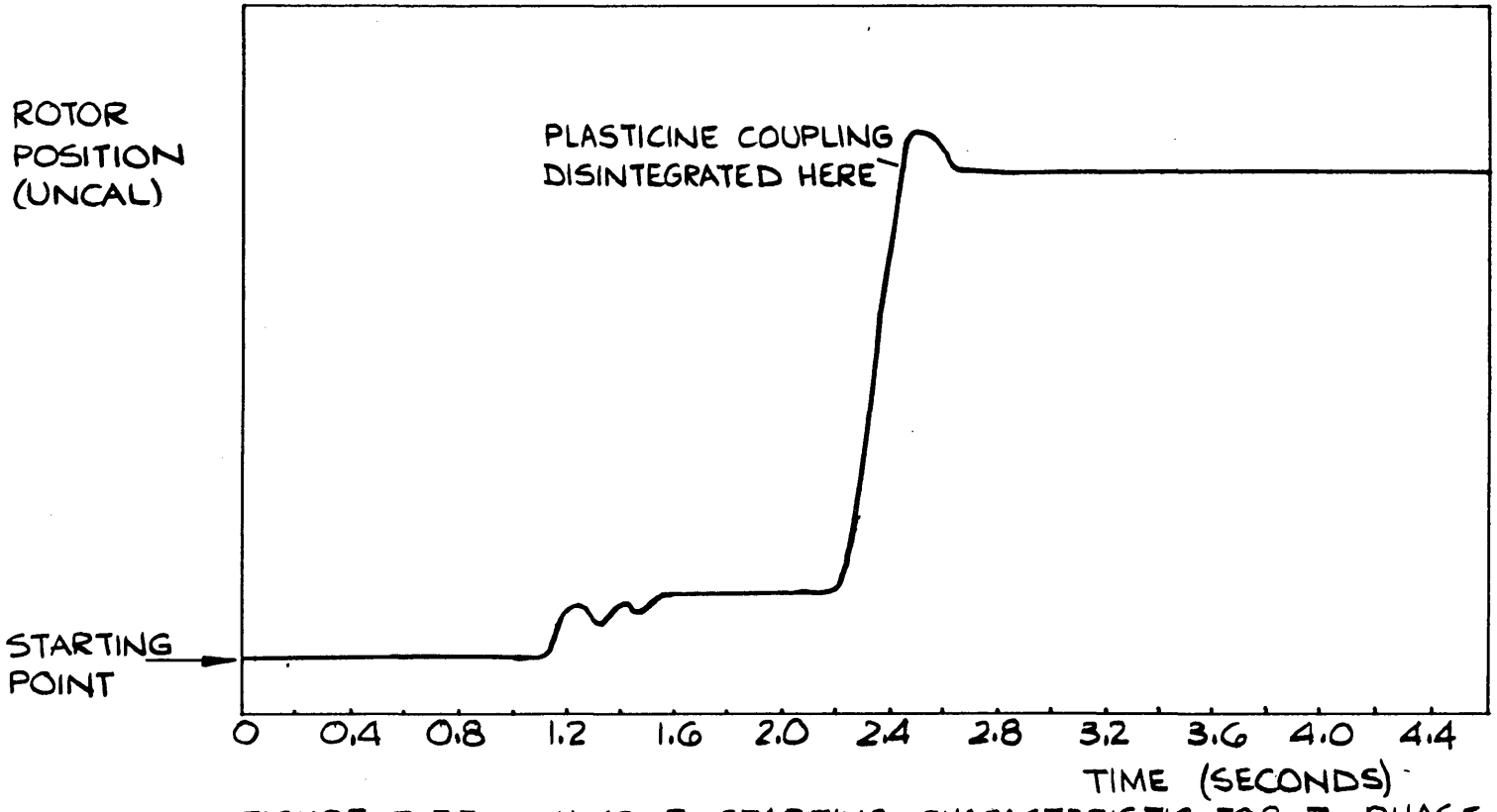


FIGURE 2.75 EXAMPLE STARTING CHARACTERISTIC FOR 7 PHASE MOTOR

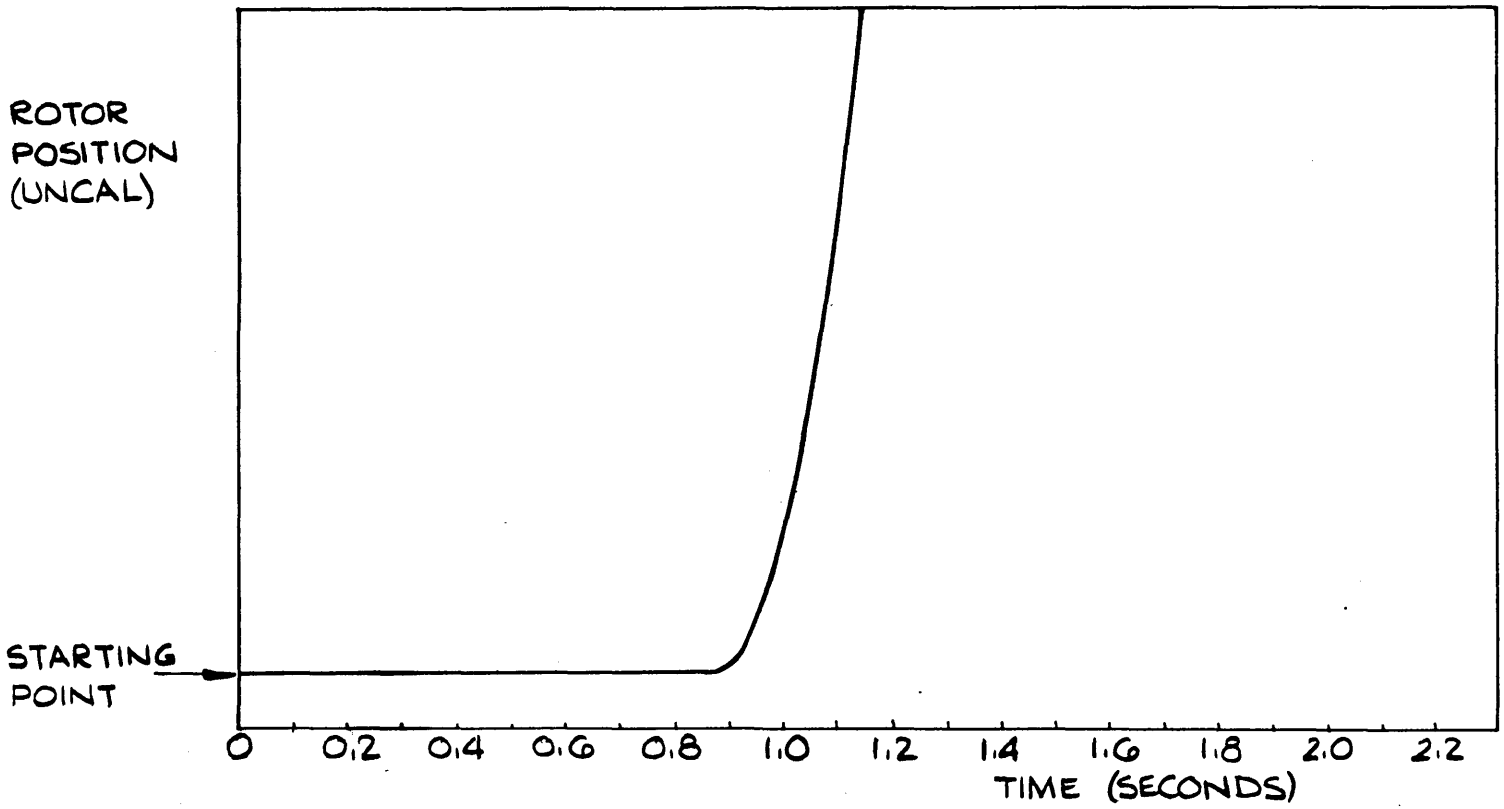


FIGURE 3.76 STARTING CHARACTERISTIC: SYNC INITIALLY UNDER OPTICAL FIBRE

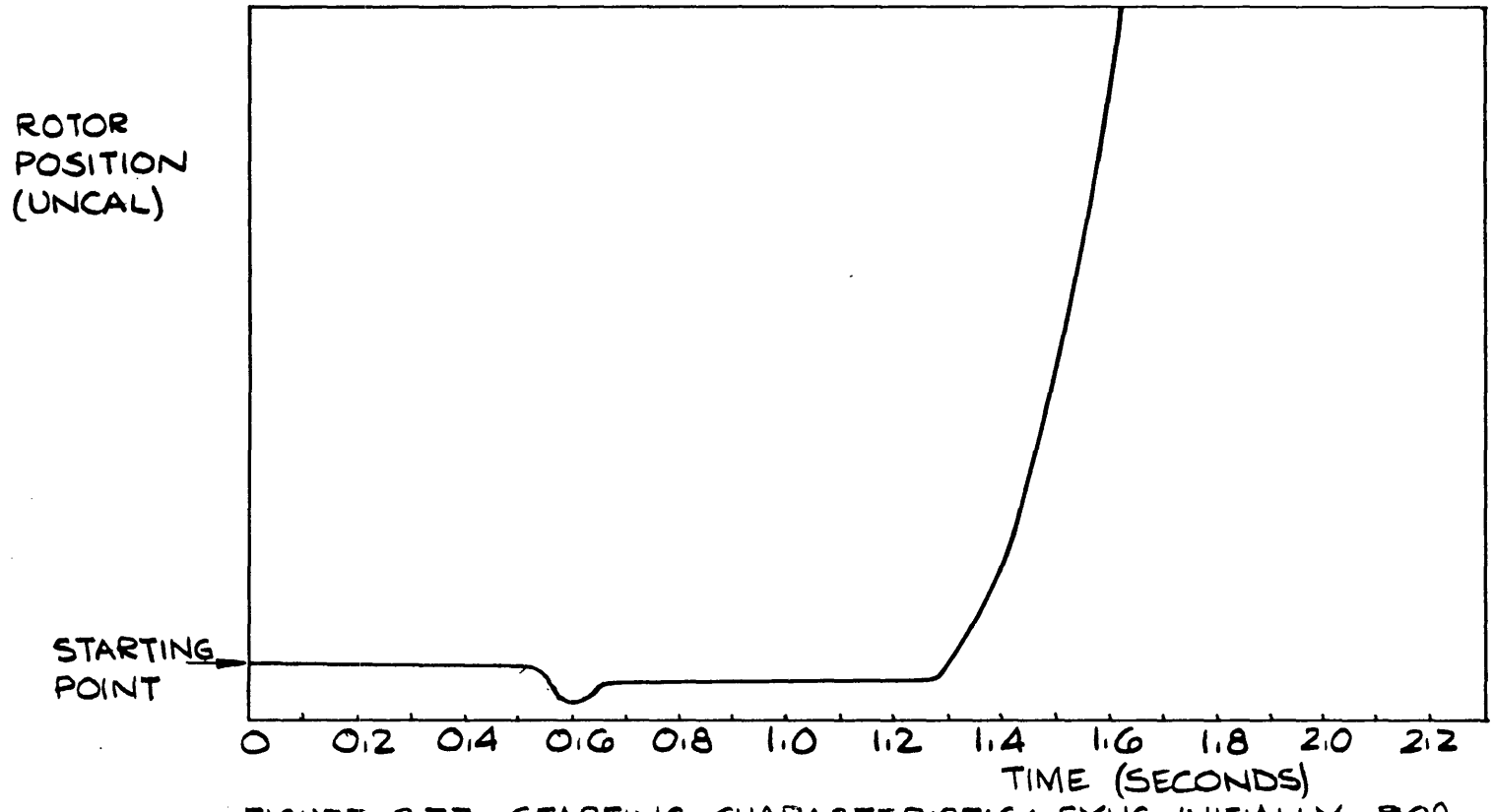


FIGURE 9.77 STARTING CHARACTERISTIC: SYNC INITIALLY 90° FROM OPTICAL FIBRE

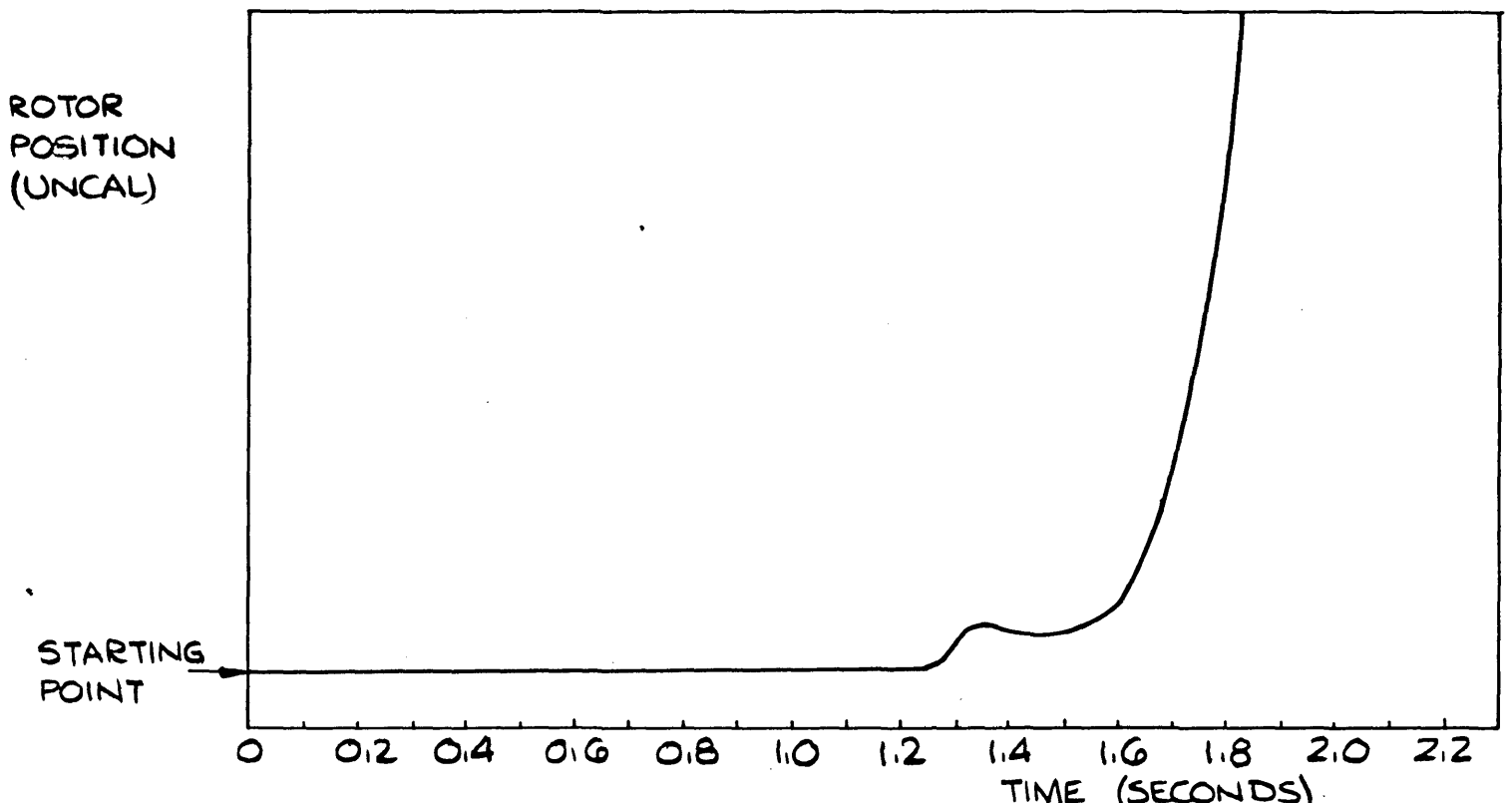


FIGURE 9.78 STARTING CHARACTERISTIC: SYNC INITIALLY OPPOSITE OPTICAL FIBRE

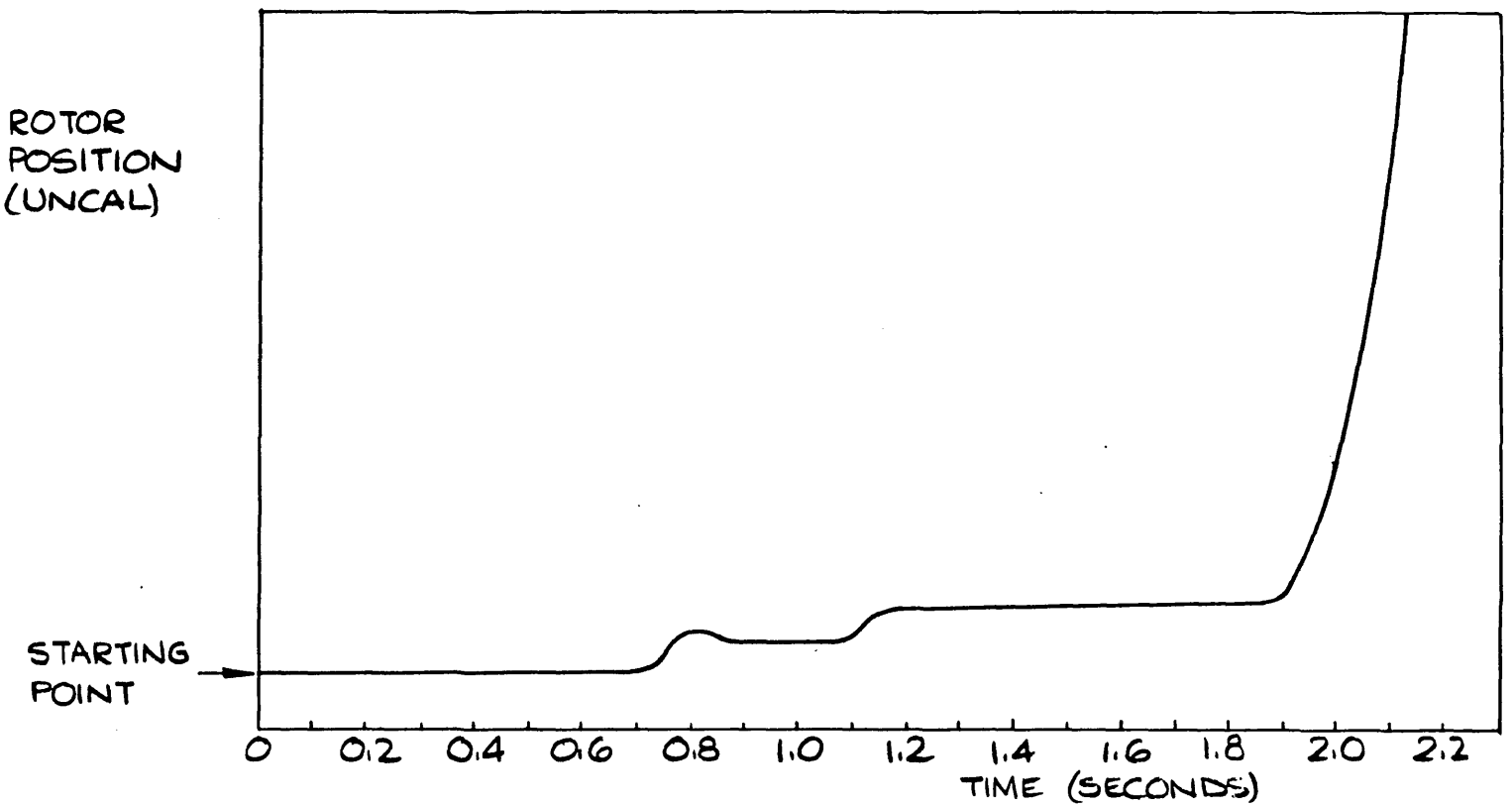


FIGURE 9.19 STARTING CHARACTERISTIC: SYNC INITIALLY 270° FROM OPTICAL FIBRE

finally, fig. 9.79 shows the characteristic when the SYNC line is initially 270° from the SYNC optical fibre - it can be seen that the rotor steps forward twice before accelerating to speed. Therefore, the maximum number of steps made by the rotor before successfully accelerating to speed during the following rotor movement is two. It should be noted that the rotor position axes in figs. 9.76, 9.77, 9.78, and 9.79 are greatly expanded, and in no way cover ten rotor revolutions.

9.3.7 General Comments on the High-Speed No-Load Motor Performance

At motor speeds in excess of 15400 rpm it was found that the eddy current brake assembly tended, unpredictably, to start vibrating violently. Therefore, in order to run the motor at speeds much greater than 15400 rpm, it was necessary to remove the eddy current brake assembly from the dynamometer rig. The dural disc mounted on the motor shaft was also removed. These changes substantially reduced the vibration of the rig and as a consequence the motor was much quieter, especially at speeds above 10000 rpm.

Having removed the eddy current brake, the motor was successfully operated in an unloaded state at speeds up to 25440 rpm. During the initial high speed runs it was noticed that there was some jitter in the phase current waveform whenever the motor speed exceeded 16400 rpm. Fig. 9.80 is a photograph illustrating the current jitter at a speed of 16430 rpm. The jitter was found to be due to missed SEG interrupts, caused by the incorrect orientation of the perspex position sensor disc on the rotor shaft. A full explanation for the jitter is given in Chapter 4, section 4.8.6. Once the disc orientation had been corrected, the motor ran at speeds up to about 22250 rpm without jitter, and fig. 9.81 shows the waveforms at this speed.

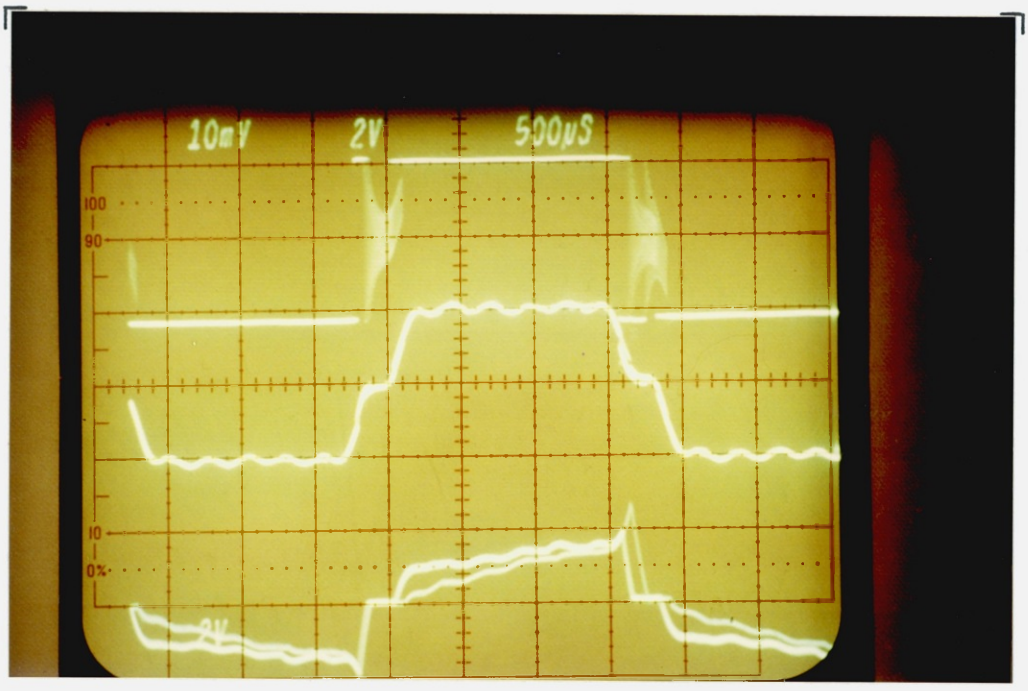


FIG.9.80. 7 PHASE MOTOR (UNLOADED) : 0° LOAD ANGLE — ±22 VOLTS INVERTER SUPPLY, 16430 RPM — TOP : INVERTER OUTPUT VOLTAGE (20V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (20V/cm) — BOTTOM : PHASE CURRENT (1A/cm) — 500μs/cm.

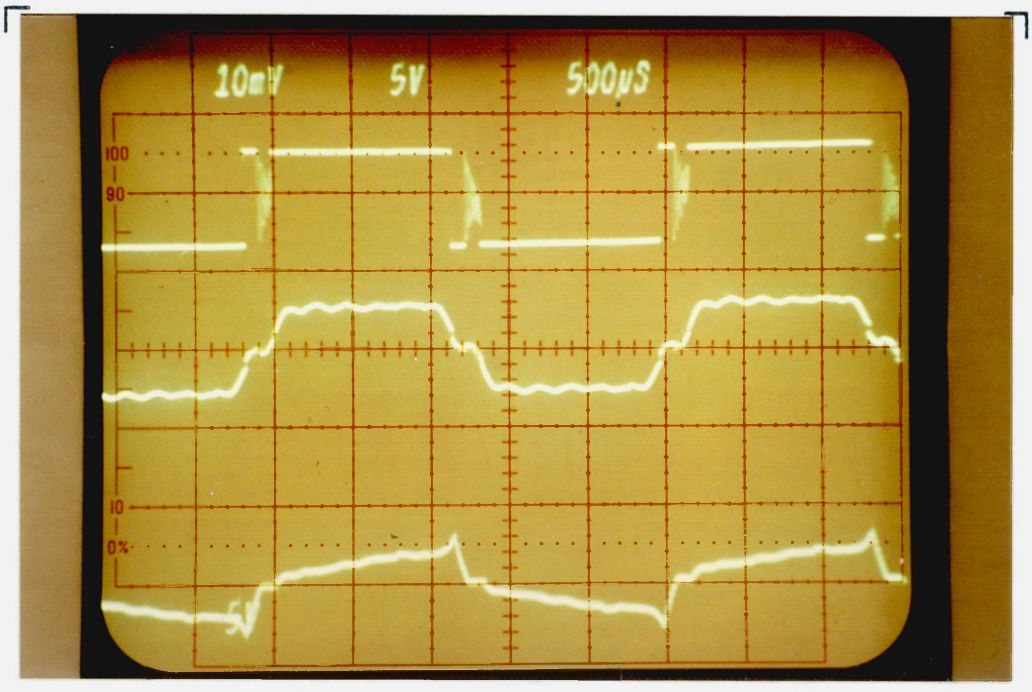


FIG.9.81. 7 PHASE MOTOR (UNLOADED) : 0° LOAD ANGLE — ±30 VOLTS INVERTER SUPPLY, 22250 RPM — TOP : INVERTER OUTPUT VOLTAGE (50V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (50V/cm) — BOTTOM : PHASE CURRENT (5A/cm) — 500μs/cm.

At speeds above 22250 rpm the jitter on the current waveform reappeared, and it got very bad as the speed approached 25000 rpm. The jitter at these speeds was due to software timing problems. Ways of overcoming the timing problems are discussed in Chapter 4, section 4.9. Photographs of typical waveforms are shown in figs. 9.82 and 9.83. Fig. 9.82 shows a remarkably jitter-free set of waveforms for the motor operating at 24060 rpm with a 0° load angle and an inverter supply voltage of ± 33 volts: however, the current waveform is far from the desired quasi-square shape. Fig. 9.83 shows the waveforms for the motor operating at 25440 with a 6° load angle and an inverter supply voltage of ± 33 volts: the improvement in the current waveform shape due to the 6° load angle can be clearly seen, but unfortunately the increase in speed resulting from the improved current waveform is accompanied by continuous current jitter (captured very clearly on the photograph).

An attempt was made to achieve a speed in excess of 25000 rpm with the load angle set to 0°. The current jitter became very bad and it appeared that a SEG interrupt was being missed on virtually every revolution. This had the effect of setting up a -6° load angle instead of the desired 0°, and as a consequence there were very large current spikes at the trailing edges of the phase current waveform as indicated in fig. 9.84. The spikes had peak values of at least 10 amps (perhaps even greater than 15 amps), and the system did not run for more than a few seconds before a hexfet failed.

The damaged hexfet was removed from its circuit board and a resistance check between its terminals was performed. It was found that the drain was short circuited to the gate, whilst the source appeared to be open circuit. The case of the hexfet was sawn open and it was observed that the connecting wire from the source terminal to the hexfet silicon chip had broken, presumably having fused due to

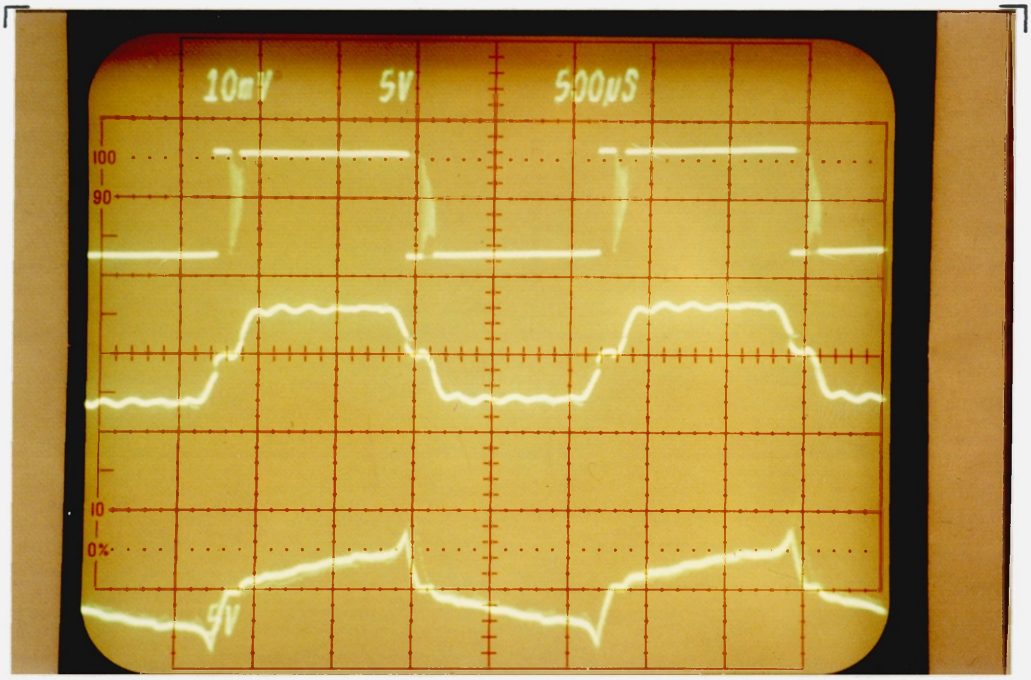


FIG. 9.82. 7 PHASE MOTOR (UNLOADED): 0° LOAD ANGLE — ±33 VOLTS INVERTER SUPPLY, 24060 RPM — TOP: INVERTER OUTPUT VOLTAGE (50V/cm) — MIDDLE: MOTOR PHASE VOLTAGE (50V/cm) — BOTTOM: PHASE CURRENT (5A/cm) — 500μs/cm.

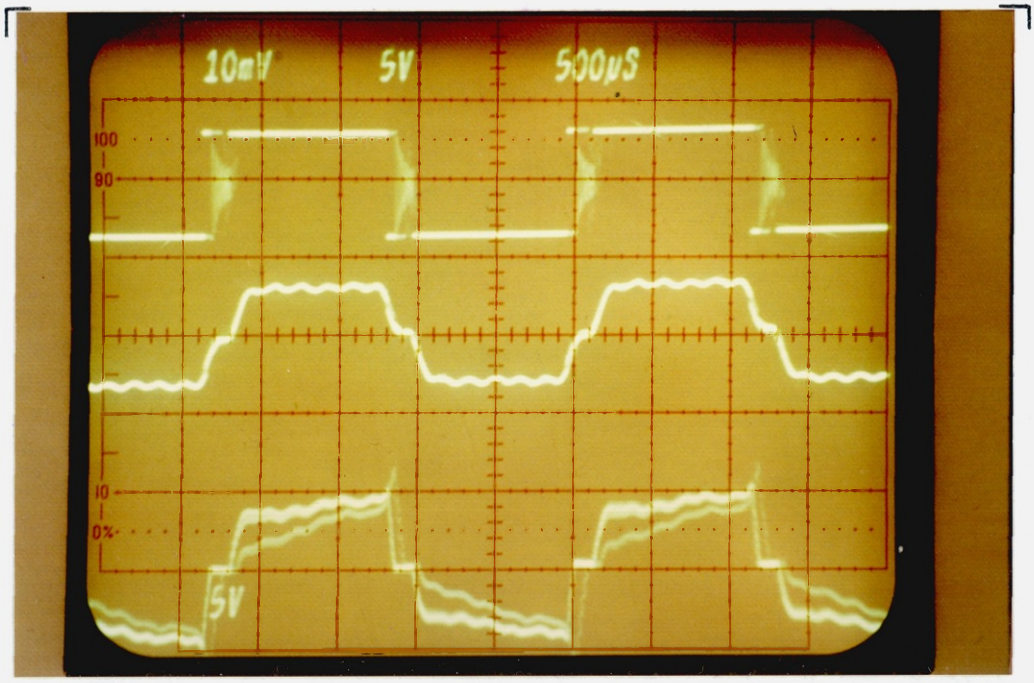


FIG. 9.83. 7 PHASE MOTOR (UNLOADED): 6° LOAD ANGLE — ±33 VOLTS INVERTER SUPPLY, 25440 RPM — TOP: INVERTER OUTPUT VOLTAGE (50V/cm) — MIDDLE: MOTOR PHASE VOLTAGE (50V/cm) — BOTTOM: PHASE CURRENT (2A/cm) — 500μs/cm.

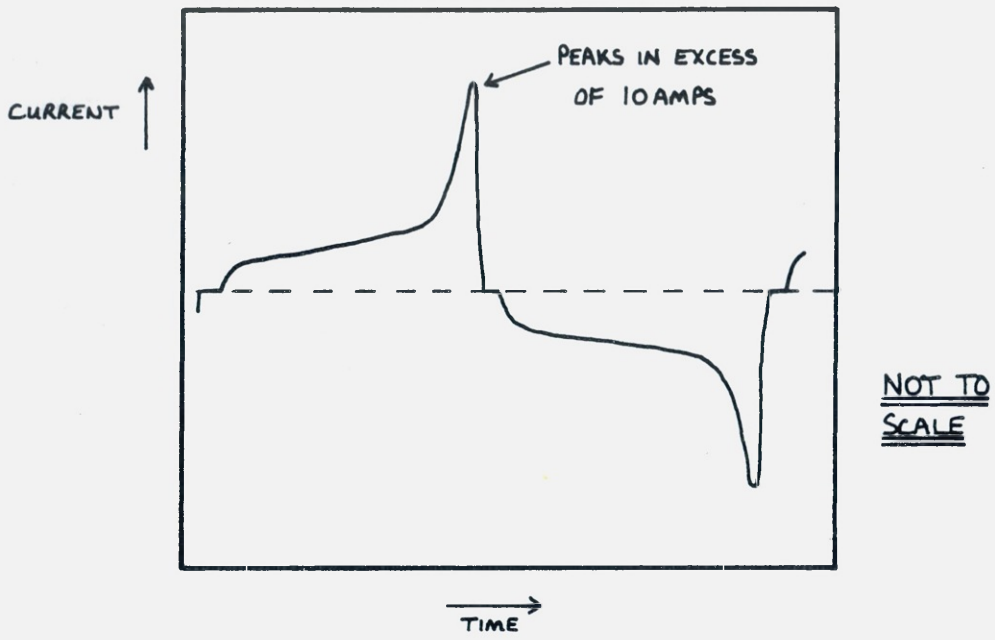


FIG. 9.84. SKETCH OF THE CURRENT WAVESHAPE OCCURRING FOR 0° LOAD ANGLE OPERATING CONDITIONS AT SPEEDS ABOVE 25000 RPM.

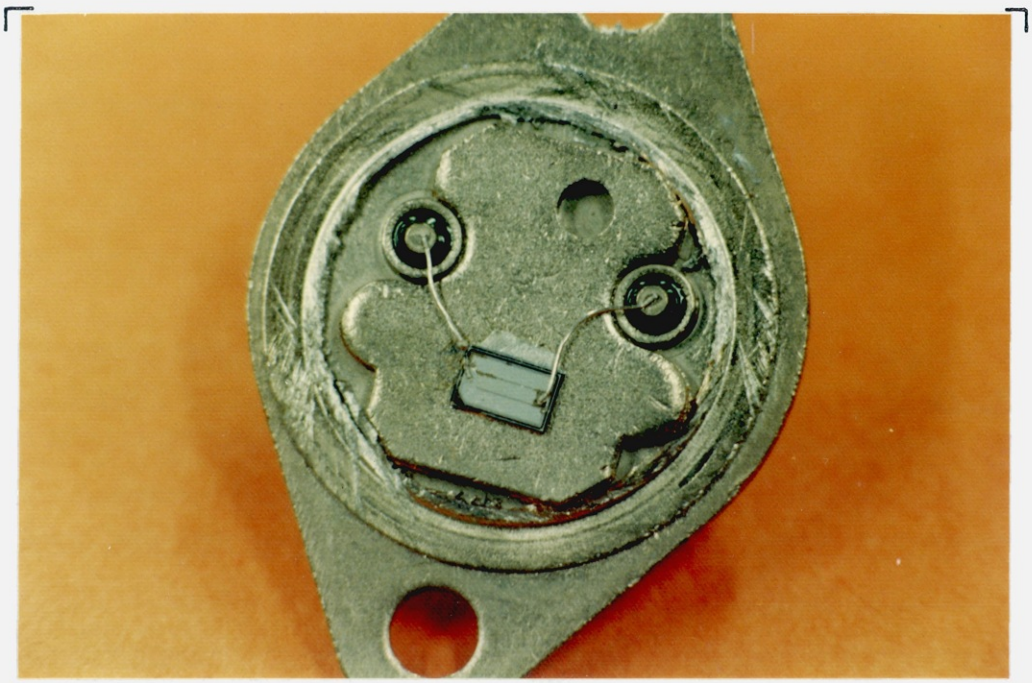


FIG. 9.85. VIEW OF THE INSIDE OF THE "FUSED" HEXFET.

the high peaks in the drain current. Fig. 9.85 is a photograph showing the inside of the damaged hexfet case: the left hand connecting wire is broken just at the point where it connects to the silicon chip. Discussions with International Rectifier have subsequently revealed that the diameter of the connecting wires has been increased since the batch of hexfets used for the 7 phase inverter were manufactured. It is therefore most probable that the hexfet simply failed due to the connecting wire fusing.

While the damaged hexfet was being replaced, the 7 phase motor was successfully run with only 6 hexfet inverter boards connected in the inverter. The 6 "active" phases were able to self-start the motor. In addition the motor appeared to run just as smoothly as with the normal complement of 7 "active" phases. This suggests that it might well be possible to design a multi-phase motor drive system so that when lightly loaded it would run on a suitable fraction of the total number of phases, the number of "active" phases being appropriately increased to cope with applied load. Reducing the number of active phases in this manner could result in worthwhile reductions in the inverter switching losses for example.

Having explained why a speed of 25000 rpm with a load angle of 0° caused damage to the inverter, it is worthwhile commenting on the actual operating conditions that occurred during the previously mentioned successful motor run at 25440 rpm with a 6° load angle (see fig. 9.83). The bad current jitter was a result of the system frequently missing SEG interrupts, but large current spikes did not occur because the effect of missing a SEG interrupt during a revolution was only to change the desired 6° load angle to 0°. Given the good match between the inverter output voltage waveform and the motor back-emf waveform, it is unlikely that dangerously large current spikes can occur at the trailing edges of the current waveform when the load angle has a 0° value. Therefore, despite the current wave-

form jitter, the motor ran reliably at 25440 rpm with no risk to the inverter. The lesson to be learned from the "fused hexfet" incident is that negative load angles, however caused, should be avoided in the square wave motor because they can result in current peaks at the trailing edge of the current waveforms: these peaks are undesirable because of the increased possibility of inverter damage and also because the phase current is required to commutate immediately following them.

As previously mentioned, speeds in excess of about 15400 rpm could only be safely achieved with the eddy current brake assembly removed from the test rig. Hence all test readings taken during the "high" speed motor runs were for no-load conditions. The motor speed, phase current, motor power input, and inverter supply voltage readings were recorded: they have been used to produce a speed versus inverter voltage characteristic and a speed versus phase voltage characteristic (see sub-section 9.3.8); in addition the results have been used to check the accuracy of the loss-torque characteristic presented in fig. 9.14 (see section 9.5).

Typical no-load readings at a motor speed of 25440 rpm with a 6° load angle were:

Inverter Supply Voltages:	±33 Volts d.c.
Positive Rail Inverter Supply Current:	4.20 Amps d.c.
Negative Rail Inverter Supply Current:	-4.20 Amps d.c.
Total Input Power to Inverter:	277.2 Watts
Total Inverter Output Power:	266.0 Watts
Inverter Efficiency:	96%
Inverter Output Phase Voltage:	31.32 Volts r.m.s.
Total Input Power to Motor:	252.0 Watts
Motor Phase Voltage:	26.90 Volts r.m.s.
Motor Phase Current:	1.370 Amps r.m.s.

No comment on the gross motor efficiency can be made: the

loss-torque characteristic (fig. 9.14) predicts a motor power output of 289 Watts at 25440 rpm, and this unfortunately is larger than the measured power input to the motor. This confirms the view, expressed in sub-section 9.3.3, that the loss-torque characteristics (figs. 9.13 and 9.14) are "too large".

It was very evident after a five minute motor run, at a speed of 25440 rpm, that the bulk of the input power to the motor was being dissipated in the form of heat in the stator. The temperature of the motor case was 60°C, whilst the temperature of the rotor shaft at the optical sensor end was 57°C. The temperature at the "load" end of the rotor shaft was 39°C. It is therefore possible that the temperature in the magnet area of the rotor could have exceeded 60°C, but checks on the motor back-emf showed that no demagnetisation had occurred. The temperature measurements certainly reinforce the view that an eddy-current-resistant stator material and no lamination burrs are required for the square-wave motor.

9.3.8 The Speed Versus Voltage Characteristics of the 7 Phase Motor System

From the motor performance characteristics presented in sub-section 9.3.3, it has been possible to produce a set of characteristics for motor speed versus inverter supply voltage. The no-load speeds for load angles of 0° and 6° are shown plotted against inverter supply voltage in fig. 9.86. It should be noted that the dural disc was attached to the rotor shaft when these characteristics were obtained. The characteristics indicate that the no-load speed is approximately proportional to the inverter supply voltage. In addition, for a given inverter supply voltage, the motor runs slightly faster with a 6° load angle than with a 0° load angle. A 140mNm gross torque load line for a 0° load angle is also shown in fig. 9.86. This characteristic shows the speed versus inverter supply voltage

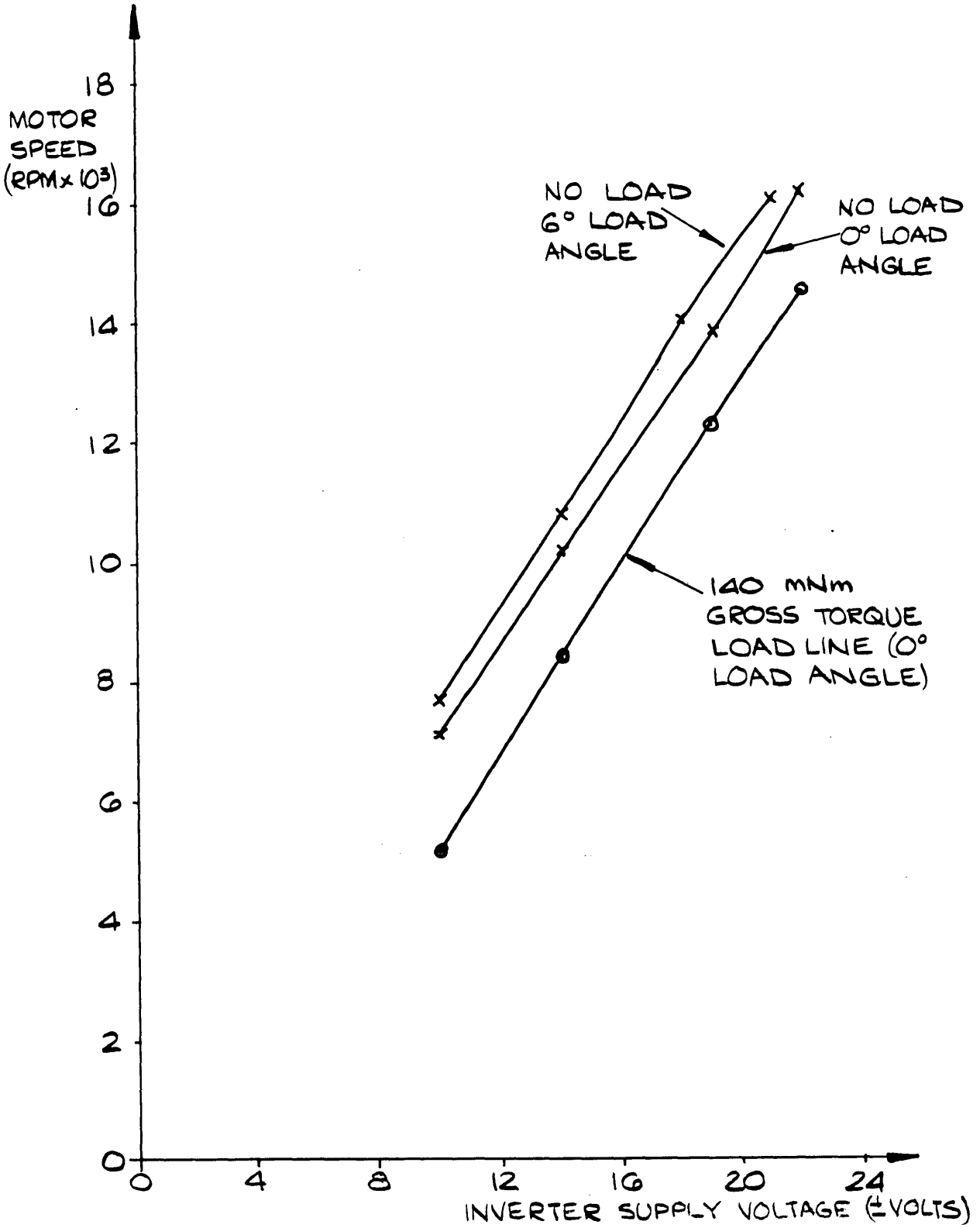


FIGURE 3.86 7 PHASE MOTOR: NO-LOAD SPEED VERSUS INVERTER SUPPLY VOLTAGE FOR 0° AND 6° LOAD ANGLES; 140 mNm GROSS TORQUE LOAD LINE

relationship when the motor is driving a constant torque load. The characteristic is more or less linear, and can be represented in equation form:

$$S = (783\ V_{inv} - 2588)\ \text{rpm} \qquad 9.18$$

where S is the 140mNm "on load" motor speed, and V_{inv} is the magnitude of the inverter supply voltage.

The linear nature of the characteristics shown in fig. 9.86 would make any control-loop software substantially simpler.

Speed versus inverter supply voltage characteristics, for speeds in the range between 15000 rpm and 24000 rpm, have been constructed from the no-load results obtained during the high speed motor runs described in sub-section 9.3.7. The no-load motor speeds for load angles of 0° and 6° are plotted against the inverter supply voltage in fig. 9.87.

A point of great interest with regard to the characteristics is that the speed of 25440 rpm, achieved with a 6° load angle and ± 33 volts supplied to the inverter, actually lies along a line projected on from the 0° characteristic. This is not due to bad experimental technique: it actually confirms what is said in sub-section 9.3.7 about the effects of missed SEG interrupts. At 25440 rpm the software timings are such that a SEG interrupt is missed on virtually every revolution. This has the effect of changing the desired 6° load angle into a working load angle of 0°, and so the fact that the 25440 rpm, 33 volt, 6° point lies along the 0° load angle characteristic is entirely logical. The point has not therefore been plotted as part of the 6° load angle characteristic because it really belongs to the 0° set of results. The current jitter due to missed SEG interrupts was not sufficiently bad at any of the other plotted speed points to affect

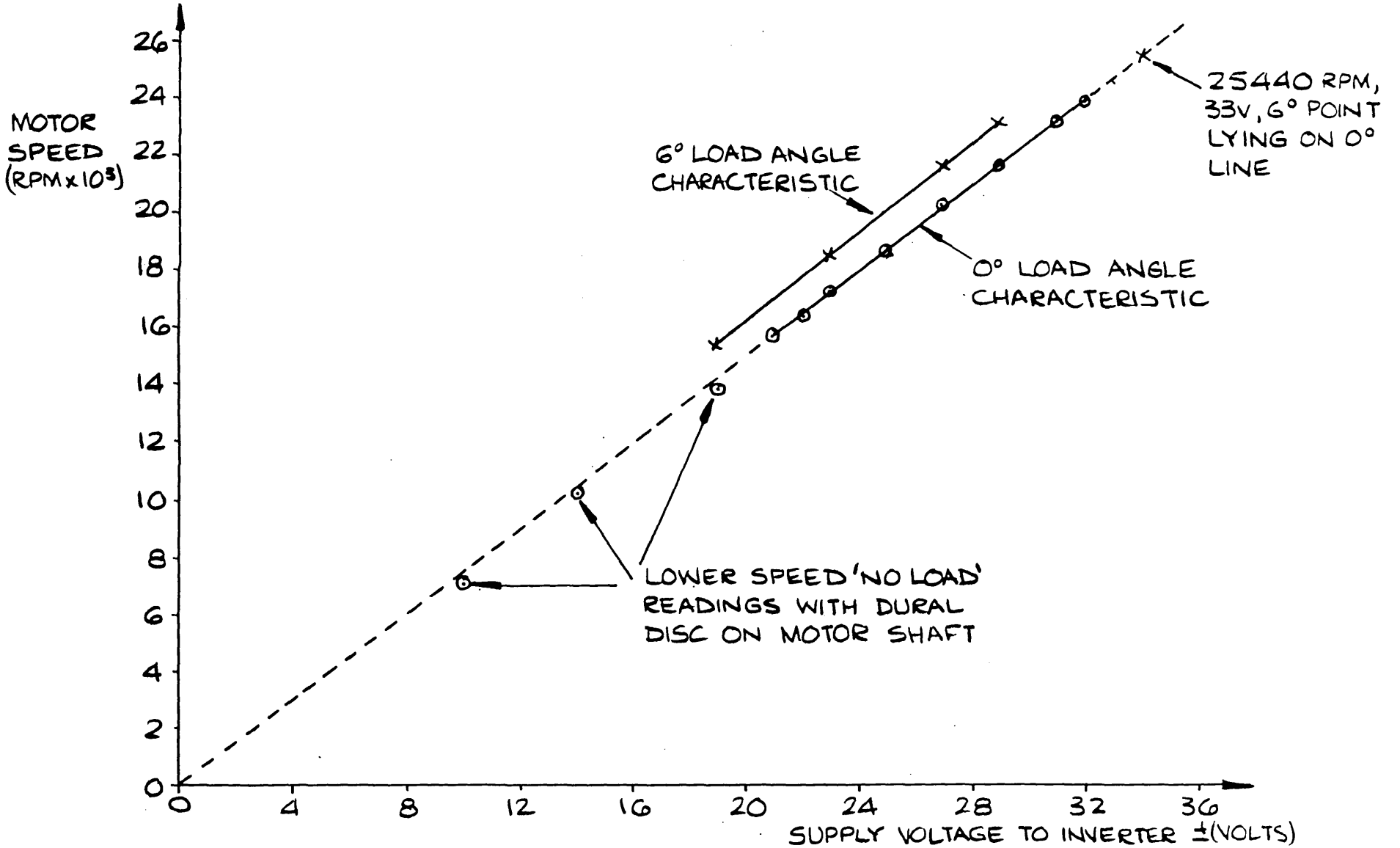


FIGURE 9.87 7 PHASE MOTOR: 'HIGH SPEED' NO LOAD SPEED VERSUS VOLTAGE APPLIED TO INVERTER CHARACTERISTICS FOR 0° AND 6° LOAD ANGLES (NO DURAL DISC ON MOTOR SHAFT)

their load angle classification.

Returning to a general discussion about the results shown in fig. 9.87, it can be stated that both the 0° and 6° characteristics are linear: the 0° characteristic intersects the axes at the origin. For a given inverter supply voltage the motor runs slightly faster with a 6° load angle than with a 0° load angle. This agrees with the "low speed" characteristics shown in fig. 9.86. The characteristics shown in fig. 9.87 were obtained with the dural disc removed from the rotor shaft: there was, therefore, a slightly smaller windage torque suffered by the rotor during the "high speed" tests than during the earlier "low speed" tests. This fact is confirmed by the three "low speed" no-load 0° load angle speed readings that are plotted on fig. 9.87. The readings all lie just below the line which is projected back from the high speed 0° load angle characteristic to the origin.

The characteristics shown in fig. 9.87 can be represented in equation form. The 0° load angle no-load speed versus inverter supply voltage equation is:

$$S = (742\ V_{inv} + 105) \quad \text{rpm} \qquad 9.19$$

where S and V_{inv} are as previously defined. The 6° load angle no-load speed versus supply voltage equation is:

$$S = (776\ V_{inv} + 592) \quad \text{rpm} \qquad 9.20$$

The speed versus voltage characteristics presented in figs. 9.86 and 9.87 include the effects of the voltage drops in the protection components. The relationship between the motor phase voltage and the motor speed, is a more accurate indication of the rpm-per-volt characteristic of the motor. The no-load motor speeds for load angles of 0° and 6° are plotted against the r.m.s. phase voltage in fig. 9.88. Both characteristics are linear and they

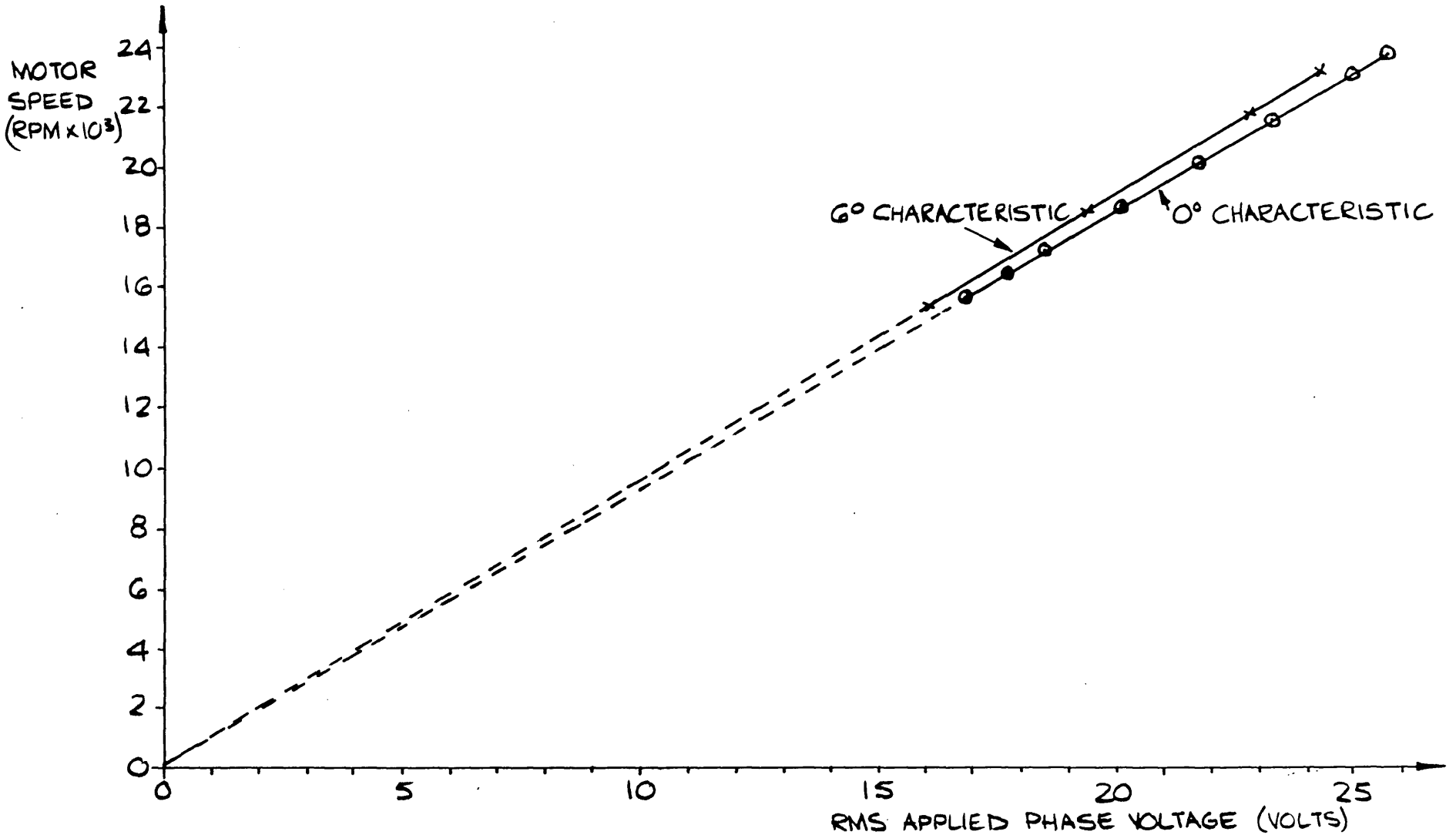


FIGURE 5.88 7 PHASE MOTOR: NO LOAD SPEED VERSUS APPLIED PHASE VOLTAGE (RMS) FOR 0° AND 6° LOAD ANGLE OPERATING CONDITIONS (NO DURAL DISC ON MOTOR SHAFT)

both intercept the axes at the origin. The equation describing the 0° load angle speed versus r.m.s. phase voltage characteristic is:

$$S = (930\ V_{rms} - 57) \quad \text{rpm} \qquad 9.21$$

where S is the motor speed and V_{rms} is the r.m.s. phase voltage (as measured at the motor phase terminals). Neglecting the small constant term, the rpm per r.m.s. volt constant is 930 for a 0° load angle.

The equation describing the 6° load angle speed versus r.m.s. phase voltage characteristic is:

$$S = (942\ V_{rms} + 128) \quad \text{rpm} \qquad 9.22$$

Hence, neglecting the constant term gives an rpm per r.m.s. constant of 942 for a 6° load angle.

To conclude the work on the speed versus voltage results it was decided to investigate the relationship between the motor phase winding back-emf and the motor speed. The back-emf could not be measured directly, and so it was derived from the phase voltages measured at the motor terminals by taking into account the small resistive voltage drops in the winding (neglecting any inductive reactance effects). The r.m.s. back-emf (E_{rms}) is given by:

$$E_{rms} = (V_{rms} - I_{rms}.R_{ph}) \quad \text{volts} \qquad 9.23$$

where V_{rms} is the r.m.s. phase voltage, I_{rms} is the r.m.s. phase current, and R_{ph} is the phase winding resistance.

The r.m.s. back-emf is related to the peak value (E_{ph}) of the quasi-square back-emf waveform by the equation:

$$E_{ph} = \sqrt{\frac{7}{6}}\ E_{rms} \qquad 9.24$$

Combining equations 9.24 and 9.23 gives:

$$E_{ph} = \sqrt{\frac{7}{6}} \; (V_{rms} - I_{rms} \cdot R_{ph}) \quad \text{volts} \qquad 9.25$$

The constant k_{rpm} relating the motor speed S to the back-emf E_{ph} is given by:

$$k_{rpm} = (S/E_{ph}) \quad \text{rpm per volt} \qquad 9.26$$

In addition, the back-emf constant, k_b, of the motor is calculated from the equation:

$$k_b = (E_{ph}/\omega) \quad \text{volts per radian per second} \qquad 9.27$$

where ω is the angular velocity at which the back-emf E_{ph} is generated.

Equations 9.25, 9.26, and 9.27 were used to calculate values of k_{rpm} and k_b for various operating conditions of the 7 phase motor. The results are presented in tabular form.

Fig. 9.89 is the table of k_{rpm} and k_b values for the motor operating under load with a 0° load angle and with an inverter supply voltage of ±10 volts. The average values of k_{rpm} and k_b are 866.5 rpm per volt and 11.02 mV per radian per second respectively.

Fig. 9.90 is the table of k_{rpm} and k_b values for the motor operating under load with a 6° load angle and with an inverter supply voltage of ±10 volts. The average values of k_{rpm} and k_b are 895.5 rpm per volt and 10.73 mV per radian per second respectively. Hence the average value of k_{rpm} with a load angle of 6° at ±10 volts is slightly larger (3.3%) than the average value with a load angle of 0°: the corresponding average value of k_b at 6° is slightly smaller (2.6%) than the average value at 0°. It therefore seems reasonable to state that the load angle of the motor

S	V_{rms}	I_{rms}	E_{ph}	K_{rpm}	K_b
MOTOR SPEED (RPM)	RMS PHASE VOLTAGE (VOLTS)	RMS PHASE CURRENT (AMPS)	BACK EMF (VOLTS)	SPEED PER VOLT (RPM/VOLT)	BACK EMF CONSTANT (mV/RAD/S)
7148	7·70	0·870	8·22	869·6	10·98
6823	7·39	1·066	7·87	867·0	11·01
6523	7·09	1·220	7·53	866·3	11·02
6171	6·74	1·410	7·13	865·5	11·03
5777	6·33	1·640	6·66	867·4	11·01
5211	5·78	1·950	6·03	864·2	11·05
4629	5·18	2·280	5·35	865·2	11·04

FIG. 9.89. TABLE OF K_{rpm} AND K_b VALUES FOR THE MOTOR OPERATING UNDER LOAD AT 0° LOAD ANGLE WITH ±10 VOLTS SUPPLIED TO THE INVERTER.

has some control over the k_{rpm} and k_b constants.

The values of constants k_{rpm} and k_b do not change to any great extent as the inverter supply voltage and motor operating speed are increased. Fig. 9.91 is the table of k_{rpm} and k_b values for the motor operating under load with a 6° load angle and with an inverter supply voltage of ±21 volts. The average values of k_{rpm} and k_b are 882.7 rpm per volt and 10.82 mV per radian per second respectively. These average values of k_{rpm} and k_b differ from the average values at 6°, ±10 volts by only -1.4% and 0.8% respectively.

Further proof of the stability in the values of k_{rpm}

S	V_{rms}	I_{rms}	E_{ph}	K_{rms}	K_b
MOTOR SPEED (RPM)	RMS PHASE VOLTAGE (VOLTS)	RMS PHASE CURRENT (AMPS)	BACK EMF (VOLTS)	SPEED PER VOLT (RPM/VOLT)	BACK EMF CONSTANT (mV/RAD/S)
7731	8·11	0·875	8·67	891·7	10·71
7423	7·80	1·008	8·32	892·2	10·70
7129	7·50	1·138	7·98	893·4	10·69
6642	7·01	1·360	7·42	895·1	10·67
6248	6·62	1·557	6·98	895·1	10·67
5718	6·09	1·830	6·38	896·2	10·65
4524	4·88	2·510	5·00	904·8	11·05

FIG. 9.90 TABLE OF K_{rpm} AND K_b VALUES FOR THE MOTOR OPERATING ON LOAD AT 6° LOAD ANGLE WITH ±10 VOLTS SUPPLIED TO THE INVERTER.

and k_b is given in fig. 9.92. Fig. 9.92 is a table of k_{rpm} and k_b values for the motor operating on no-load with a 0° load angle and with inverter supply voltages in the range ±21.0 volts to ±32.0 volts. It can be seen that the values of k_{rpm} and k_b change very little over the speed range 15670 rpm to 23860 rpm. The average values of k_{rpm} and k_b are 863.9 rpm per volt and 11.05 mV per radian per second respectively. These average values of k_{rpm} and k_b differ from the average values at 0°, ±10 volts by only -0.3% and 0.3% respectively.

Finally, it is interesting to compare the open circuit value of k_b measured before any motor operations, and the

S	V_{rms}	I_{rms}	E_{ph}	k_{rms}	k_b
MOTOR SPEED (RPM)	RMS PHASE VOLTAGE (VOLTS)	RMS PHASE CURRENT (AMPS)	BACK EMF (VOLTS)	SPEED PER VOLT (RPM/VOLT)	BACK EMF CONSTANT (mV/RAD/S)
16097	17·03	1·300	18·25	882·0	10·83
15934	16·83	1·385	18·03	883·7	10·81
15651	16·54	1·480	17·71	883·7	10·81
15274	16·16	1·630	17·28	883·9	10·80
14863	15·77	1·810	16·84	882·6	10·82
14717	15·62	1·880	16·67	882·8	10·82
14400	15·31	2·025	16·32	882·4	10·82
12917	13·86	2·765	14·67	880·5	10·85

FIG. 9.91. TABLE OF K_{rpm} AND K_b VALUES FOR THE MOTOR OPERATING UNDER LOAD AT 6° LOAD ANGLE WITH ±21 VOLTS SUPPLIED TO THE INVERTER.

average value of k_b for all the results presented in figs. 9.89 to 9.92 inclusive. The initial measured value of k_b was 11.80 mV per radian per second. The averaged value of k_b from the values given in figs. 9.89 to 9.92 is 10.91 mV per radian per second. The initial and averaged values therefore differ by only about 8%. This difference can be attributed to the errors inherent in the methods used to calculate the values of k_b. Each value of k_b was calculated from the "peak" phase voltage: in the case of the initial pre-motor-run value, the peak phase voltage was calculated from the average phase voltage by using a 7/6

V_{inv}	S	V_{rms}	I_{rms}	E_{ph}	K_{rms}	K_b
INVERTER SUPPLY VOLTAGE (±VOLTS)	MOTOR SPEED (RPM)	RMS PHASE VOLTAGE (VOLTS)	RMS PHASE CURRENT (AMPS)	BACK EMF (VOLTS)	SPEED PER VOLT (RPM/VOLT)	BACK EMF CONSTANT (mV/RAD/S)
21·0	15670	16·88	1·090	18·11	865·3	11·04
22·0	16430	17·74	1·125	19·04	862·9	11·07
23·0	17220	18·56	1·130	19·93	864·0	11·05
25·0	18640	20·11	1·190	21·59	863·4	11·06
27·0	20180	21·77	1·250	23·38	863·1	11·06
29·0	21580	23·26	1·300	24·98	863·9	11·05
31·0	23130	24·95	1·380	26·80	863·1	11·06
32·0	23860	25·67	1·400	27·58	865·1	11·04

FIG. 9.92. TABLE OF K_{rpm} AND K_b VALUES FOR THE MOTOR OPERATING ON NO LOAD AT 0° LOAD ANGLE WITH VARIOUS INVERTER SUPPLY VOLTAGE VALUES.

multiplying constant: in the case of the averaged "motor-run" value, the peak phase voltage was calculated from the r.m.s. phase voltage by using a $(7/6)^{\frac{1}{2}}$ multiplying constant. In both cases the multiplying constants were based on the assumption that the phase voltage waveform was of the desired quasi-square wave shape. Since this was not strictly true, it is not surprising that there is an 8% difference between the pre-run open circuit value of k_b and the averaged motor-run value of k_b.

It is concluded from the values of k_b that no signif-

icant demagnetisation of the rotor magnets occurred during the motor operating tests. It can also be seen that the rotor flux is not modified as the phase current increases since the value of k_b does not change (for example, see fig. 9.91). The large effective stator-rotor airgap in combination with the high coercivity polymer bonded rare earth magnets therefore appear to produce the desired operating conditions: that is, constant rotor flux.

9.3.9 The Attempt to Perform "High" Speed Load Tests Using the String/Spring Balance Load Technique

By using the string and spring balance "dynamometer" it was possible to load the motor up to its maximum torque capability. Unfortunately the speeds at which this could be achieved were rather low, in order to prevent the string rapidly catching fire due to the friction between it and the rotor-mounted pulley. In general, it was found that full motor torque at a speed in excess of about 2000 rpm could only be maintained for a maximum of 10 seconds before the string broke. It was very difficult to take a full set of readings and a waveform photograph in such a short period of time. However, after a great deal of persistence, a few "high" speed load tests were successfully completed. The results of one such test are presented below:

Inverter Supply Voltages:	+23.15 / -23.32 volts d.c.
Load Angle:	0°
Motor Speed:	8500 rpm
Inverter Supply Rail Currents:	21.0 amps d.c. (each)
Power Input to Inverter:	976 Watts
Power Output from Inverter:	875 Watts
Inverter Output Phase Voltage:	21.69 volts r.m.s.
Phase Current:	6.2 amps r.m.s.
Power Input to Motor:	525 Watts
Motor Phase Voltage:	10.1 volts r.m.s.
Nett Torque:	0.39 Nm
Gross Torque:	0.455 Nm

It should be noted that the gross motor torque was estimated with the aid of the loss torque characteristic shown in fig. 9.13. The nett and gross motor efficiencies are 66.1% and 77.1% respectively. The inverter has an efficiency of 89.7%: this compares well with the inverter efficiency measured during earlier tests (see sub-sections 9.3.3 and 9.3.4).

The motor efficiencies must be regarded with some caution because of the crude torque measurement technique. It was very difficult to hold the load torque steady, and so the voltage and current readings tended to vary during the course of a set of measurements. Nevertheless, the test shows that the motor can produce a significant power output under load. The nett and gross power outputs for the "spot" load test at 8500 rpm were 347 Watts and 405 Watts respectively. These are, of course, nowhere near the originally planned power output level, but it is almost certain that great improvements in nett power output are possible with suitable changes to the stator core and winding.

Fig. 9.93 shows a photograph of the inverter and motor phase waveforms for the load test at 8500 rpm (the speed was actually about 8100 rpm at the instant of the photograph). The current waveform has a good quasi-square shape and it reaches a peak value of about 9.0 amps. The phase current takes about 0.6ms to rise to its on value of approximately 7.0 amps. The half cycle period of the waveform is about 3.8ms and so the current rise occupies 15.8% of the half cycle period. This may appear to be a large proportion, but the waveform has a good quasi-square shape. The quality of the quasi-square current waveform depends very much on the current rise (and fall) times. A quasi-square waveform "quality figure", which relates the rise and fall times to the half cycle period, is discussed in section 9.6.

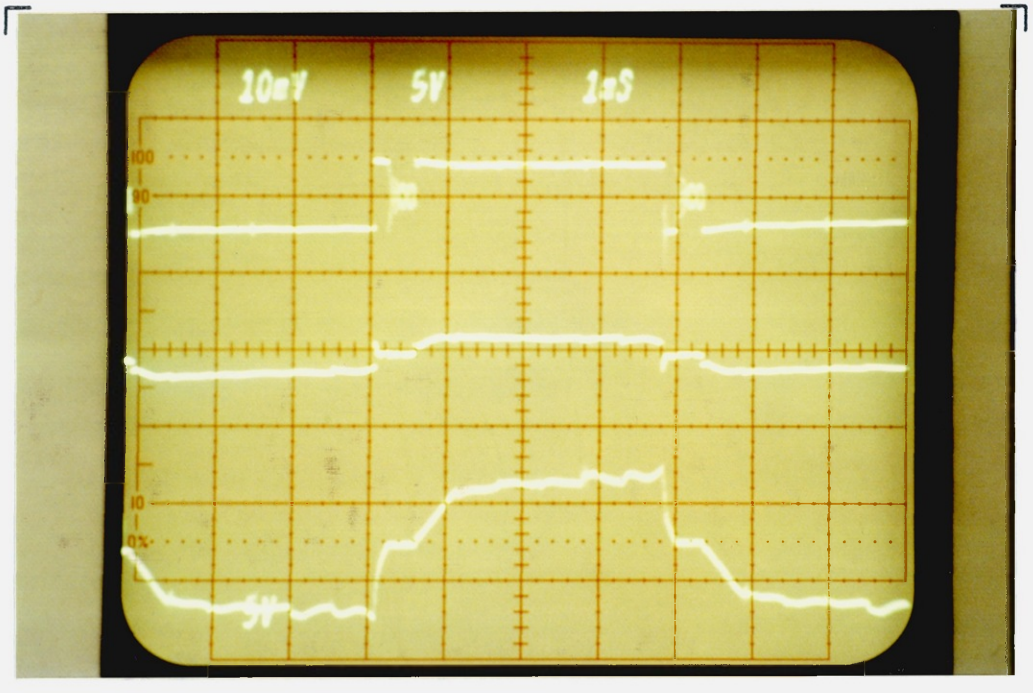


FIG. 9.93. 7 PHASE MOTOR : 0° LOAD ANGLE — LOAD TEST AT 8500 RPM, 0·39 Nm NETT TORQUE — TOP : INVERTER PHASE VOLTAGE (50V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (50V/cm) — BOTTOM : PHASE CURRENT (10A/cm) — 1ms/cm.

9.3.10 The "Smoothness" of the Inverter Supply Rail Currents

The way in which the switching of the phases of the inverter is interlaced ensures that at any time, three phases draw current from the positive supply rail of the inverter and three phases draw current from the negative supply rail. Therefore, ideally, since each phase draws the same magnitude of current, there should be a constant current drawn from each inverter supply rail for a given motor load condition. In practice, switching transients and freewheel currents cause there to be some ripple in the inverter supply rail currents.

The positive supply rail current of the inverter was monitored with a Tektronix current probe. Fig. 9.94 is a photograph showing the positive rail current waveform for the motor running on no load with a 0° load angle and an inverter supply voltage of ±7 volts. The current has an average magnitude of about 2.5 amps and the ripple has a maximum amplitude of 0.5 amps. The current can therefore be classed as essentially constant.

Fig. 9.95 is a photograph showing the inverter positive rail current waveform and the motor phase one voltage waveform. The photograph was taken with the motor running on no load with a 0° load angle and an inverter supply voltage of ±8 volts. The zero volt sections of the phase voltage waveform coincide with current ripple peaks in the current waveform: the remaining ripple peaks coincide with the various zero volt sections of the six other phase voltage waveforms. The phase currents commutate during the zero volt sections and so it is not surprising that the ripple in the inverter supply rail current coincides with the zero volt sections.

The inverter positive supply rail current waveform was monitored at higher current levels by loading the motor.

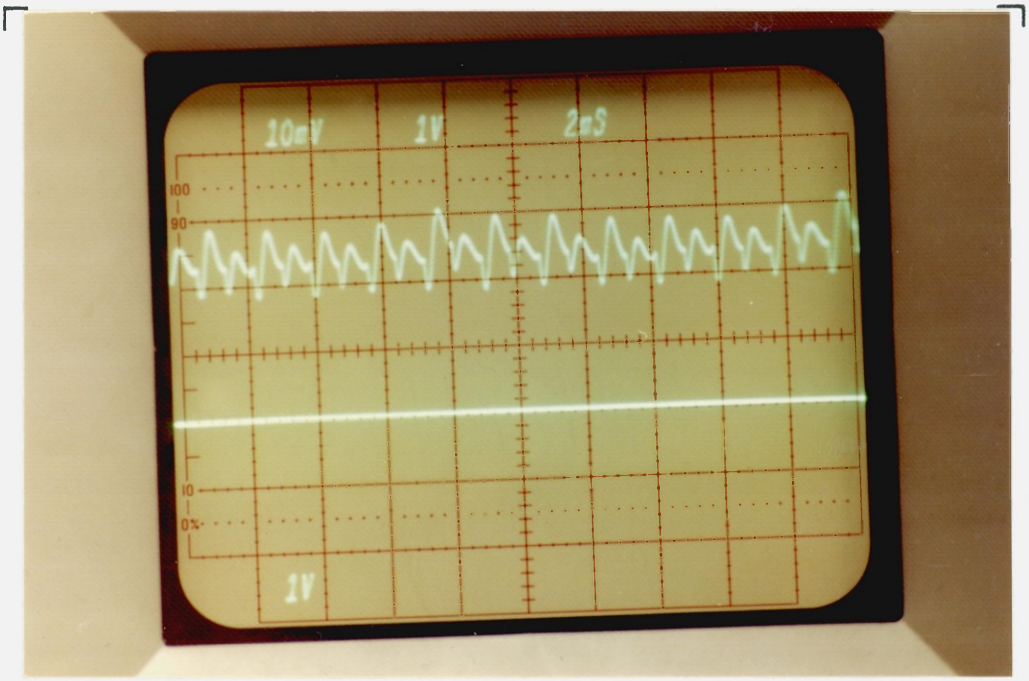


FIG. 9.94. THE INVERTER POSITIVE SUPPLY RAIL CURRENT WAVEFORM — 1A/cm (ZERO INDICATED BY STRAIGHT LINE TRACE) — ±7 VOLTS SUPPLIED TO INVERTER; MOTOR UNLOADED; 0° LOAD ANGLE — 2ms/cm.

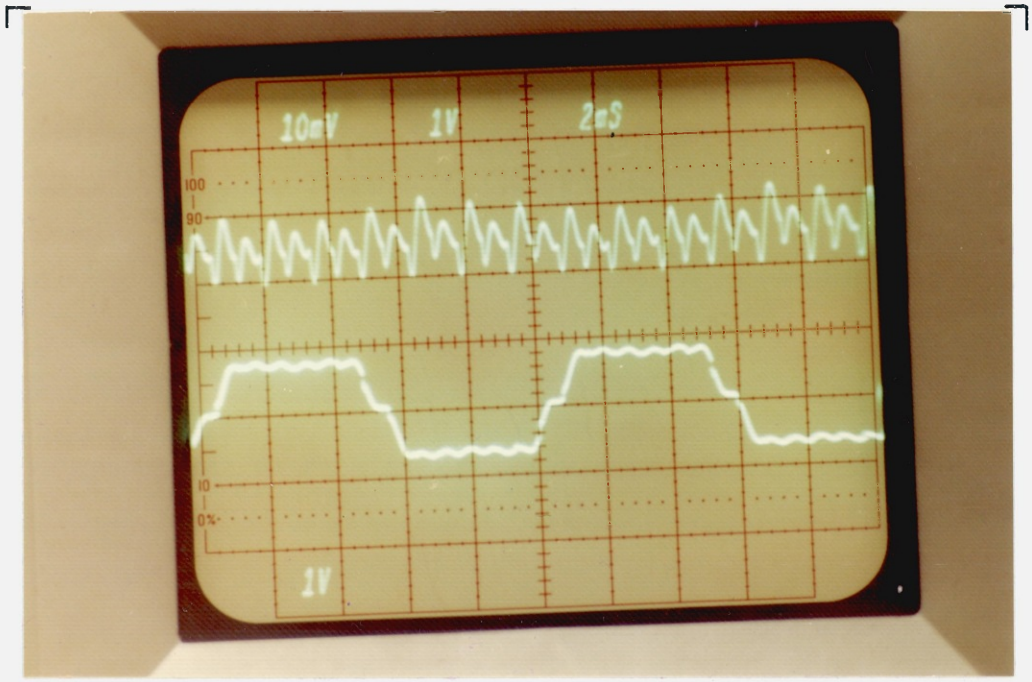


FIG. 9.95. TOP: THE INVERTER POSITIVE SUPPLY RAIL CURRENT WAVEFORM (1A/cm) — BOTTOM: MOTOR PHASE VOLTAGE WAVEFORM (10V/cm) — ±8 VOLTS SUPPLIED TO INVERTER; MOTOR UNLOADED; 0° LOAD ANGLE — 2ms/cm — (NOTE ZERO FOR CURRENT WAVEFORM COINCIDES WITH VOLTAGE WAVEFORM CENTRE LINE).

The current waveform was of the same form (that is, essentially constant) as in the earlier no load tests.

The inverter negative supply rail current waveform was also monitored and was found to have the same form as the positive supply rail current. It can therefore be concluded that in practical terms, under given load conditions, the inverter does indeed draw balanced constant currents from its positive and negative voltage supply rails. Each supply rail of the inverter is an ideal load for a "chopper" variable voltage power supply. Hence the MOSFET inverter could be supplied from a pair of programmable chopper power supplies. This is an attractive power supply solution for several reasons: choppers are very efficient; they are electrically simple and robust; they are relatively cheap and yet reliable in operation. It is therefore proposed that any future development of the square-wave motor system, should incorporate chopper-based voltage source power supplies, as replacements for the linear programmable voltage supplies used in the tests reported in this thesis.

9.3.11 The Phase Current Waveform Shape with the Inductive Section of the Protection Components Removed

The protection components were incorporated into the system, between the inverter and the motor, in order to prevent the phase currents exceeding the current rating of the hexfets during normal motor operation. The photographs of the inverter and motor waveforms presented earlier in this chapter show that the protection components did prevent excessive peaks in the phase currents.

The effectiveness of the inductive section of each set of protection components, in preventing current peaks at the leading and trailing edges of the current waveform, was checked by deliberately shorting out the inductor in phase 1 whilst the motor was operating on no load. Fig.

9.96 is a photograph showing the inverter and motor waveforms for the motor operating at 0° load angle with ±7 volts supplied to the inverter. Fig. 9.97 shows the same waveforms on an expanded timebase and fig. 9.98 shows the current waveform only. The phase current waveform has an r. m.s. value of 0.885 amps. However, the measured r.m.s. value gives no indication of the peaked nature of the waveform. It can be clearly seen in fig. 9.97 that the peak current values at the leading and trailing edges of the waveform are about 3.0 amps and 1.5 amps respectively, whilst the average "on" current is about 0.75 amps. Thus the current at turn-on is approximately fout times larger than the average "on" current. Therefore, it is clearly evident that the inductive section of each set of protection components does perform a very necessary task.

The reason for the large turn-on current peak can be easily seen in fig. 9.97: there is a substantial difference between the inverter output voltage and the motor phase voltage; the resulting potential difference therefore drives a large phase current through the 1Ω series protection resistor.

It is interesting to note than the removal of the 1.25mH protection inductor eliminates the 100kHz fuzz in the zero volt section of the inverter output voltage waveform. This is because with the absence of the inductor, the impedance between the inverter output and ground, via the motor winding, is sufficiently small that the 100kHz pick-up is simply absorbed without trace.

The removal of the inductor from the protection components obviously reduces the total phase inductance. This has the direct effect of shortening the freewheel periods in the inverter output voltage waveform. The reduction of the freewheel periods could increase the reliability of the hexfets since it reduces the reverse-voltage stresses across the devices.

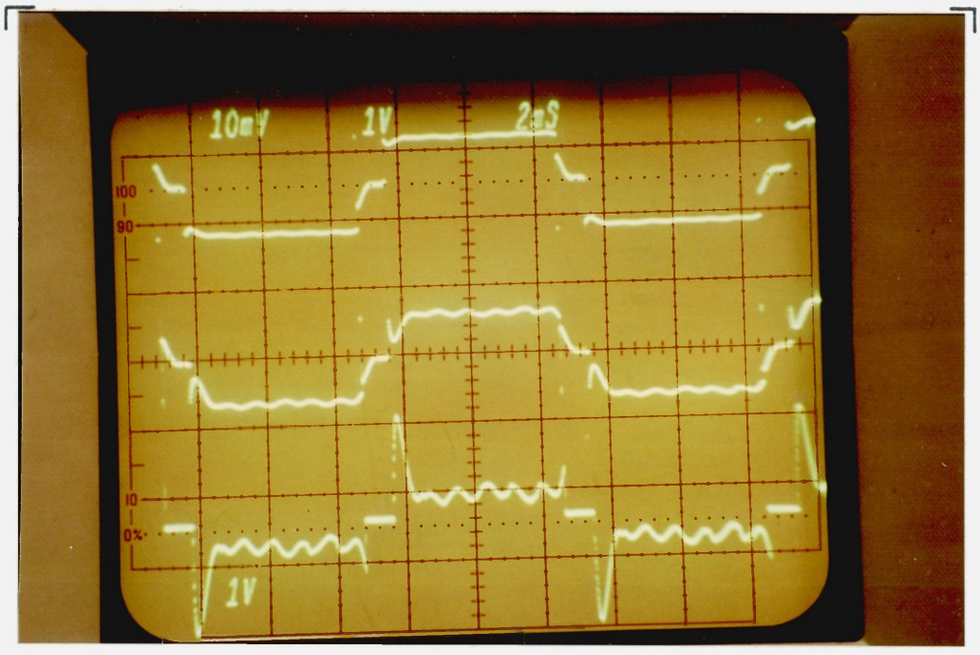


FIG. 9.96. PHASE ONE WAVEFORMS FOR THE 7 PHASE MOTOR WITH 1·25mH INDUCTOR REMOVED FROM PHASE ONE : 0° LOAD ANGLE ; ±7 VOLTS SUPPLIED TO INVERTER ; NO LOAD ; 5070 RPM : 0·885 AMPS RMS PHASE CURRENT — TOP : INVERTER OUTPUT VOLTAGE (10V/cm) — MIDDLE : MOTOR PHASE VOLTAGE (10V/cm) — BOTTOM : PHASE CURRENT (2A/cm) —— 2ms/cm.

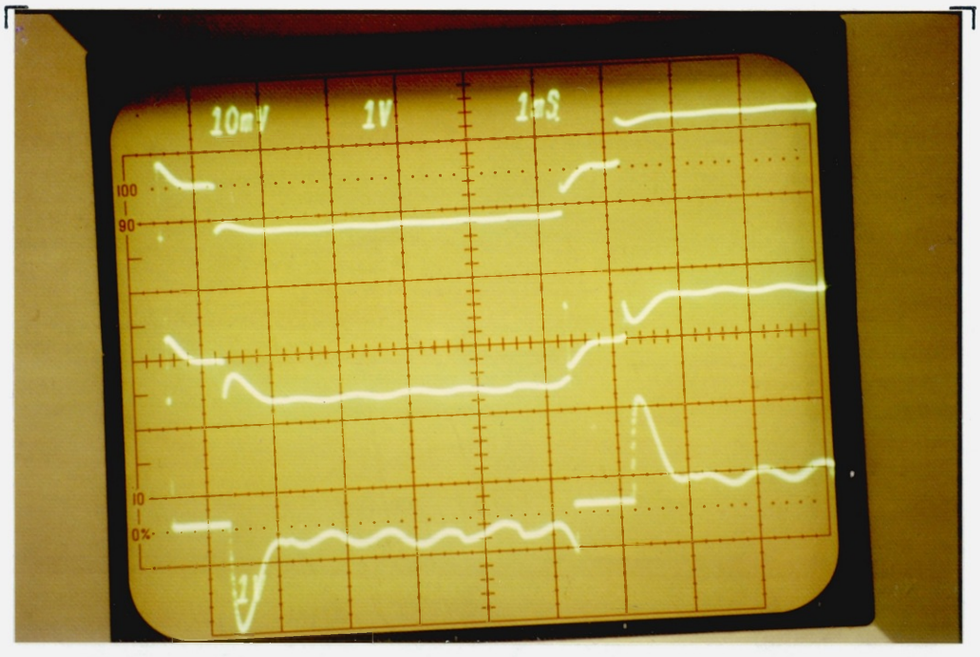


FIG. 9.97. 7 PHASE MOTOR WITH 1·25mH INDUCTOR REMOVED FROM PHASE ONE : CONDITIONS AND WAVEFORMS AS FOR FIG 9.96 EXCEPT TIMEBASE —— 1ms/cm.

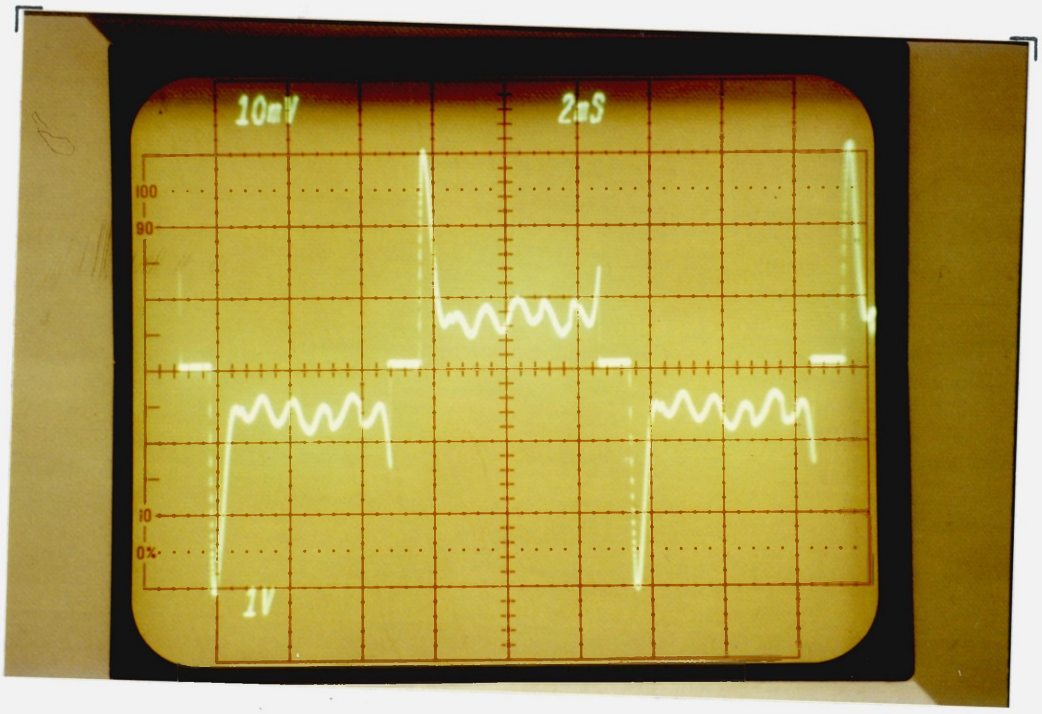


FIG. 9.98. 7 PHASE MOTOR WITH 1·25mH INDUCTOR REMOVED FROM PHASE ONE — PHASE CURRENT WAVEFORM (1A/cm) — MOTOR CONDITIONS AS FOR FIG. 9.96 — 2ms/cm.

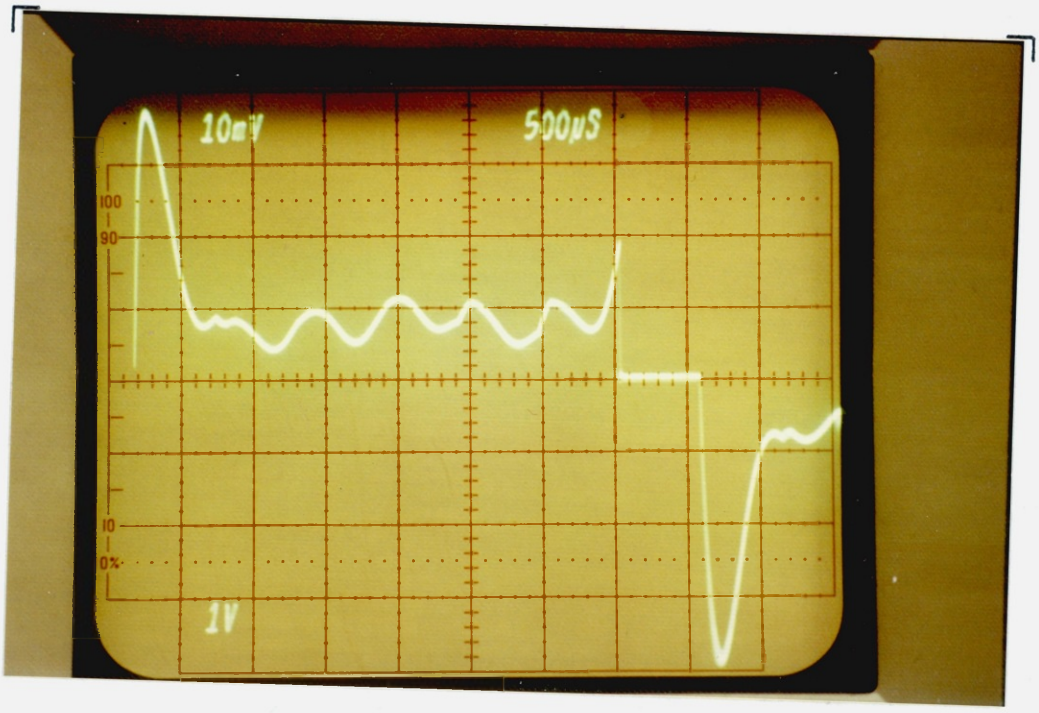


FIG. 9.99. 7 PHASE MOTOR WITH 1·25mH INDUCTOR REMOVED FROM PHASE ONE — PHASE CURRENT WAVEFORM (1A/cm) — ±10 VOLTS SUPPLIED TO INVERTER; 0° LOAD ANGLE; NO LOAD; 7500 RPM; 1·10 AMPS RMS PHASE CURRENT — 500μs/cm.

The removal of the inductor does seem to lead to some distortion in the motor phase voltage waveform: the areas of distortion coincide with the freewheel voltage spikes and the peaks at the leading and trailing edges of the current waveform. The phase current has much more ripple in it due to the lower total phase inductance.

Finally, the form of the "no protection inductor" phase current waveshape can be seen very clearly in fig. 9.99: the photograph was taken with the motor operating on no load at 0° load angle with ±10 volts supplied to the inverter. The current has an r.m.s. value of 1.10 amps with leading and trailing edge peaks of 3.8 amps and 1.9 amps respectively.

It can be concluded that for the system described in this thesis, the protection inductors are beneficial in two main ways: firstly, they limit the magnitude of the peaks in the phase current waveforms; secondly, they smooth out any variations in the "on" values of the phase currents (caused by the ripple voltages on the phase winding back-emfs).

9.3.12 Check on the Rotor Flux Density at the Conclusion of the Tests

The rotor flux density was checked at the conclusion of the tests performed on the motor. The open circuit phase voltage was measured using an Avometer whilst the rotor was rotated by the eddy current brake unit. It was found that the value of the generated emf per rpm was identical to that measured just after the motor was first assembled (that is, the back-emf constant, k_b, had a value of 11.80mV per radian per second). Hence, it can be concluded that the rotor magnets did not suffer any demagnetization during the motor test programme.

9.4 Comments on the Inverter Efficiency

The tests on the 7 phase motor system indicate that the hexfet inverter operates with an efficiency in the region of 90%, even when the magnitudes of the phase currents approach the hexfet rated current value of 7.0 amps. For example, the "high speed" load test results presented in sub-section 9.3.9 show that at an r.m.s. phase current of 6.2 amps, the inverter efficiency was 89.7%. Hence the 90% efficiency figure is not only true for "light-load" conditions.

The inverter efficiency was calculated from the measured input and output powers of the inverter. It is interesting to compare the measured inverter power loss (the difference between the input and output powers), against the calculated inverter power loss.

The efficiency of the inverter is essentially determined by the total series resistance encountered by the phase currents, as they flow via the hexfet circuit boards to the motor windings (the switching losses associated with the hexfets or the RC snubbers are negligible at the frequencies employed on the 7 phase motor system; in addition the very small gate drive power requirements of the hexfets are insignificant). The total series resistance on a hexfet circuit board, is principally made up of the hexfet on-resistance and the semiconductor fuse resistance. The on-resistance for the IRF230 hexfets is a maximum of 0.4Ω. The resistance of the 10CT semiconductor fuses can be obtained from the fact that they dissipate 4 Watts at their maximum current of 10 amps: that is, they have a resistance of 0.04Ω. Therefore, neglecting the resistance of the printed circuit board tracks and connectors, it is reasonable to assume that the total series resistance on a hexfet board (R_{total}) is 0.44Ω.

The power-loss per hexfet circuit board, P_{bd}, is then

given by:

$$P_{bd} = I_{rms}^{2} . R_{total} \qquad 9.28$$

where I_{rms} is the r.m.s. phase current. The total inverter power loss P_{il} is then simply:

$$P_{il} = 7\ P_{bd} \qquad 9.29$$

The calculated inverter power loss has been compared against three sets of measured inverter power losses. The calculated and measured results are presented graphically. Fig. 9.100 shows the measured and calculated power losses plotted against the r.m.s. phase current for the motor operated on load, with a 0° load angle and an inverter supply voltage of ±10 volts. Figs. 9.101 and 9.102 present similar sets of information as fig. 9.100: fig. 9.101 is for the motor on load with a 0° load angle and an inverter supply voltage of ±22 volts; fig. 9.102 is for the motor on load with a 6° load angle and an inverter supply voltage of ±18 volts.

In all cases the measured power losses are larger than the calculated losses, although the margin of error decreases as the phase current increases. The differences are believed to be chiefly due to the inaccuracies involved in measuring the inverter input power. The input power was calculated from readings of the inverter input voltages and currents (both d.c. quantities). The large direct currents drawn by the inverter were measured to the nearest tenth of an amp on moving coil ammeters. It is possible that this accuracy of current measurement was not sufficient. For example, consider the situation if the inverter is supplied with ±22 volts and the supply rail currents are somewhere between 4.0 amps and 4.1 amps. If the current reading is rounded to 4.0 amps, the positive rail power input is 88 Watts; a current value of 4.1 amps yields a power value of 90.2 Watts. Thus a difference of 3.2 Watts

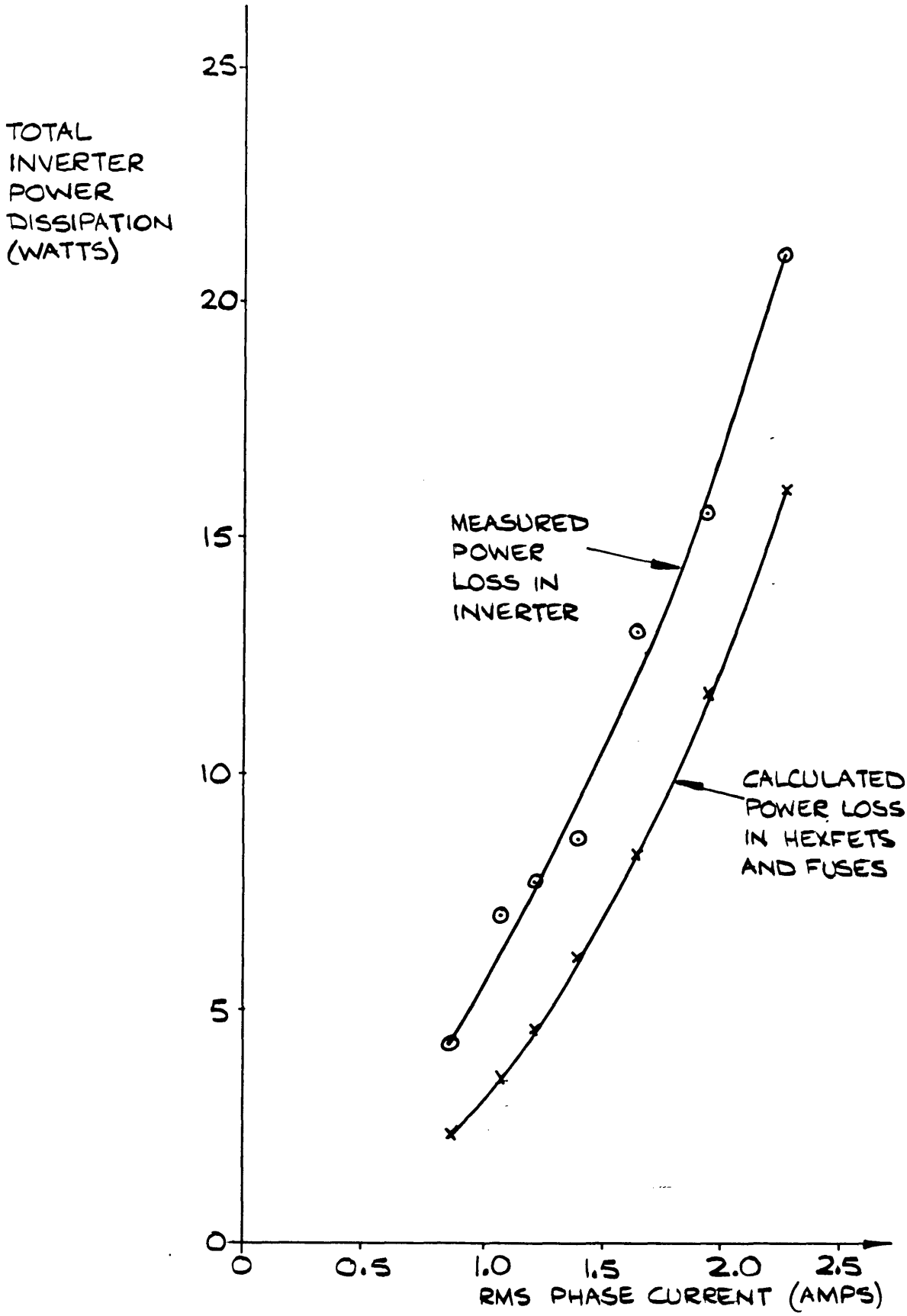


FIGURE 3.100 7 PHASE HEXFET INVERTER - MEASURED AND CALCULATED POWER LOSSES VERSUS PHASE CURRENT FOR 0°, ± 10 VOLT OPERATION OF SYSTEM

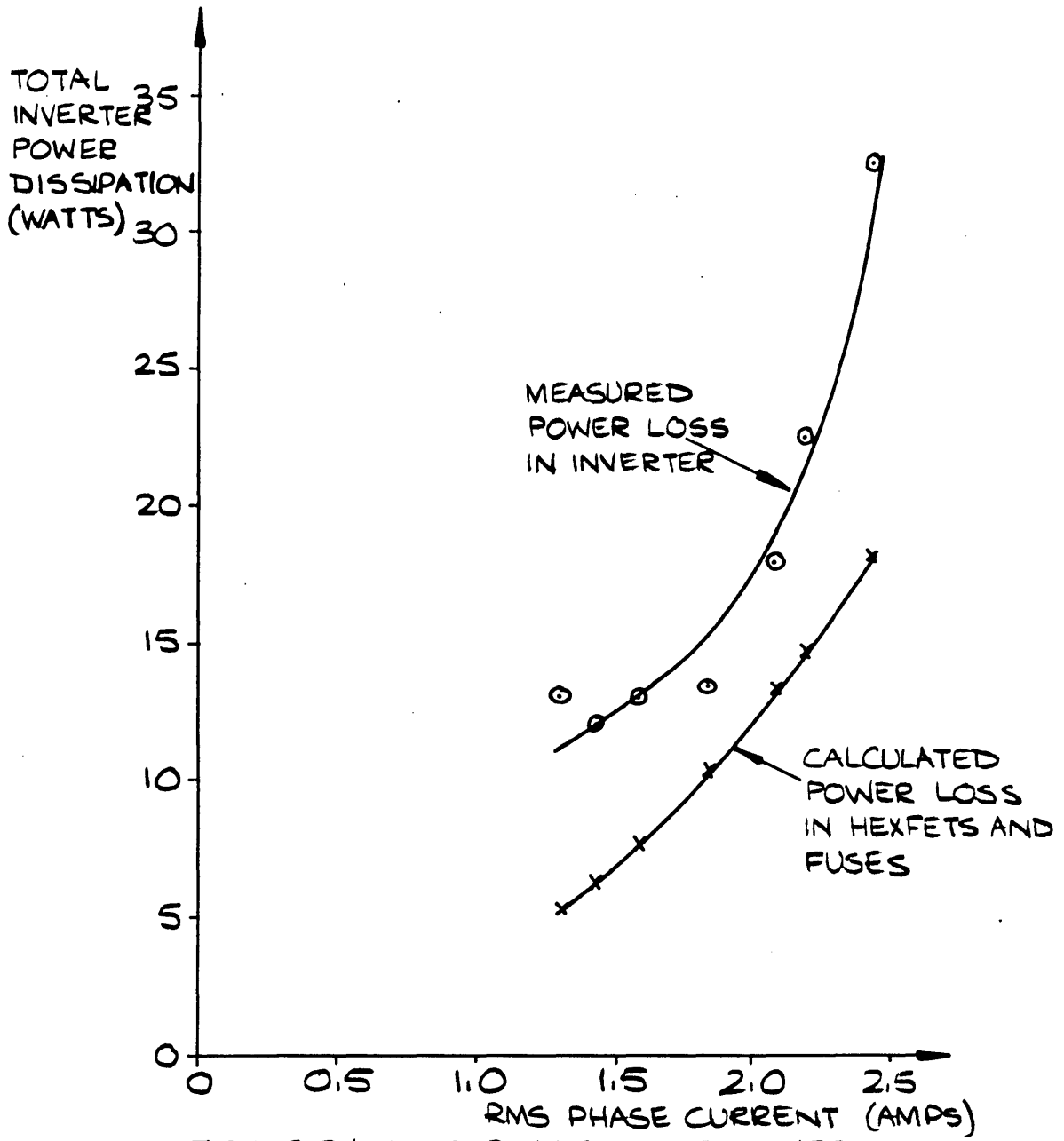


FIGURE 9.101 7 PHASE HEXFET INVERTER- MEASURED AND CALCULATED POWER LOSSES VERSUS PHASE CURRENT FOR 0°, ± 22 VOLT OPERATION OF SYSTEM

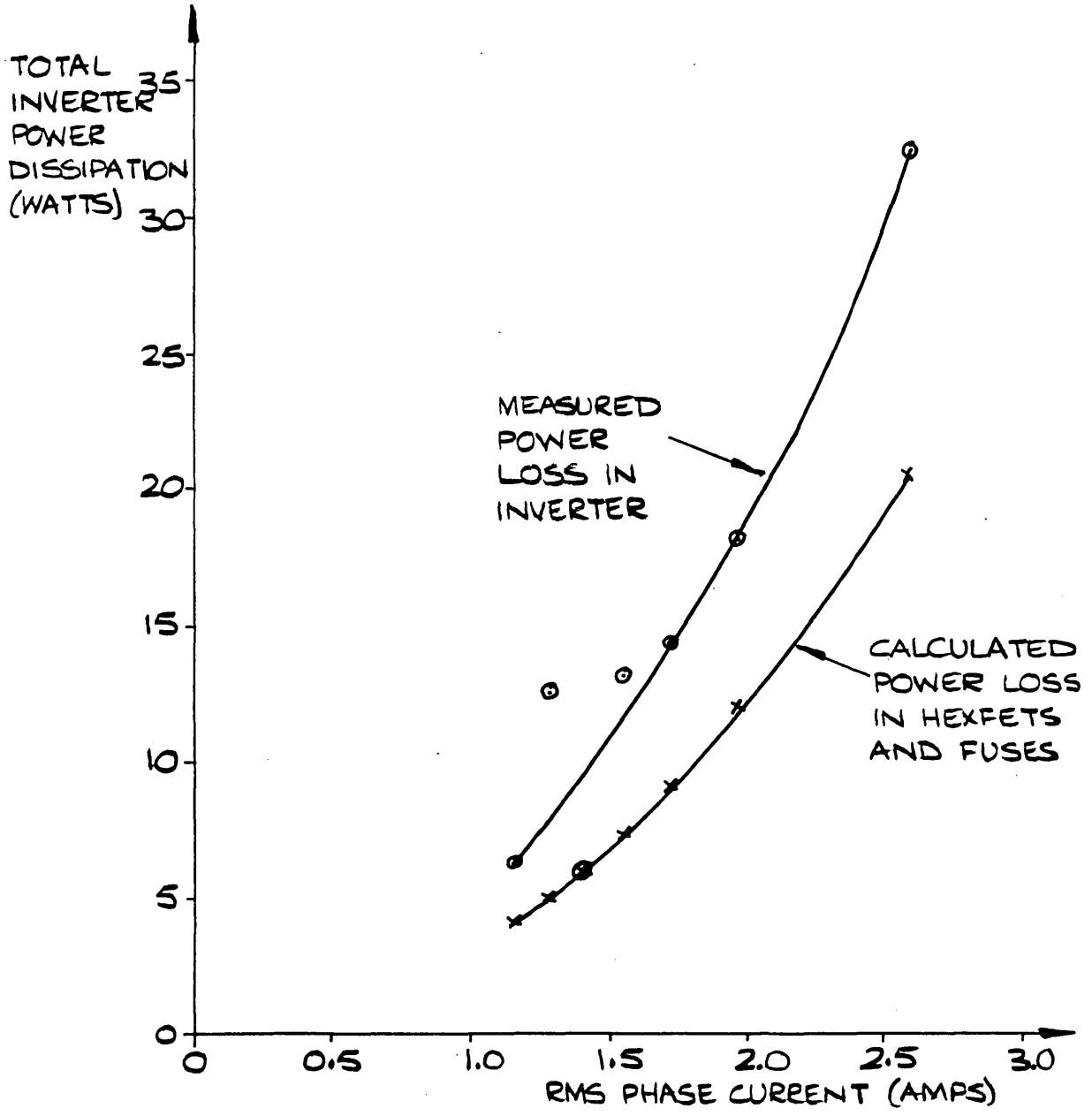


FIGURE 3.102 7 PHASE HEXFET INVERTER - MEASURED AND CALCULATED POWER LOSSES VERSUS PHASE CURRENT FOR 6°, ±18 VOLT OPERATION OF SYSTEM

in the positive rail power input can result from the way in which the current reading is rounded, and the total difference for both supply rails is 6.4 Watts. Such a difference could drastically affect the value of inverter power loss obtained from the difference between the input and output powers.

It is also possible that some of the discrepancy between the measured and calculated power losses, is due to the fact that not all the phase currents are exactly equal, nor are the hexfet resistances all identical.

If the calculated inverter losses are more accurate than the measured losses, this would mean that the inverter efficiency is actually well in excess of 90%. However, without a remeasurement of the inverter losses by a more accurate method, there is little more that can be said on the subject of the inverter efficiency, apart from the fact that the efficiency figures do not take account of the power consumption from the inverter logic supplies (+5 volts and +15 volts), but this is not a serious omission.

Allowing for the various errors which are inherent in the simple measurement and calculation methods used, it is reasonable to conclude that the inverter losses can be attributed to the hexfet on-resistances and the fuse resistances. If high voltage hexfets with lower on-resistances become available, it will undoubtedly be possible to construct MOSFET inverters with efficiencies significantly greater than 90% over the entire operating range of phase currents and voltages.

The basic shape of the efficiency versus phase current curves for a MOSFET inverter can be predicted readily if a simple resistive model for the inverter is used. This is possible because a MOSFET can be modelled to a fair degree of accuracy simply by its drain-source on resistance. Since, typically, an inverter is made up of 'n' identical

phase units, each with a series resistance R ohms (made up principally by the MOSFET drain-source resistance), the efficiency of the inverter as a whole can be calculated by considering just one phase as follows.

The d.c. input power to an inverter phase unit is $V_{in}.I$ Watts, whilst the loss power in the phase unit is $I^2.R$ Watts (both powers being independent of the mark/space ratio of the inverter output waveform). The efficiency η_1 of the phase unit is then simply:

$$\eta_1 = \frac{\text{input-losses}}{\text{input}} \qquad 9.30$$

$$\eta_1 = \frac{V_{in}.I - I^2.R}{V_{in}.I} \qquad 9.31$$

If the inverter is operated from a constant voltage supply (V_{in} = constant), as in the tests reported in sub-section 9.3.3, equation 9.31 reduces to:

$$\eta_{1v} = (1 - k_1.I) \qquad 9.32$$

where $k_1 = R/V_{in}$ 9.33

Thus with a constant inverter supply voltage, the efficiency falls linearly with increasing phase current, I.

Alternatively, the relationship between the efficiency and phase current can be considered with regard to the actual power output of the inverter. Ideally, the square-wave motor is a unity power factor load, and so if the MOSFET inverter produces a square-wave phase output of amplitude V_o volts at I amps, the power output is $V_o.I$ Watts. The power loss within the phase unit is $I^2.R$ Watts, and the inverter efficiency η_2 is:

$$\eta_2 = \frac{\text{output}}{\text{output + losses}} \qquad 9.34$$

Hence:

$$\eta_2 = \frac{V_o.I}{V_o.I + I^2.R} \qquad 9.35$$

If the output voltage amplitude is maintained constant as the phase current varies, such as in the tests reported in sub-section 9.3.4, the efficiency relationship reduces to:

$$\eta_{2v} = \frac{1}{1 + k_2.I} \qquad 9.36$$

where $k_2 = R/V_o$ 9.37

Thus, with a constant inverter output-voltage amplitude, the efficiency falls with increasing phase current I, as shown by equation 9.36.

The previously assumed hexfet circuit board series-resistance value of 0.44Ω can be substituted into equations 9.33 and 9.37 if desired. Then, for example, with operating conditions of V_{in} = 10 volts and I = 1.0 amps, equation 9.32 predicts an inverter efficiency η_{1v} of 95.6%. This correlates reasonably well with the spot measured value of 91.5% under these conditions (see fig. 9.23). Equations 9.32 and 9.36 have not been checked for accuracy against all the measured inverter efficiencies presented in sub-sections 9.3.3 and 9.3.4. Nevertheless, it is felt that the equations give a good indication of how the efficiency of a MOSFET inverter is basically related to the load current.

9.5 Comments on the Accuracy of the Loss-Torque Characteristics

The rotor I.W.F. loss-torque versus speed character-

istics presented in figs. 9.13 and 9.14 are made up by three constituent torques, as explained in sub-section 9.3.2. Briefly, the three constituent retardation torques are bearing friction torque, windage torque, and a reaction torque due to stator eddy currents induced by the rotor flux. The significant stator heating during the motor tests is an indication that the eddy current reaction torque is probably the major component of the total loss-torque.

The gross motor efficiency characteristics presented in sub-section 9.3.3 include the loss-torque effects. Obviously any inaccuracy in the loss-torque characteristics has a direct effect on the accuracy of the calculated values of gross motor efficiency. The loss-torque characteristic shown in fig. 9.13 was used for the gross motor efficiency results in sub-section 9.3.3. Several of the gross motor efficiency characteristics presented in sub-section 9.3.3 show motor efficiencies of 100%. Since such efficiencies are impossible for a machine such as the 7 phase motor, it can be concluded that either the measurements of nett output power from the motor were in general too large (that is, the dynamometer was "over-reading"), or the overall magnitude of the loss-torque characteristics were too large (due to inaccuracies in their measurement and calculation). Checks on the dynamometer during the motor tests showed that it was reasonably accurate, and so it is probable that the errors in the gross motor efficiency are due to inaccuracies in the loss-torque characteristic of fig. 9.13. The loss-torque characteristic shown in fig. 9.14 is probably similarly inaccurate, since it was derived in an identical manner to the characteristic shown in fig. 9.13.

The loss-torque of a motor can be estimated from a knowledge of its no-load input power. Hence it was decided to check the characteristics in figs. 9.13 and 9.14 against spot values of loss-torque calculated from the no-load motor input power.

The no-load input power to the 7 phase motor is consumed by the copper losses in the windings, the iron losses in the stator due to the phase currents, and the I.W.F. rotational losses on the rotor. Allowance for the copper losses can be easily made, but there is no simple formula for the stator iron losses caused by the stator phase currents. However, the 7 phase motor was designed so that it would have a low stator inductance: that is, the stator driven flux per ampere-turn of stator current is very small as previously stated. Hence, the eddy current stator losses due to the stator phase currents should be negligible. If this is so, the power equation for the motor on no-load is:

$$P_{in} = P_{rot} + P_{cu} \quad \text{Watts} \qquad 9.38$$

where P_{in} is the no-load input power to the motor, P_{rot} is the rotational losses of the rotor, and P_{cu} is the copper loss of the phase windings. The rotational losses are given by:

$$P_{rot} = T_{lt}.\omega \quad \text{Watts} \qquad 9.39$$

where T_{lt} is the total I.W.F. rotational loss-torque and ω is the angular velocity of the rotor. Substitution of equation 9.39 in 9.38 and rearranging yields:

$$T_{lt} = (P_{in} - P_{cu})/\omega \quad \text{Nm} \qquad 9.40$$

The total copper losses for the 7 phase motor are given by:

$$P_{cu} = 7.I_{rms}^{2}.R_{ph} \quad \text{Watts} \qquad 9.41$$

where I_{rms} is the r.m.s. phase current and R_{ph} is the phase winding resistance. Therefore equation 9.40 can be written:

$$T_{lt} = (P_{in} - 7.I_{rms}^{2}.R_{ph})/\omega \quad \text{Nm} \qquad 9.42$$

Equation 9.42 was used to calculate spot values of

loss-torque from the no-load measurements obtained during the motor tests. The no-load measurements for various motor speeds, with the dural disc mounted on the rotor shaft, were used to calculate the loss-torque points presented in fig. 9.103. The loss-torque characteristic shown in fig. 9.13 is replotted in fig. 9.103 to enable a check on its accuracy to be made. It can be seen that the loss-torque values calculated from the 0°, 6°, and 13° load angle no-load measurements are remarkably close to the loss-torque characteristic. On average, it appears that the values predicted by the characteristic are perhaps slightly "too large", but the error is only of the order of a few percent.

The no-load measurements from the high speed motor tests (conducted without the dural disc on the rotor shaft), were used to calculate the loss-torque points presented in fig. 9.104. The loss-torque characteristic shown in fig. 9.14 is presented in fig. 9.104 for comparison with the estimated loss-torque points. It can be seen that the loss-torque values calculated from the 0° and 6° load angle measurements are all below the loss-torque characteristic. It would therefore appear that values of loss-torque predicted by the characteristic in fig. 9.14 are over-sized throughout the whole speed range, but the error does not seem to be more than 10%.

The no-load power method of calculating loss-torque therefore seems to suggest that the loss-torque characteristics are essentially correct in form, but are possibly too large in magnitude by a few percent over the entire speed range.

The rotational loss-torque can actually be estimated to a reasonable degree of accuracy by using the appropriate torque prediction equation of the 7 phase motor. Equation 6.71 relates the gross-torque T to the r.m.s. phase current I_{rms}:

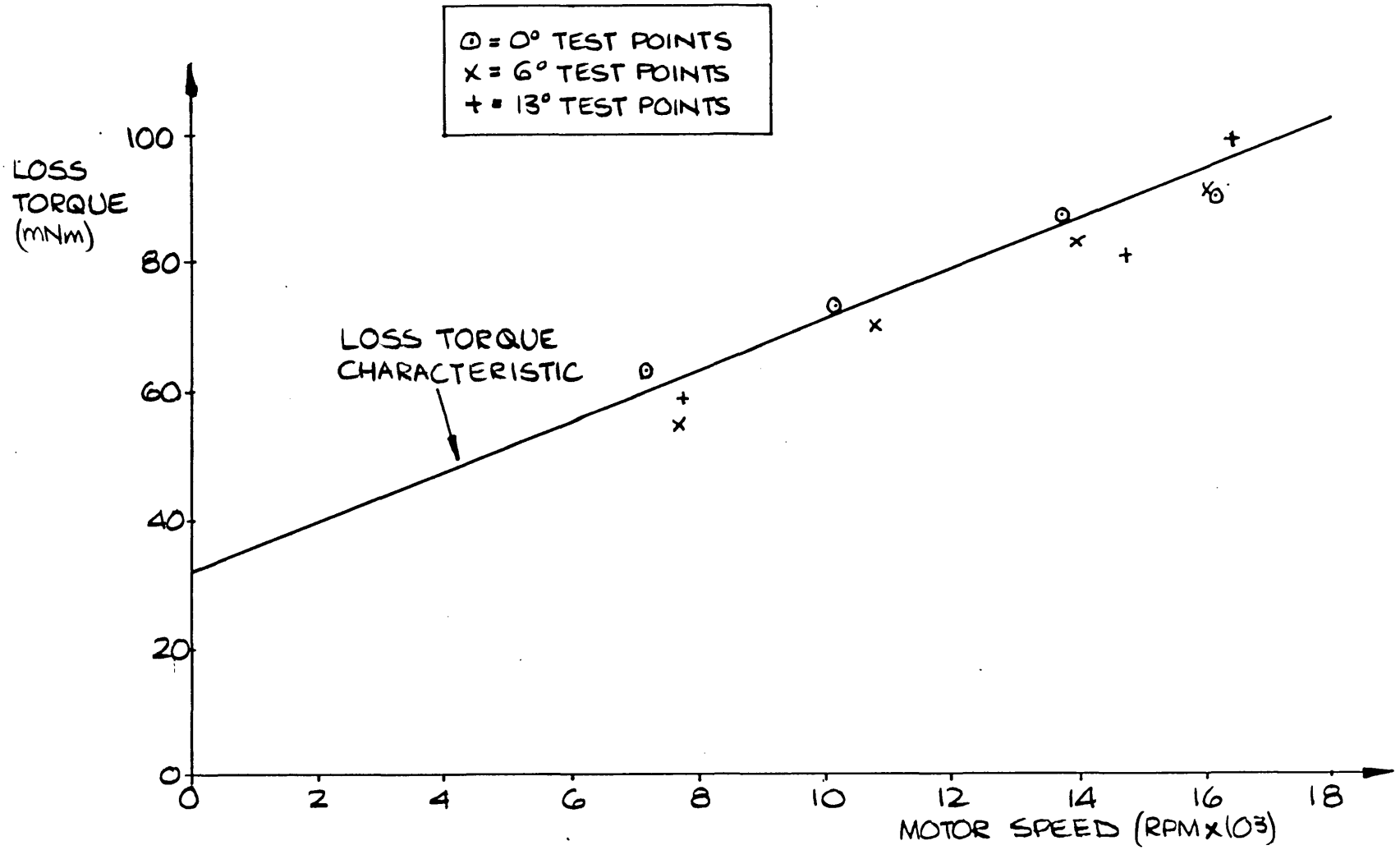


FIGURE 9.103 COMPARISON BETWEEN THE LOSS-TORQUE CHARACTERISTIC AND SPOT POINTS OF LOSS TORQUE CALCULATED FROM THE NO-LOAD MOTOR POWER CONSUMPTION (WITH DURAL DISC ON ROTOR)

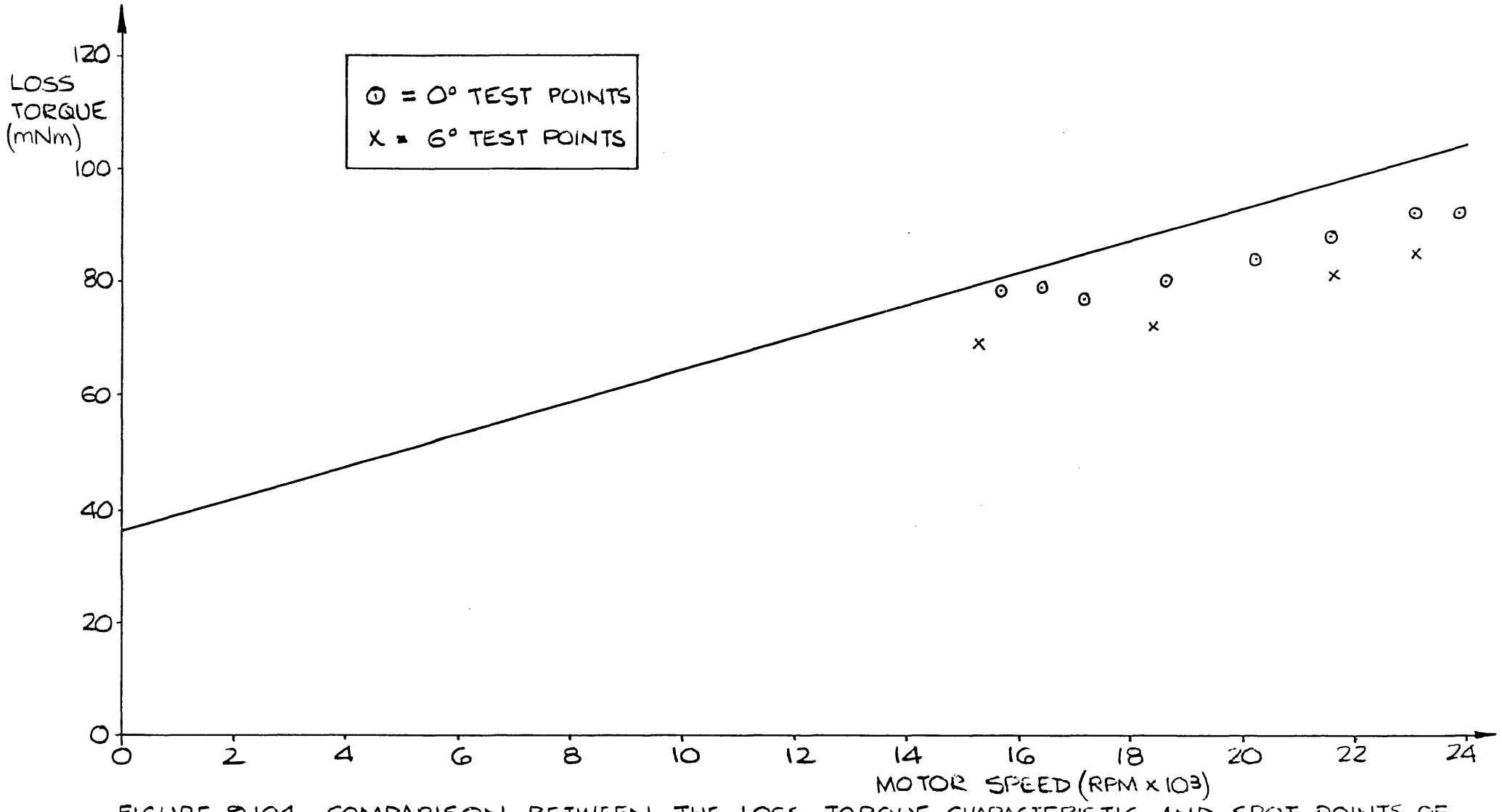


FIGURE 9.104 COMPARISON BETWEEN THE LOSS-TORQUE CHARACTERISTIC AND SPOT POINTS OF LOSS TORQUE CALCULATED FROM NO-LOAD MOTOR POWER CONSUMPTION (NO DURAL DISC ON ROTOR)

$$T = 12.\sqrt{7/6}.B_{ag}.N_{ph}.l_s.r_{gm}.I_{rms} \qquad (6.71)$$

This equation predicts the gross-torque to a surprising degree of accuracy as is shown in sub-section 9.3.5. Substitution of $B_{ag} = 0.298T$, $N_{ph} = 14.5$, $l_s = 0.05m$, and $r_{gm} = 0.0273m$ into equation 6.71 yields equation 9.13:

$$T = 7.645 \times 10^{-2}.I_{rms} \quad Nm \qquad (9.13)$$

When a motor is in an "unloaded" state, the gross-torque T is totally absorbed by the I.W.F. rotational loss-torque T_{lt}. Therefore, equation 9.13 becomes:

$$T_{lt} = 7.645 \times 10^{-2}.I_{rms} \quad Nm \qquad 9.43$$

Equation 9.43 can then be used with the measured no-load r.m.s. phase current values to predict the motor rotational loss-torque.

The no-load r.m.s. phase currents for various motor speeds, with the dural disc mounted on the rotor shaft, were substituted into equation 9.43 to calculate the predicted loss-torque points presented in fig. 9.105. The loss characteristic shown in fig. 9.13 is replotted in fig. 9.105 to enable a comparison between it and the predicted points to be made. It can be seen that the loss-torque values calculated from the 0^o, 6^o, and 13^o load angle no-load phase currents are reasonably close to the loss-torque characteristic, although all the predicted points lie "above" the characteristic.

The no-load r.m.s. phase currents from the 0^o load angle high speed motor tests (conducted without the dural disc on the rotor shaft) were used to calculate the loss-torque points presented in fig. 9.106. The loss-torque characteristic shown in fig. 9.14 is also presented in fig. 9.106 for comparison with the calculated points. It can be seen that all the calculated loss-torque values lie very

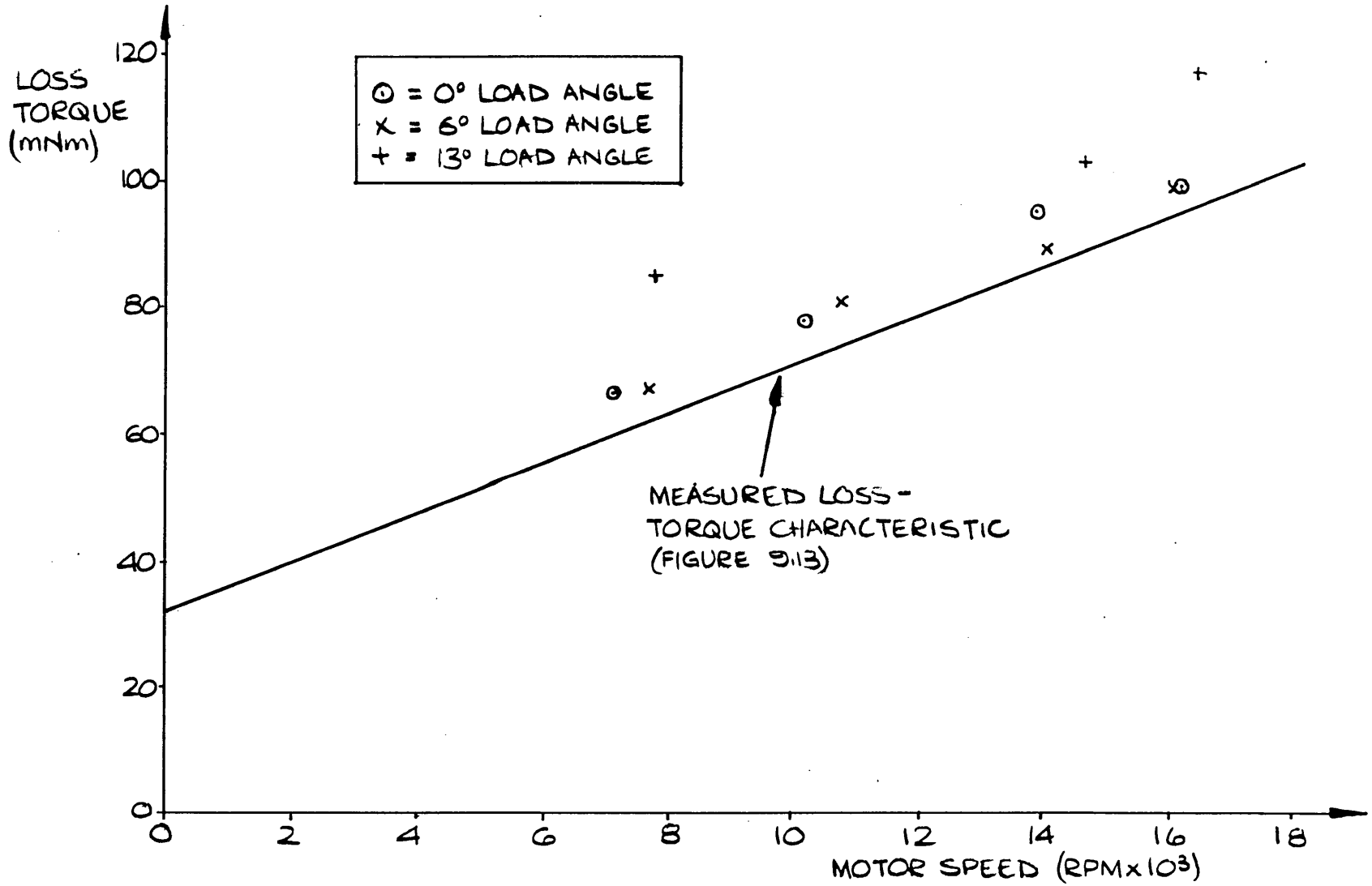


FIGURE 9.105 COMPARISON BETWEEN MEASURED LOSS-TORQUE CHARACTERISTIC (WITH DURAL DISC ON ROTOR) AND CALCULATED LOSS-TORQUE USING TORQUE CONSTANT (EQUATION 9.43)

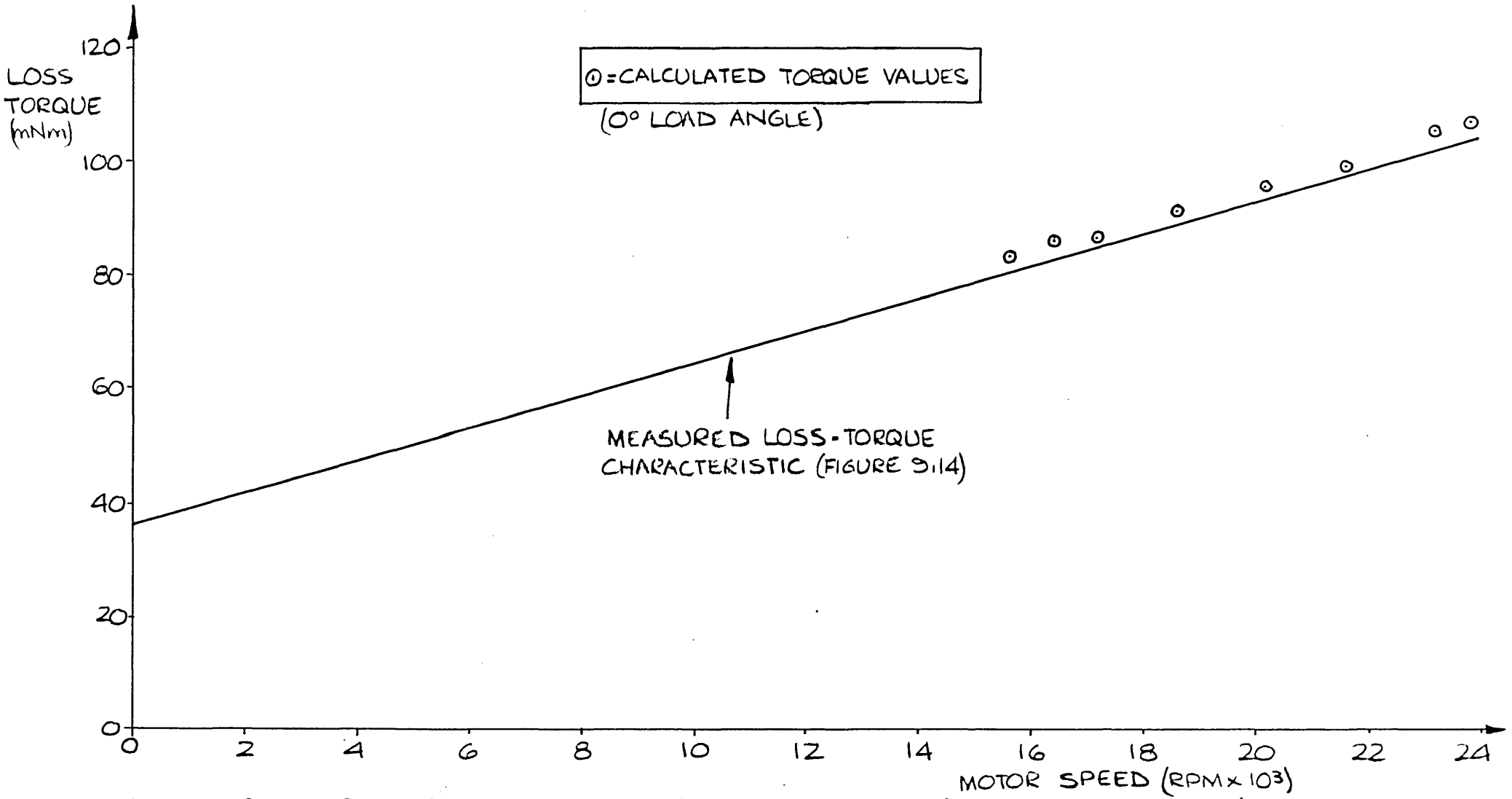


FIGURE 9.10G COMPARISON BETWEEN MEASURED LOSS-TORQUE CHARACTERISTIC (NO DURAL DISC ON ROTOR) AND CALCULATED LOSS-TORQUE USING TORQUE CONSTANT (EQUATION 9.43)

slightly above the characteristic.

In general, it can be concluded that the loss-torque values calculated by the torque-equation method indicate that both the loss-torque characteristics may be slightly "under-sized" over their respective speed ranges.

Hence, on the one hand, the no-load power method of calculating loss-torque suggests that both the loss-torque characteristics are slightly too large in magnitude over their entire speed ranges, whilst on the other hand, the torque-equation method suggests that both the loss-torque characteristics are slightly too small over their entire speed ranges. Both loss-torque calculation methods confirm that the loss-torque characteristics are of the correct "form" and are reasonably accurate.

It is reasonable to assume that the no-load power method gives the most accurate indication of the true magnitudes of the loss-torque characteristics, because the power input to the motor was measured to a reasonable degree of accuracy by an electronic wattmeter. The fact that the loss-torque values which were calculated by the torque-equation method all lie above the loss-torque characteristics, may be due to the assumptions on which the method is based. The torque-equation method assumes that the phase current has an ideal quasi-square waveform, whereas in practice the current waveform is rarely, if ever, of this form. Hence, it is likely that the loss-torques calculated by equation 9.43, for given r.m.s. phase current values, are not strictly accurate.

Therefore, if the loss-torque values calculated from the no-load power measurements are taken as being correct, it can be concluded that the magnitudes of the two loss-torque characteristics (figs. 9.13 and 9.14) are too large. This conclusion is supported by the previously mentioned fact that some of the calculated gross motor efficiency

values presented in sub-section 9.3.3 are 100%: a reduction in the magnitudes of the loss-torque characteristics would certainly reduce the calculated values of gross motor efficiency to a level that is compatible with the physical parameters of the 7 phase motor.

9.6 Comments on the Phase Current Waveform Shape

In general, the photographs of motor waveforms presented in this chapter show that the 7 phase motor does operate with the desired form of quasi-square phase current. It was apparent during the motor tests, that the shape of the phase current was dependent to some extent on both the speed of the motor and the magnitude of the phase current. However, it was usually possible to keep the current waveform reasonably close to the desired shape by selecting an appropriate load angle (0°, 6°, or 13°).

The current waveform shape has a direct effect on the maximum torque that the 7 phase motor can deliver. The torque "T" developed by the motor is proportional to the average phase current, I_{av}. That is:

$$T = k_t . I_{av} \quad Nm \qquad 9.44$$

where k_t is an arbitrary constant. The ratio (in a given half cycle) of the average phase current, I_{av}, and the peak phase current, I_{ph}, can be defined as the phase current utilisation factor, U:

$$U = I_{av}/I_{ph} \qquad 9.45$$

The utilisation factor depends directly on the phase current waveform shape. It has a maximum value (6/7) when the phase current waveform has the desired quasi-square wave shape. The motor torque can be written in terms of the peak phase currents and the utilisation factor:

$$T = k_t.U.I_{ph} \qquad 9.46$$

The peak phase current in the 7 phase motor system is set by the continuous drain current rating of the hexfets (the phase winding also places a limit on the peak phase current, but its thermal time constant is much larger than that of the hexfets). Examination of equation 9.46 indicates that in order to obtain a specified maximum torque from the motor, whilst at the same time using hexfets with the minimum possible drain current rating, it is necessary to ensure that the utilisation factor is as close to 6/7 as possible. When this is arranged and the motor is fully loaded, each hexfet carries a current just less than or equal to its rated drain current, whenever it conducts. The cost of hexfets is to some extent dependent on the drain current rating, and so the use of hexfets that possess a current rating just sufficient to cope with the full-load motor conditions, allows the cheapest possible hexfets to be chosen. This approach can also be regarded as making good economic use of the hexfets because the current rating is fully utilised. Unfortunately, these "cost-effective" benefits are gained at the expense of having no current overload capability under full-load conditions.

It is obviously beneficial if the phase current waveforms in the 7 phase "square wave" motor system, are as close to the optimum quasi-square shape as possible, since the system is based on the idea of a "square wave" of rotor flux interacting with a "square wave" of stator phase currents. However, there is no feedback in the system to ensure that the current waveform has the optimum quasi-square shape at any torque loading or speed. Therefore, it is of interest to determine a relationship, which can be used to predict the ranges over which the speed and phase current magnitudes can vary without there being any significant resultant distortion to the desired current waveform shape.

The question of what constitutes a good quasi-square current waveform shape is very difficult. However, in general, it is the rise and fall times of the current waveform that "spoil" the shape. In particular, it appears from the experimental results, that the rise time is the major reason why the current waveform shape changes as the load conditions on the motor vary. Therefore, if current waveforms which have rise times less than a certain percentage of the half cycle time are classed as "good" quasi-square waves, it is possible to derive an equation to predict under what conditions the "good" shape can exist. The equation can aptly be described as the quasi-square goodness formula.

Having studied all the available phase current waveforms it was concluded that the quasi-square shape was "good" providing the rise time did not exceed about 15% of the half cycle time. If the current is assumed to rise more or less linearly up to its on value of I_{ph} amps in a time t_r seconds, the current waveshape is of the form shown schematically in fig. 9.107.

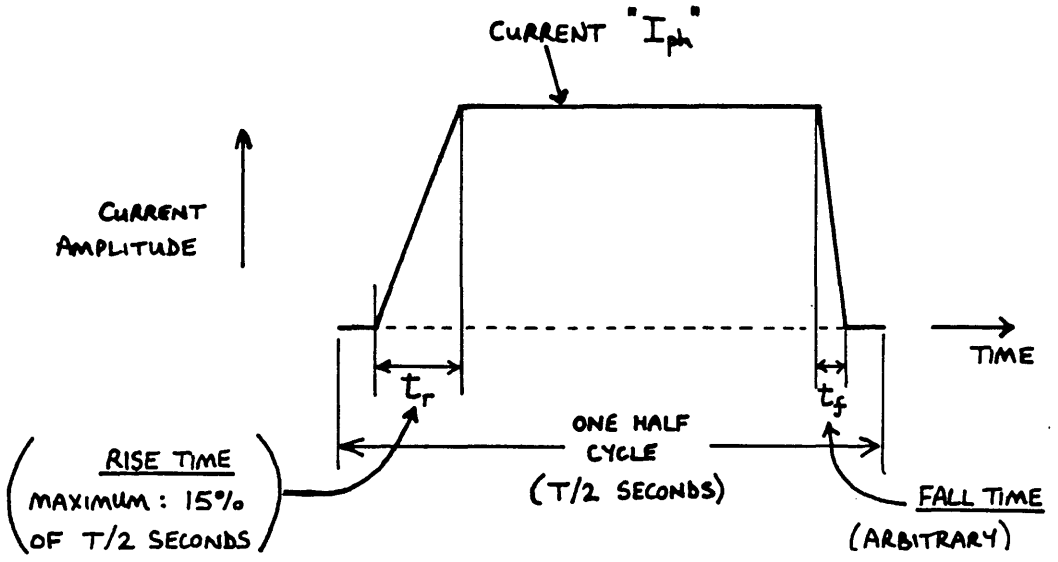


FIG. 9.107. WORST CASE RISE TIME FOR A "GOOD" QUASI-SQUARE CURRENT WAVEFORM.

The rise time shown in fig. 9.107 is actually 15% of the half cycle period, and so the diagram represents the worst "quasi-square" shape that falls within the chosen "good waveform" definition. It should be noted that the fall time, t_f, is shown for completeness in fig. 9.107, but its length is completely arbitrary.

If the period of one complete cycle of phase current is T seconds, then for a good current waveshape the rise time t_r must satisfy the equation:

$$t_r/(T/2) \leqslant 0.15 \qquad 9.47$$

If the resistance of the phase winding is neglected, the rate of rise of phase current during the rise time is given by:

$$(dI_{ph}/dt) = (V_{ph} - E_{ph})/L_{ph} \text{ Amps/second} \qquad 9.48$$

where V_{ph} is the applied phase voltage, E_{ph} is the back-emf, and L_{ph} is the phase inductance.

The actual value of the back emf during the rise time period is somewhat indeterminate owing to the fringing-flux effects within the machine. Hence, for the purposes of this analysis it is assumed to be zero. Equation 9.48 then reduces to:

$$(dI_{ph}/dt) = (V_{ph}/L_{ph}) \qquad 9.49$$

The current rises to its "on" value of I_{ph} amps by the end of the rise time:

$$I_{ph} = (dI/dt).t_r \qquad 9.50$$

Combining equations 9.47, 9.49, and 9.50 yields, after rearrangement:

$$(I_{ph}.L_{ph}/T.V_{ph}) \leqslant 0.075 \qquad 9.51$$

The period of each current cycle, T, is related to the motor speed, S (in rpm), by the relationship:

$$T = 60/S \quad \text{seconds} \qquad 9.52$$

Substituting for T in equation 9.51 yields:

$$I_{ph}.L_{ph}.S/V_{ph} \leqslant 4.5 \qquad 9.53$$

The phase inductance L_{ph} is a constant. In addition, the applied phase voltage V_{ph} can be regarded as a constant for given operating conditions. Hence, equation 9.53 can be rearranged to give:

$$I_{ph}.S \leqslant 4.5\ V_{ph}/L_{ph} \qquad 9.54$$

Equation 9.54 (the "quasi-square goodness formula") shows that for given values of V_{ph} and L_{ph}, the shape of the current waveform depends on the phase current magnitude I_{ph} and the motor speed S. The current waveform has an acceptable quasi-square shape, providing the product of the phase current and the motor speed does not exceed a value set by the right hand side of equation 9.54. Therefore, by using equation 9.54, limiting values of phase current at particular motor speeds can be calculated: providing these limits are not exceeded, the motor should operate with "good" quasi-square phase current waveforms.

It should be noted that the effective phase inductance during the current rise and fall times, is the sum of the phase winding inductance and the protection inductance. This effective phase inductance should therefore be used in equation 9.54. Similarly, the effective applied phase voltage is actually the inverter output voltage and not the motor terminal phase voltage.

The practical usefulness of equation 9.54 can be assessed by checking it for motor conditions that occurred during the tests. The effective phase inductance in the 7 phase motor system is 1.35mH. During the tests, the motor was found to run at a no-load speed of 7150 rpm, when the load angle was set to 0° and the inverter output voltage was about 10 volts in amplitude. Substituting V_{ph} = 10 volts, L_{ph} = 0.00135 henries, and S = 7150 rpm into equation 9.54 gives:

$$I_{ph} \leqslant 4.662 \quad \text{amps}$$

That is, providing the phase current amplitude is not greater than 4.662 amps, the waveform should have a good quasi-square form. This was certainly so in practice. Fig. 9.17 shows the waveforms for the motor operating on no-load at a 0° load angle and with an inverter supply voltage of ±10 volts. The phase current amplitude is about 1 amp, and so according to equation 9.54, the current waveform should have the desired quasi-square shape: in fact, the waveform is very close to the optimum shape.

The analysis used to derive equation 9.54 assumes that the motor has negligible phase resistances. Speed can then be taken as proportional to the applied phase voltages, and this fact can be used to eliminate the speed and voltage terms from equation 9.54. Rearranging equation 9.54 yields:

$$I_{ph} \leqslant (4.5/L_{ph}).(V_{ph}/S) \quad \text{amps} \qquad 9.55$$

The assumption of negligible phase resistances means that during the steady-state portions of the phase current on periods, the motor back emf, E_{ph}, can be taken as equal to the applied phase voltage, V_{ph}. Hence, V_{ph} can be replaced by E_{ph} in equation 9.55:

$$I_{ph} \leqslant (4.5/L_{ph}).(E_{ph}/S) \quad \text{amps} \qquad 9.56$$

However, E_{ph}/S is the reciprocal of k_{rpm}, the rpm-per-volt constant for the motor discussed in sub-section 9.3.8 (see equation 9.26). Equation 9.56 can therefore be written as:

$$I_{ph} \leqslant 4.5/(k_{rpm}.L_{ph}) \quad \text{amps} \qquad 9.57$$

The right hand side of equation 9.57 is a constant for any given value of L_{ph}, and consequently equation 9.57 shows that the phase current limit value for reasonable waveform squareness is constant over the whole operating speed range. The accuracy of this prediction is open to some doubt because of the assumptions and approximations made in its derivation. Indeed, the prediction is not likely to be strictly true for the 7 phase motor, since it possesses appreciable phase resistance (albeit in the form of the protection component resistors), and so its speed is only approximately proportional to the applied phase voltages. However, substituting a typical value of 864 rpm per volt for k_{rpm} (from sub-section 9.3.8) into equation 9.57, yields a limit value for the phase current of 3.9 amps (with L_{ph} = 1.35mH). This limit value ties in well with the approximate value of current at which the quasi-square waveform distortion started to become appreciable during the tests on the motor. The current level of 3.9 amps is 56% of the IRF230 hexfet's continuous drain current rating (7.0 amps) and 65% of the motor's rated continuous phase current (6.0 amps). It is interesting that the distortion at a given current is not predicted as worsening with increasing speed. This is due to the assumed increasing level of phase voltage available to ramp the phase current up to its new value at the start of each half-cycle, this being an inherent feature of constant volts/frequency operation with the motor.

This matter deserves more detailed examination. Time did not permit such an examination to be carried out, but it is hoped that an indication has been given of an inter-

esting avenue along which the analysis work could be directed in the future.

The accuracies of equations 9.54 and 9.57 are basically limited because of the simplifications made during their derivation. Nevertheless, they are useful design aids, since they enable the range of operating conditions over which "good" quasi-square current waveforms exist to be estimated. If future development work is carried out on the square wave motor concept, it would be worthwhile improving the accuracy of the "quasi-square goodness formula" by deriving it in a more rigorous manner, but this is not justified at present. Much work is needed to fully determine the factors that affect the current waveform shape.

The experiments on the 7 phase motor system have undoubtedly shown that a quasi-square current waveform can be achieved, without the need for any feedback, if the phase inductances of the motor are kept as small as possible. In a practical system where economic considerations may be paramount, the possible reduction in the required current ratings of the inverter switches (made possible by the good utilisation factor of the phase currents) is an attractive benefit of the "square-wave-philosophy". Some deviation from the optimum square-wave must obviously be catered for by proper design of the inverter/motor system. The quasi-square quality formula indicates that the current waveform shape is related directly to the required amplitude of the phase current, and the phase current amplitude is, of course, directly related to the load torque imposed on the motor. The current waveform distortion that occurs as the motor is loaded from zero to full load torque, can be minimised by limiting the phase current magnitude that is required at full-load. A very effective method of reducing the required phase current magnitude at full load torque, is to increase the number of phases, so that the individual phase loading is reduced. Increasing the number

of phase windings also has the effect of reducing the phase inductance, L_{ph}. Reference to the quasi-square quality formula (equation 9.54), shows that the reduction of both the maximum required phase current amplitude and the phase inductance is very beneficial to the phase current waveform shape. This is because the inequality relationship can be satisfied more easily when the phase current and phase inductance are both reduced in magnitude. Hence, although a large number of phases may at first appear to be an expensive proposition, it is in fact a simple way of ensuring good "square-wave" operation over a wide portion of the motor operating range, and this makes the use of an inverter which has minimum-rated-components much more feasible in a practical application.

Therefore, the criterion for choosing the number of phases might well be the "quality" of quasi-square wave current required: a desire to operate the inverter switches close to their current ratings can be more easily achieved as the number of phases increases. Of course, the savings in component costs that may be possible due to the reduced component ratings, may be totally outweighed by the increased number of components required, and further work is needed on this particular aspect of the project, before any firm conclusions with regard to the most economic number of phases can be made.

9.7 Conclusions

In general it is concluded that the experimental testing of the motor was a success and has provided useful information about the 7 phase motor system.

It is fairly evident from the test results that a reconstruction of the 7 phase motor is required, before any further work on the development of the square wave motor system can proceed. The eddy current losses in the

present stator ironwork are far too high.

The appreciable cogging torque, caused by the open slots in the stator, is not really acceptable since very low speed operation is difficult, available load torque at starting is limited, and the motor is rather noisy at high speeds. The cogging could be reduced by careful selection of the stator tooth-widths and slot widths. However, a possibly more useful solution would be to narrow the stator slot openings, (i.e. use semi-closed rather than open slots), since this would increase the phase inductance and so perhaps make the external protection inductors unnecessary.

A rewind of the motor is necessary if the originally specified 3kW rating is ever to be achieved: the incorrect number of phase turns and the lower than expected rotor air-gap flux density have seriously reduced the power output capability.

Nevertheless, the matched quasi-square inverter-motor principle has been successfully demonstrated. The motor operation is stable, and the phase current waveform is very close to the desired optimum shape. Speeds of up to 25000 rpm were reliably achieved during the tests. This is remarkable when it is considered that the rotor was not balanced in any way during its construction. It would be sensible to dynamically balance the rotor before any further tests: this would reduce the vibration and noise that occurs at high speed.

The torque measurements were severely limited by the performance of the eddy current brake dynamometer. In retrospect, it is not surprising that the ability of the dynamometer to load the motor was so disappointing, because the dynamometer was built with the minimum possible amount of design and constructional effort. An improved dynamometer rig is definately required for any future tests.

In spite of some shortcomings in the test results presented in this chapter, there is no doubt that a matched square wave inverter-motor system is feasible, and such a system could be developed into a commercial product in due course.

CHAPTER 10

CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

This thesis has described an unconventional way of achieving a matched inverter-fed synchronous-motor drive system, based on the use of a 7 phase quasi-square wave voltage source inverter feeding a specially designed 7 phase quasi-square wave synchronous motor, with the whole system maintained in synchronism by a microprocessor. In addition, the thesis has presented a sinusoidal analysis of the steady-state characteristics of a voltage-forced auto-piloted synchronous motor, and the accuracy of the analysis, when used to predict the performance of a sinusoidal motor fed from a quasi-square inverter, has been assessed practically on a small motor system.

10.1 Summary of Work on the 7 Phase "Quasi-Square Wave" Motor Drive System

A 7 phase quasi-square synchronous-motor system was designed, built, and tested. The complete system comprised a programmable d.c. voltage supply, a 7 phase inverter, a 7 phase synchronous motor, a microprocessor based control unit, a data-entry keypad, and a rotor position sensor. Each part of the system required some original design and construction work. Several recently developed components, circuits, and materials were very successfully employed in the various units forming the system. For example: hexfet power MOSFET's were used as the switching elements in the 7 phase inverter; a little-known digital speed-tracking circuit was used as the basis for the tachometer; polymer

bonded rare earth magnets were used as the flux source on the rotor of the 7 phase synchronous motor; carbon fibre was used to form a strength ring for the rotor of the 7 phase motor; a Texas TMS9900 microprocessor was used to autopilot the system and to supervise the motor starting, etcetera; optical fibres were used in the rotor position sensor unit.

The operation of the drive system was very encouraging. The phase current waveforms were remarkably close to the desired quasi-square shape, and the microprocessor successfully autopiloted the motor up to a maximum (software limited) speed of just over 25000 rpm. The drive system operated reliably throughout the entire period of the tests reported in the thesis.

Without exception, it can be stated that all parts of the 7 phase quasi-square system performed well.

The 7 phase 3kW hexfet inverter was very reliable in operation. The inverter is not restricted to use within the 7 phase drive system. It forms a versatile multiphase high-frequency facility, with which pulse width modulation work at frequencies in excess of 20kHz could be performed.

The TMS9900 microprocessor system operated without any significant problems. The execution times of the autopiloting software routines imposed a speed limit on the motor of just over 25000 rpm, but this limit can be increased by small changes to the software and hardware. Of great significance is the fact that the microprocessor system did not suffer from any electromagnetic interference. This is obviously a major reason for the reliable operation of the system.

The successful operation of the 7 phase motor, at speeds up to just over 25000 rpm, has proved that a 2 pole rotor, having pole-face mounted permanent magnets, can be

constructed using very simple techniques and yet have the necessary mechanical strength. Other aspects of the motor design, notably the stator laminations, were less than satisfactory and resulted in the power output being well below the originally specified 3kW, but the general concept of a square wave synchronous motor was demonstrated.

Both the keypad and the digital tachometer functioned without any problems despite the relative complexity of their respective circuits.

The keypad received extensive use during the motor tests and it was found that the microprocessor responded to all keypad commands or data entries, regardless of the speed at which the motor was being autopiloted. That is to say, the microprocessor was never totally occupied by the autopiloting task, and so communication between the system operator and the system was always assured.

The tachometer provided a 16 bit digital speed signal over the range 16 rpm to 65535 rpm in 1 rpm steps, as well as a decimal display of the speed value. The response of the tachometer was excellent. The principle upon which the tachometer circuit is based has certainly been proved.

Finally, the rotor position detector unit for the 7 phase motor worked very well and provided rotor position information with a resolution of 6.43°. The relatively cheap polymer optical-fibres that were used as the basis of the position sensor system were found to be very robust. The logic signals obtained from the position sensor unit were free from electromagnetic interference, and the frequency responses of the optical photo-amplifiers were more than sufficient to enable successful autopiloting of the motor up to the microprocessor-limited maximum speed. At all speeds the signals from the position sensor unit enabled the microprocessor to maintain the motor and inverter in a synchronised state to a precision of 6.43°.

10.2 Summary of the Work on the Voltage Forced Characteristics of an Autopiloted Synchronous Motor

The operating characteristics for an autopiloted sinusoidal synchronous motor, operated in a voltage-forced mode, have been presented. The effect of both phase inductance and resistance was included in the analysis from which the characteristics were obtained. The analysis indicates that the performance characteristics of a voltage forced autopiloted synchronous motor (with a cylindrical or salient pole rotor) can be drastically altered simply by adjusting the chosen setting of the motor load angle. In addition, at a given motor speed there is an optimum load angle setting which produces a maximum motor torque per applied phase volt.

The effect of load angle setting, on the performance characteristics of a voltage-forced autopiloted synchronous motor, was checked by making practical measurements on a small drive system. The system was based on a sinusoidally distributed 3 phase magslip machine fed by a quasi-square wave voltage source inverter. The practically determined characteristics compared well with the theoretical characteristics, which were based on the fundamental component of the applied quasi-square phase voltage waveform.

Methods of adjusting the load angle of an autopiloted synchronous motor have been discussed, and various rotor position detector systems have been compared. Electronic circuits capable of selecting several discrete load angle settings were designed, built, and tested. They were found to work reliably and accurately, and in some senses formed "electronic gearboxes" for the drive system.

10.3 Conclusions Drawn from the Work on the 7 Phase Motor Drive System

The work on the 7 phase motor system has clearly demonstrated that a matched quasi-square inverter/synchronous-motor system can be realised by appropriate design of the various system components. Many of the features incorporated into the system have been found to be useful and worthwhile.

It is concluded from the work that a microprocessor can be used to autopilot a synchronous motor drive system up to very high operating speeds (25400 rpm), whilst maintaining close control over the relative positions of the stator current pattern and the rotor field flux (to within 6.43^{o}). The initial target speed of 30000 rpm is possible if small software and hardware changes are made to the system. The work has shown that a microprocessor is a very powerful logical unit with which it is possible to easily set up (and modify) logical tasks such as the motor starting sequence, and so forth.

A very important finding from the work on the 7 phase motor is that a reasonable quasi-square back-emf waveform can be obtained relatively easily, without the need for complex rotor-flux/stator-winding design calculations. The very good quasi-square phase current waveforms that were observed during the motor tests were a direct result of the natural system behaviour: over a wide range of load and speed conditions the system naturally operates with the optimum phase current waveform. It is concluded that the desired phase current waveforms can be achieved without the need for feedback networks or pulse width modulation techniques. In general, it is felt that a "square wave" system has much in its favour, especially because it enables good utilisation of the inverter switching devices to be achieved.

The system tests have shown that the hexfet inverter

unit is reliable and performs as required. The measured switch-on time for the hexfets of about 60ns was well within the manufacturer's typical switch-on time, and so it is reasonable to conclude that the simple drive circuits for the hexfets have a performance in excess of that actually required. The absence of any hexfet failures in "strange circumstances" is a good indication that they are robust devices, and it supports the manufacturer's claim that hexfets do not require any snubber protection.

There are several general conclusions that are worth mentioning at this point. The system has demonstrated that optical fibres can be employed usefully in a rotor position sensor unit. Whilst there was no particular need to use optical fibres in the 7 phase motor system, it is clear that they will be increasingly used in motor drive systems because of the benefits of reduced electromagnetic pick-up and good isolation that can be achieved by their use. The unusual tachometer circuit employed in the system has shown that a fast response digital tachometer, which provides accurate and frequent speed readings, is possible. The tachometer is stable and requires relatively little logic to implement it. The successful use of the keypad control unit has demonstrated that a simple input peripheral is more than adequate for the control of a motor drive system. The limited number of keys on a keypad minimises the chances of an erroneous data/command entry, and yet because the keys have functions related specifically to the requirements of a motor drive, it is possible for the operator to have full control over the drive system. Undoubtedly such keypad controllers will become very common on industrial drive systems.

10.3.1 Problem Areas Identified During the Tests on the 7 Phase Motor System

As one might expect, a number of problem areas were identified during the tests on the system. The principal

problems were: the motor speed limit imposed by the microprocessor software; the excessive eddy current losses in the stator laminations; the fairly high cogging torque; the slight mismatch between the applied motor phase voltage and motor phase back-emf waveforms at the leading and trailing edges.

The limit on the maximum motor speed was a direct result of the execution times of the autopiloting software; at the maximum attainable speed, the autopiloting interrupts occur so frequently that it is possible for an interrupt to be missed, because of the way the software instructions were arranged. Simple modifications to the autopiloting routines and the hardware have been suggested in Chapter 4, and there is good reason to believe that the original target speed of 30000 rpm is possible if the suggested modifications are implemented.

The eddy current losses in the stator laminations were to some extent expected because rather thick (25 thou") lamination material was used. The eddy current brake effect generated by the rotor flux sweeping through the stator laminations produced an unacceptably large loss-torque on the rotor. It is, however, encouraging to note from the test results that there are no stray losses within the motor, and so as it should be possible to reduce the stator eddy current losses, it is almost certain that the available motor output torque can be greatly increased, thus improving the overall motor efficiency. The obvious method of reducing the stator eddy current losses would be to reduce the lamination thickness and use a better grade of steel. The use of an electrically non-conducting magnetic material, such as ferrite might be worth considering.

The large cogging torque of the motor (0.271Nm) is a direct result of the open slot geometry of the stator. Open slots were chosen to help minimise the motor phase inductance so that the rise time of the phase current was

short. However, the mismatch between the applied phase voltage and the phase winding back-emf meant that, in practice, an inductance much greater than the phase inductance of 100μH was required to prevent large phase current spikes, and so it was necessary to use the external "protection components" described in the thesis. It is quite probable that the problem of high cogging torque and the need for protection components could be jointly dealt with, to some extent, by using closed slots in the stator. This would remove (or certainly greatly reduce) the cogging effect, and the increase in phase inductance due to the closed stator slots would go some way to removing the need for the protection components. However, it is not necessary to resort to closed slots if the need for a low phase inductance remains paramount, because the cogging torque could almost certainly be significantly reduced by slight skewing and/or carefully adjusting the relative dimensions of the stator slot pitch and rotor pole arc at the design stage. (Obviously any skewing of the stator would affect the "quality" of the generated quasi-square back-emf waveform to some extent, but very little skewing should be necessary to considerably reduce the peak cogging torque. Hence, slight skewing is well worth future consideration.)

The mismatch between the applied motor phase voltage and phase back-emf waveforms is unfortunate because it does result in the requirement for the protection components. Some improvement in the back-emf waveform may be possible by careful design of the rotor flux pattern to reduce fringing flux. The effort needed to accurately calculate the rotor flux distribution would be very worthwhile, because any improvement in the back-emf waveform would enable some reduction in the values of the protection components to be made (ideally, the improvement would be such that the protection components would be no longer necessary).

10.3.2 Ways in Which the System Could be Simplified or Modified to Make it Practical and Cost Effective

Having discussed the problems that were of greatest concern during the tests, it is appropriate at this stage of the conclusions to consider how the system might be simplified or modified to obtain a practical cost effective system. Some comment is worthwhile with regard to the protection components, the microprocessor, the hexfets, the position sensor, and the inverter power supply.

10.3.2.1 The Protection Components

The fact that protection components were necessary between the inverter and the motor was really the result of inadequate system design during the early stages of the project. The necessary extra inductance and resistance was simply added between the inverter and motor to avoid having to redesign the stator. In a commercial system, one might reasonably expect that all the necessary inductance and resistance (needed for the safe operation of the system) would be incorporated into the motor winding. Nevertheless, in a minimum-motor-impedence design, there are sound engineering reasons why the use of external components to form the bulk of the winding impedance is attractive: namely, that the resistive power loss due to the total phase resistance is mainly dissipated outside the motor body and the "iron" loss due to the magnetic field in an external inductor can be made negligible by the use of an air-cored or ferrite-cored design.

10.3.2.2 The Microprocessor Control Unit

Following the tests on the 7 phase motor system, there is no doubt that a microprocessor can be used very successfully to autopilot an inverter-fed synchronous motor at high speeds. There is, however, great scope for employing microprocessors in motor drive systems to perform tasks

other than the autopiloting function, and so in the development of a commercial autopiloted synchronous motor drive from the 7 phase system, it would be important to decide exactly what to use the microprocessor for. All of the control and monitoring tasks in the 7 phase motor system can be achieved by conventional analogue and digital circuits, but it makes technical and economic sense to try to replace as many of the discrete circuits by one microprocessor, since such an approach reduces the component count and should improve reliability. The only disadvantage of using a single-microprocessor to replace as much of the discrete circuitry as possible, is that the system is obviously seriously disabled (perhaps dangerously) if the microprocessor fails in any way. However, microelectronics tend to be very reliable in use, and so this aspect of single microprocessor operation is not as serious as it might at first appear. A single-microprocessor system in which discrete circuits play an insignificant role in the control functions can aptly be termed a "pure" system.

In the case of the 7 phase motor system, the single TMS9900 microprocessor was unable to autopilot the motor up to the desired maximum speed of 30000 rpm, due to the execution times of the autopiloting interrupt service routines. It is fair to say, however, that in all other respects the microprocessor performed well, and the fact that the motor was successfully autopiloted up to a speed of 25400 rpm was a significant achievement for a first attempt. Indeed, it is possible that a speed of 30000 rpm might be attainable by the 7 phase motor system if a few minor software and hardware changes suggested in Chapter 4 are made. Nevertheless, it seems fairly safe to assume that at present, a single-microprocessor would be hard pressed to perform all the required tasks in a developed version of the 7 phase motor system. This assumption is based on the fact that additional functions (such as speed regulation) would be required in a commercial system, and it seems unlikely that these could be performed by any of the pre-

sently available standard microprocessors, because the autopiloting function demands so much of the computation time at high motor speeds. Obviously, as faster microprocessors become available it will be easier to implement "pure" single-microprocessor systems in which discrete circuits are at a bare minimum.

At present, in situations such as the 7 phase motor system where a single-microprocessor cannot perform all of the required tasks at the desired level of performance, there are three alternative strategies available.

The first alternative is to abandon the use of a microprocessor and perform all functions by discrete circuitry: that is, a "hardwired" approach. However, this approach is rather drastic and it does mean that the jobs that a microprocessor can do very well (such as system health checks, etcetera) would have to be performed by dedicated hardware.

The second alternative is to use discrete logic for those tasks which the microprocessor cannot manage or is least suited to. This approach produces a "hybrid" system since it combines a microprocessor with function-units made of discrete circuitry. The autopiloting task is relatively simple and yet because it occurs so frequently it does tend to monopolise the computational time of the microprocessor. By arranging discrete logic to perform the autopiloting task, it is possible to remove any software based speed restrictions from the motor. The microprocessor can then be used to perform the more involved but less frequent control operations such as the motor start procedure. It is of course very useful to be able to use the microprocessor for the autopiloting task, because it enables complex changes to the inverter control signals to be made simply by software modifications, but this feature is useless if the microprocessor cannot perform the basic autopiloting task fast enough. Hence, a very logical

development of the 7 phase motor system using a hybrid approach, is to base the system around the microprocessor with the autopiloting function dealt with by discrete logic. The microprocessor deals with all other tasks including motor start sequences , speed regulation, system housekeeping, data display, etcetera. If necessary, the discrete logic might be microprocessor-programmable to enable changes in the autopiloting function to be selected.

The third alternative, and one which is becoming increasingly mentioned in the literature, is to use more than one microprocessor: that is, a "multi-processor" system. The number of microprocessors required depends on the functions that have to be executed. For example, if each task requires virtually constant "attention" from a microprocessor, it would be sensible to allocate one microprocessor per task. This would significantly simplify the software related to a particular task, because there would be no need to make allowance for other tasks interrupting the execution of the software. A one-microprocessor-per-task approach might seem expensive, but as microprocessor costs come down further, it will become increasingly attractive.

Of the three alternatives to the "pure" single microprocessor system, it is felt that a multi-processor approach is the most attractive. With specific regard to the organisation of the 7 phase motor system, it is felt that a two microprocessor arrangement would be satisfactory: one dealing solely with the autopiloting; the other dealing with all other tasks such as starting, speed regulation, keypad decoding, etcetera. It is not thought that any problems would be encountered with regard to communication between the two microprocessors: the philosophy would be to arrange the dedicated autopiloting microprocessor as an independent logic unit which would do as it is "told" by the main system microprocessor. The microprocessors would have independent memory areas.

On the question of the amount of interfacing required between the microprocessor and the peripherals it communicates with, it is certainly true that a microprocessor other than the TMS9900 could have significantly simpler interfacing requirements. The serial transfer of data between the TMS9900 and the TMS9901 I/O port is a serious drawback of the Texas system. Most microprocessor systems are available with very useful and versatile interface ports, and so it is probable that significant reductions in the amount of discrete interfacing logic could be achieved in any development of the 7 phase system. Indeed, the increasing availability of microprocessors having on-chip memory and I/O facilities, makes the idea of a multi-processor control system requiring virtually no extra components very feasible in the future. It is concluded from the microprocessor work carried out on the 7 phase motor system, that microprocessors do have a very important role to play in motor drive systems, and they should be used in any development of the 7 phase motor.

10.3.2.3 The Inverter

The use of hexfets in the inverter of the 7 phase motor system was a complete success. The hexfets operated reliably throughout the entire period of the tests, and although at present hexfets are more expensive than bipolar transistors or thyristors, the low-power gate-drive requirements of hexfets does much to compensate for this.

There is significant scope for improvement of the inverter with regard to the physical size of the circuit boards. The present inverter was designed so that components were readily accessible for repairs. In a developed commercial inverter there would not ideally be any need for repairs, and so the whole set of circuits could be squeezed into a much smaller volume.

With reference to the drive circuits of the hexfets,

it would be technically and economically more sensible to use integrated circuit gate drivers in a commercial system. This would reduce initial assembly costs, improve reliability, help to reduce the circuit volume, and aid fault diagnosis. It can only be a matter of time before MOSFET manufacturers incorporate a suitable drive circuit within MOSFET cases: if and when this occurs, it will be possible to obtain optimum switching performance from the MOSFET devices even when they are controlled by signals that have poor rise and fall times.

The parasitic transistors within the hexfets functioned well as the inverter freewheel diodes but they are relatively slow devices (the "diode" within a IRF230 hexfet has a typical reverse recovery time of 650ns). Therefore, it might be prudent in any future work to use discrete high-speed epitaxial diodes connected across the hexfets to form the freewheel paths.

In general it is felt that power MOSFETs are good devices to use in a drive system because they are robust switches and they require so little protection. It is almost certain that the available ratings of MOSFETs will continue to increase for some time to come. The increased ratings, and the ease with which power MOSFETs can be connected in parallel or series seems almost certain to lead to a situation where power MOSFETs will be the dominant switch devices used in the low and medium power sectors of the power electronics market.

10.3.2.4 The Rotor Position Detector Unit

The optical position sensor unit used on the 7 phase motor system worked very reliably, and it clearly demonstrated that optical fibres can be used very successfully in rotor position detection applications. However, a major problem with an optical type of position sensor is that its performance can be affected if it operates in a dirty

environment. Even the build up of a thin layer of dust, on any of the surfaces through which the light is transmitted, can be enough to stop an optical position detector functioning. The intervals at which an optical position sensor might require some form of cleaning, to maintain its performance, can be extended by totally enclosing, in a clean-air environment, the surfaces which transmit light. Nevertheless, a position sensor which is insensitive to dirt on its active surfaces is a more attractive proposition (on practical grounds) for a commercial product.

Hence, it is felt that an alternative position sensor system might well be more appropriate for the 7 phase motor system, should it ever be developed into a practical, cost-effective system. Position sensors based on the hall effect are widely used in industry, and they provide good resolution and high reliability at relatively low cost. Therefore, in the event that a cheap, sealed, optical position sensor is not feasible, it may well be that a hall effect position sensor should be incorporated into the 7 phase motor system in any future development.

10.3.2.5 The Inverter Power Supply Arrangement

The power supply arrangement used for testing the hexfet inverter is in no sense appropriate for a real drive system. Ideally, what is required is a cheap, robust, variable voltage supply. The fact that the phase currents of the 7 phase motor are interlaced, with the result that the inverter supply rail currents are essentially constant for a given load, is very useful, because it means that each supply rail of the inverter forms an ideal load for a chopper type of variable voltage power supply. Chopper power supplies based on thyristor switches are commonly used in d.c. motor drive systems, and at present are probably the best form of power supply to employ in any development of the 7 phase motor system.

10.3.3 Viability of the "Square Wave" Motor Drive Concept

It is felt that a practical and reliable 7 phase motor drive system could be developed in the light of the experience gained during the work described in this thesis. The main stumbling-block to any future development work, appears to be the reluctance of drives manufacturers to consider anything other than standard 3 phase systems, in which all efforts to match the motor and inverter are concentrated on the generation of sinusoidal waveforms by the inverter. Fortunately, there have been signs over recent years that unusual machine designs are becoming more widely acceptable, and a good example of this is the way that multi-phase stepper motors have rapidly been developed for use in various positioning-drive applications. It is hoped that if the basic simplicity of the square wave drive system becomes more widely appreciated, there will be a positive response from the drives industry and further development of the idea will take place.

It is interesting to note that a "pure" quasi-square wave system, in which both the inverter and motor produce quasi-square waveforms, can be very simply controlled by digital circuits (such as microprocessors) because action is only required at discrete points in every revolution, no matter what the speed of the motor is. A "pure" sinusoidal system, in which both the inverter and motor produce sinusoidal waveforms, is not so easily controlled by digital means, since continuous (or very frequent) information is necessary from a rotor position sensor, in order to generate a sinusoidal output voltage waveform from an inverter. As microprocessors and digital electronics become more widespread in all walks of life, it is possible that the simplicity of the square-wave system will become evident to motor drive manufacturers and their customers.

With regard to the practical viability of a quasi-square wave motor drive system, it is clear from the test

results and the ideas presented in this chapter for improvements to the system, that a reliable and efficient motor drive could be developed. The commercial viability of such a system is less clear at present. The system is unconventional and further work is necessary comparing the costs of the system as built, cheapened versions of the system, and "conventional" high speed systems before conclusions on this point can be reached. The costs of power MOSFETs, carbon fibre, polymer bonded rare earth magnets, and microprocessors are almost certain to fall as their usage in industry increases, and so a motor drive based on the system described in this thesis should become commercially more viable in the future. Of course no high speed drive need be restricted to high speed applications: low speed applications can be catered for by using reduction gearboxes. The cost, reliability, and efficiency of modern gearboxes are sufficiently favourable to make this development feasible and worthwhile.

10.4 Conclusions Drawn from the Work on the Voltage-Forced Characteristics of an Autopiloted Synchronous Motor (Magslip), and Suggestions for Future Work

The work on the effect of load angle, on the voltage-forced characteristics of an autopiloted synchronous motor, has clearly shown the effect of load angle setting on performance. The tests on the magslip system have verified the general performance trends indicated by the sinusoidally based analysis. It is therefore concluded that load angle control can be used to modify the characteristics of an autopiloted voltage-forced synchronous motor as required. The good agreement between the measured performance results of the magslip and the characteristics predicted by the sinusoidal analysis is also interesting. From this, one can conclude that the sinusoidal analysis can be used to obtain reasonably accurate predicted performance characteristics for a sinusoidally distributed synchronous motor,

even when it is fed from a voltage-source quasi-square wave inverter as the magslip was. However, it is suggested that in the future the analysis should be extended to take due account of the quasi-square applied voltage waveforms.

From the work on the electronic load angle adjustment circuits, it is concluded that load angle settings can be varied reliably and accurately without the need to resort to mechanical adjustments of the rotor position detector unit. The work has demonstrated that either discrete logic or a microprocessor can be used to control the load angle adjustment. The circuitry needed to implement adjustable load angle operation is so simple that there can be no doubt such systems are commercially viable. An obvious future development of the work should be the implementation of a microprocessor based system, which can set the load angle to its optimum value at any motor speed. There does not appear to be any reason why load angle adjustment facilities should not be a standard feature on voltage-forced autopiloted synchronous motors in the future.

10.5 Suggestions for Future Work on the 7 Phase "Quasi-Square Wave" Motor

The original aim of the 7 phase motor drive project was to design, build, and test a complete quasi-square wave synchronous motor system. The system was intended to comprise a novel synchronous motor, a MOSFET inverter, a microprocessor based controller, an optical rotor position detector, and a chopper based power supply for the inverter. The fact that only the chopper power supply had to be omitted from the project work is a clear indication that the aims of the project were almost completely achieved, although it is unfortunate that the test program was to some extent incomplete, owing to the shortcomings of the dynamometer.

MOSFETs, carbon fibre, and polymer bonded rare earth magnets have all been employed to great advantage in the motor system, and the quasi-square wave basis of operation has been shown to be feasible. The work has also shown that a microprocessor is a useful logic element to employ in the synchronising loop of an autopiloted synchronous motor, and it should be retained in such a role if at all possible.

It is felt that any continuation of the work reported in this thesis should be concerned with the development of a practical drive system from the present system. The aim should be to reduce the number of electronic components required and to minimise the total volume of the units making up the drive. The textile and aerospace industries are two obvious markets towards which the development work could be directed.

Some work is required in the future on all parts of the 7 phase motor system including the motor, the rotor position detector, the inverter and its power supply, and the control electronics.

Future work on the motor should be principally concentrated on the stator. The stator must be rebuilt and possibly redesigned using either thinner laminations of better steel or even solid ferrite, in order to reduce the rotor-induced eddy current losses. It is thought that semi-closed stator slots should be chosen for the new stator in an attempt to increase the phase inductance. By suitable redesign of the stator and the phase windings, it should be possible to increase the phase impedance sufficiently to enable the inverter protection components to be dispensed with. The use of semi-closed stator slots should also help to reduce the cogging torque of the motor. Some effort should be directed in the future towards obtaining an accurate plot of the rotor airgap flux density. A knowledge of the airgap flux distribution will enable the rotor design

to be altered, in order to reduce the cogging torque and improve the waveform of each phase winding back-emf. Some attention must also be devoted to mechanical aspects such as the dynamic balancing of the rotor and the choice of reliable low-friction bearings capable of sustained operation at high speeds.

Work in the future on the position detector system should be aimed at producing a compact and reliable unit. It is thought that a position sensor based on hall-effect devices might be worth investigation.

There is remarkably little that needs to be done to the inverter. Work on the inverter should primarily be directed towards reducing its overall size and cutting the number of drive circuitry components. However, much work is required on the variable voltage supplies for the inverter. It is felt that a thyristor-based chopper unit should be designed, built, and evaluated for use on the 7 phase motor system. Ideally, the chopper unit should be able to operate directly from the mains supply (via a standard rectifier and smoothing arrangement).

Future work on the control electronics of the 7 phase system should be aimed at minimising the number of integrated circuits required in order to reduce the size, cost, and power consumption of the control unit. It is felt that the most fruitful path to follow is to develop a multiprocessor system. By careful design of both hardware and software it should be possible to evolve a basically simple control unit capable of operating the 7 phase motor system over the required speed range. The use of "one-chip" microprocessors would enable a great reduction in the component count to be achieved. Both the keypad and tachometer are important elements of the control unit for some applications and both could be significantly improved by further work. The keypad logic circuits could be completely replaced by a single microprocessor plus a few ancilliary

integrated circuits. The tachometer circuits could be either integrated into a single large-scale-integration circuit or replaced by a microprocessor based arrangement. The scope for work on the whole control unit is enormous and it is in this area of work that the greatest reductions in cost and complexity can be achieved.

With regard to the performance tests on the 7 phase motor, it is essential that any future work should include the construction or purchase of a suitably rated dynamometer unit, to enable a comprehensive set of steady-state motor performance characteristics to be obtained. In addition, some effort should be directed at obtaining a set of dynamic motor test results, to enable the stability of the motor system to be assessed. A long term objective if the work is continued, should be the development of a simple analytical model of the motor system, to enable the performance characteristics of the motor to be predicted.

There are several variations in the operation of the 7 phase motor that are worthy of investigation in the future. One that is especially easy to implement on a microprocessor-based system is the control of the number of active phases: the idea behind this strategy is to disable several of the phases when the motor is lightly loaded in an attempt to reduce inverter switching losses, etcetera. Another interesting variation in the operation of the basic 7 phase motor system is to arrange the controlling logic so that it actively tries to maintain a good quasi-square phase current waveform: to achieve this the load angle of the motor must be varied as necessary to keep the phase current waveform as close to the optimum as possible, and such a task is almost certainly best dealt with by a microprocessor.

There is clearly no shortage of work that can be performed on the 7 phase "square wave" motor system in the future. It is the author's opinion that the suggestions

for future work are well justified by the successful operation of the present system, as reported in this thesis.

10.6 Supplement to List of References

During the writing of this thesis several interesting publications on power MOSFETs have come to the notice of the author. Forsythe (10.1) has published details on the technique of paralleling power MOSFETs to produce a composite switch with a higher power rating, and Clemente et al (10.2) have written a paper explaining the switching performance of power MOSFETs. International Rectifier have published a hexfet databook (10.3) which contains useful design advice on how to use hexfets.

APPENDIX 5A.

IRF230
IRF231
IRF232
IRF233

N Channel

200 Volt, 0.4 Ohm HEXFET™

The HEXFET™ technology is the key to International Rectifier's advanced line of power MOSFET transistors. The efficient geometry and unique processing of the HEXFET design achieve very low on-state resistance combined with high transconductance and great device ruggedness.

The HEXFET transistors also feature all of the well established advantages of MOSFETs such as voltage control, freedom from second breakdown, very fast switching, ease of paralleling, and temperature stability of the electrical parameters.

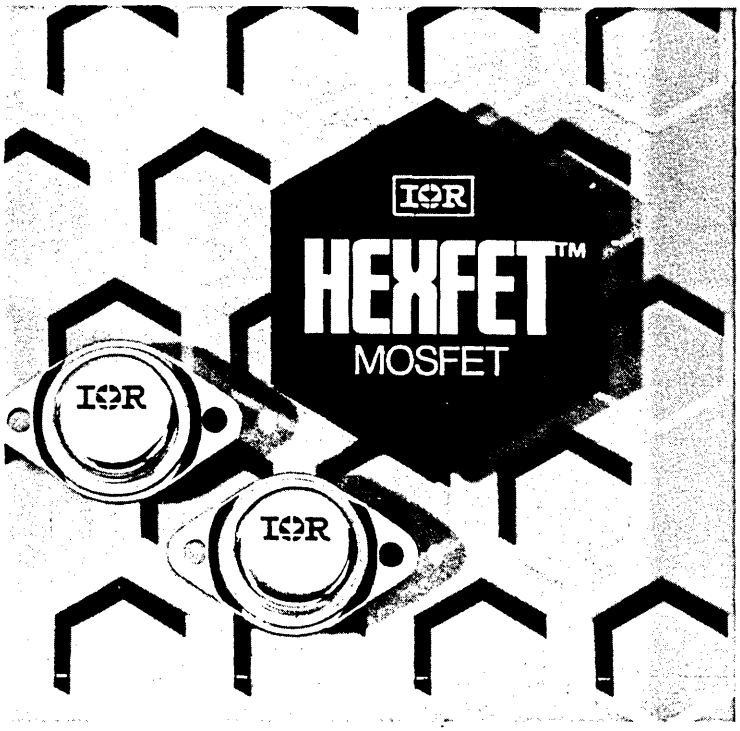
They are well suited for applications such as switching power supplies, motor controls, inverters, choppers, audio amplifiers, and high energy pulse circuits.

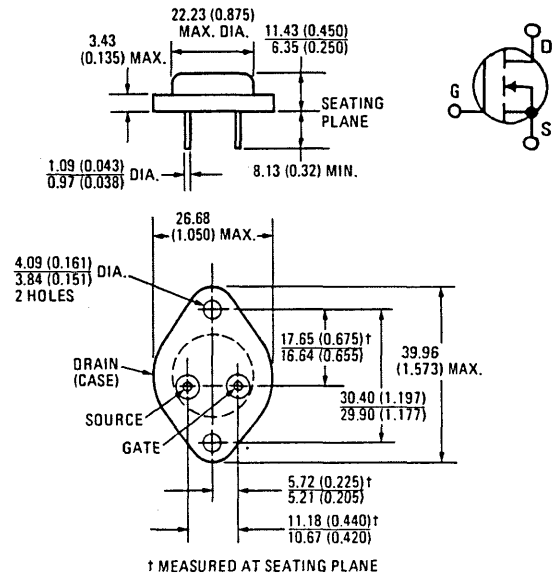
Features:

- Fast Switching
- Low Drive Current
- Ease of Paralleling
- No Second Breakdown
- Excellent Temperature Stability

Product Summary

Part Number	V_{DS}	R_{D} (on)	I_{D}
IRF230	200V	0.4Ω	7A
IRF231	150V	0.4Ω	7A
IRF232	200V	0.6Ω	6A
IRF233	150V	0.6Ω	6A





Conforms to JEDEC Outline TO-204AA (TO-3)
Dimensions in Millimeters (and Inches)

MOSFET TRANSISTORS – DATA SHEET NO. PD-9.306B
REVISED MAY, 1980

Absolute Maximum Ratings

Parameter		IRF230	IRF231	IRF232	IRF233	Units
V_{DS}	Drain – Source Voltage	200	150	200	150	V
V_{DGR}	Drain – Gate Voltage (R_{GS} = 1 MΩ)	200	150	200	150	V
I_D @ T_C = 91.0°C	Continuous Drain Current	7	7	6	6	A
I_{DM}	Pulsed Drain Current	15	15	12	12	A
V_{GS}	Gate – Source Voltage	±20				V
P_D	Max. Power Dissipation	75 (See Fig. 11)				W
	Linear Derating Factor	0.6 (See Fig. 11)				W/K
I_{LM}	Inductive Current, Clamped	(See Fig. 1 and 2) L = 100 μH				A
		15	15	12	12	
T_J T_{stg}	Operating and Storage Temperature Range	-55 to 150				°C
	Lead Temperature	300 (0.063 in. (1.6mm) from case for 10 sec.				°C

Electrical Characteristics @ T_C = 25°C (Unless Otherwise Specified)

Parameter		Type	Min.	Typ.	Max.	Units	Conditions
BV_{DSS}	Drain – Source Breakdown Voltage	IRF230 IRF232	200	–	–	V	V_{GS} = 0V
		IRF231 IRF233	150	–	–	V	I_D = 1.0 mA
$V_{GS(th)}$	Gate Threshold Voltage	ALL	2.0	–	4.0	V	V_{DS} = V_{GS}, I_D = 1 mA
I_{GSS}	Gate – Body Leakage	ALL	–	–	100	nA	V_{GS} = 20V
I_{DSS}	Zero Gate Voltage Drain Current	ALL	–	0.1	1.0	mA	V_{DS} = Max. Rating, V_{GS} = 0V
			–	0.2	4.0	mA	V_{DS} = Max. Rating, V_{GS} = 0V, T_J = 125°C
$I_{D(on)}$	On-State Drain Current	IRF230 IRF231	7.0	–	–	A	V_{DS} = 25V, V_{GS} = 10V
		IRF232 IRF233	6.0	–	–	A	
$R_{DS(on)}$	Static Drain-Source On State Resistance	IRF230 IRF231	–	0.25	0.4	Ω	V_{GS} = 10V, I_D = 3.5A
		IRF232 IRF233	–	0.4	0.6	Ω	
g_{fs}	Forward Transconductance	ALL	2.5	4.5	–	S	V_{DS} = 25V, I_D = 3.5A
C_{iss}	Input Capacitance	ALL	–	700	900	pF	V_{GS} = 0V, V_{DS} = 25V, f = 1.0 MHz (See Fig. 10)
C_{oss}	Output Capacitance	ALL	–	250	450	pF	
C_{rss}	Reverse Transfer Capacitance	ALL	–	80	150	pF	
$t_{d(on)}$	Turn-On Delay Time	ALL	–	30	50	ns	I_D = 4A, E_1 = 0.5 BV_{DSS} (See Fig13) T_J = 125°C (MOSFET Switching times are essentially independent of operating temperature.)
t_r	Rise Time	ALL	–	70	140	ns	
$t_{d(off)}$	Turn-Off Delay Time	ALL	–	50	100	ns	
t_f	Fall Time	ALL	–	70	140	ns	

Thermal Resistance

R_{thJC}	Junction-to-Case	ALL	–	–	1.67	K/W	
R_{thCS}	Case-to-Sink	ALL	–	–	0.2	K/W	Mounting surface flat, smooth, and greased.
R_{thJA}	Junction-to-Ambient	ALL	–	–	30	K/W	Free Air Operation

Body-Drain Diode Ratings and Characteristics

I_{DR}	Continuous Reverse Drain Current	IRF230 IRF231	–	–	7	A	Modified MOSFET symbol showing the integral reverse P-N junction rectifier.
		IRF232 IRF233	–	–	6		
I_{DRM}	Pulsed Reverse Drain Current	IRF230 IRF231	–	–	15	A	
		IRF232 IRF233	–	–	12		
V_{SD}	Diode Forward Voltage	IRF230 IRF231	–	–	1.45	V	T_J = 25°C, I_F = I_{DRM}, V_{GS} = 0V
		IRF232 IRF233	–	–	1.31	V	
t_{rr}	Reverse Recovery Time	ALL	–	650	–	ns	T_J = 150°C, I_F = I_{DRM}, dI_F/dt = 100 A/μs
Q_{RR}	Reverse Recovered Charge	ALL	–	10.0	–	μC	T_J = 150°C, I_F = I_{DRM}, dI_F/dt = 100 A/μs

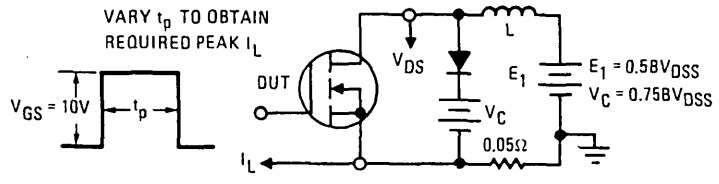


Fig. 1 – Clamped Inductive Test Circuit

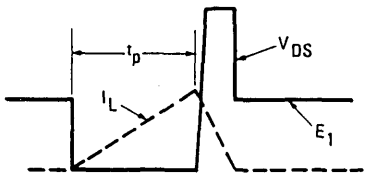


Fig. 2 – Clamped Inductive Waveforms

APPENDIX 5A CONTINUED.

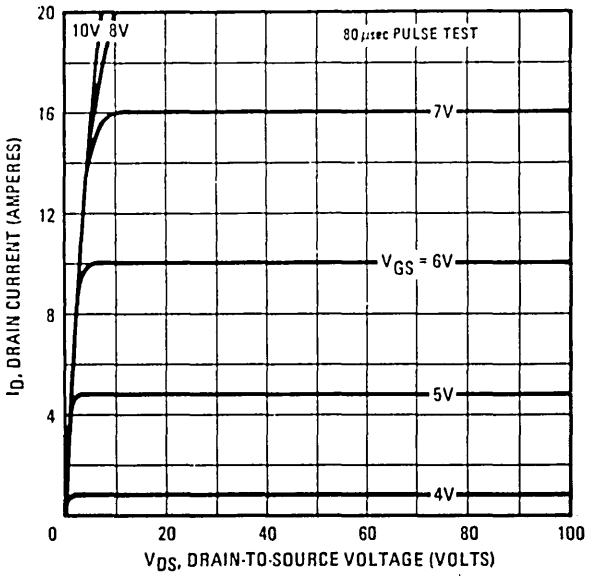


Fig. 3 – Typical Output Characteristics

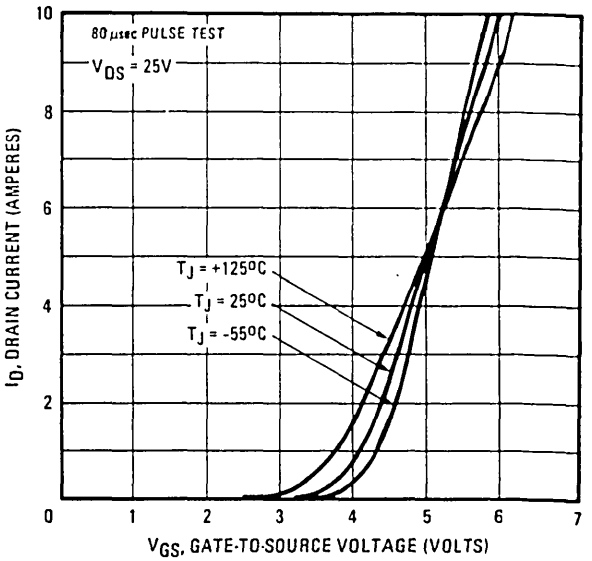


Fig. 4 – Typical Transfer Characteristics

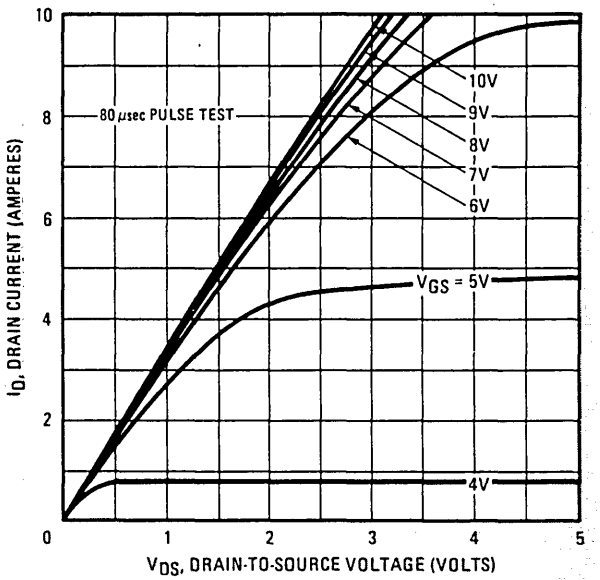


Fig. 5 – Typical Saturation Characteristics (IRF230, IRF231)

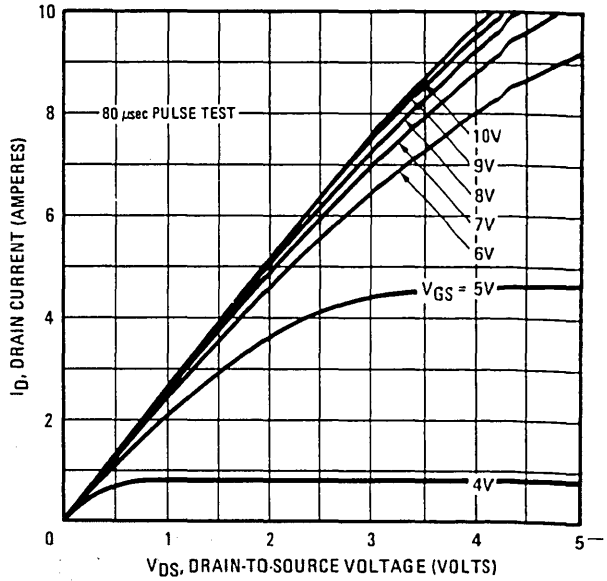


Fig. 6 – Typical Saturation Characteristics (IRF232, IRF233)

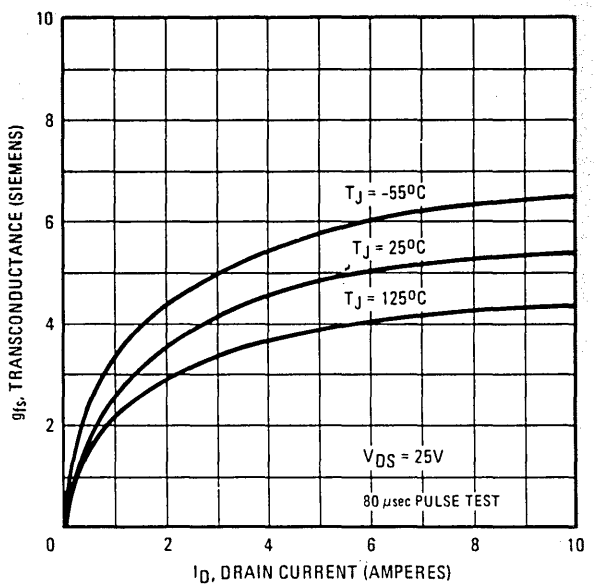


Fig. 7 – Typical Transconductance Vs. Drain Current

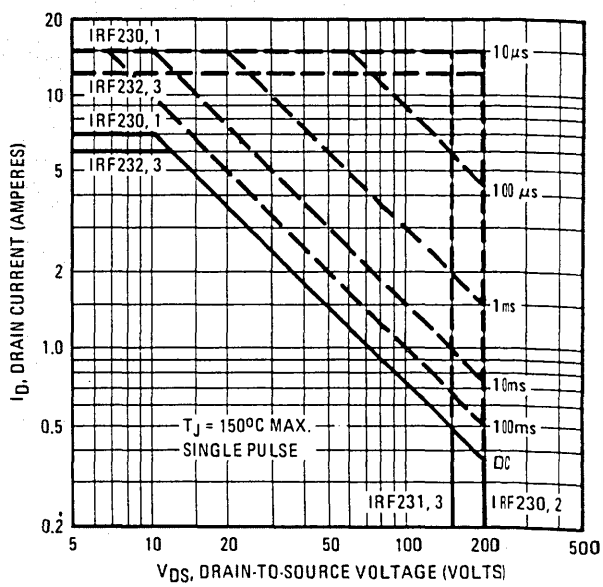


Fig. 8 – Maximum Safe Operating Area

APPENDIX 5A CONTINUED.

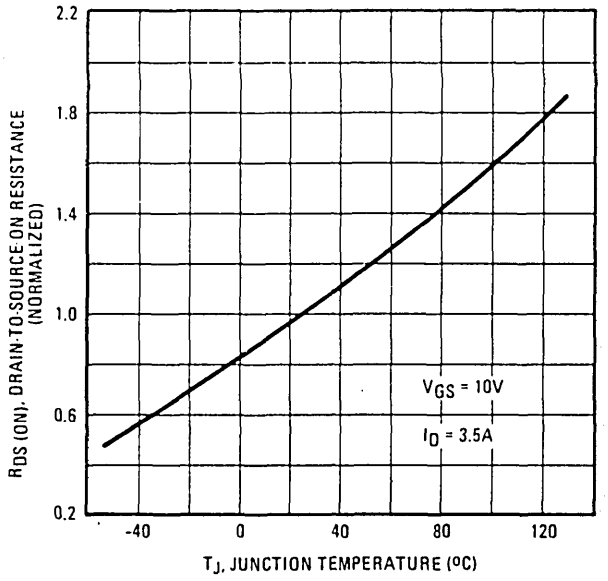


Fig. 9 – Normalized On-Resistance Vs. Temperature

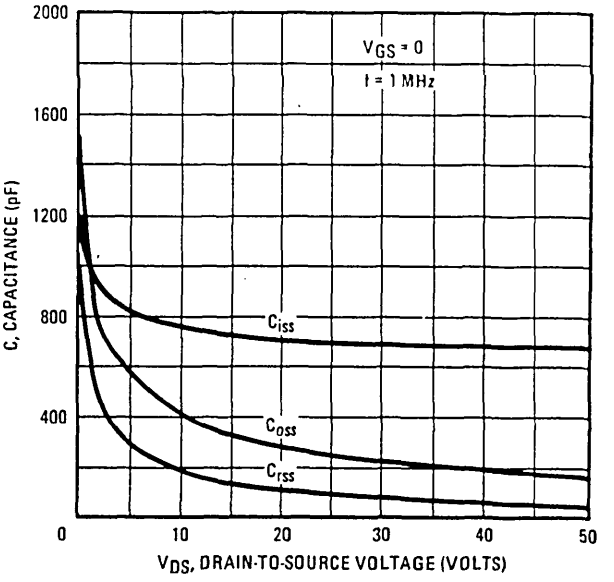


Fig. 10 – Typical Capacitance Vs. Drain-to-Source Voltage

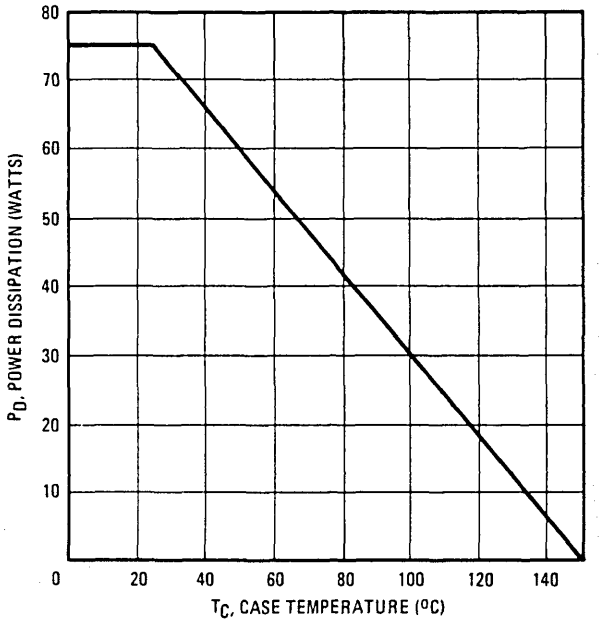


Fig. 11 – Power Vs. Temperature Derating Curve

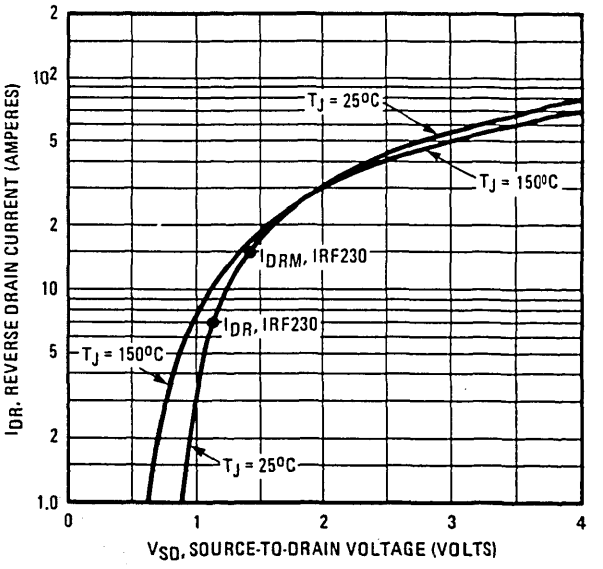


Fig. 12 – Maximum Body-Drain Diode Forward Voltage

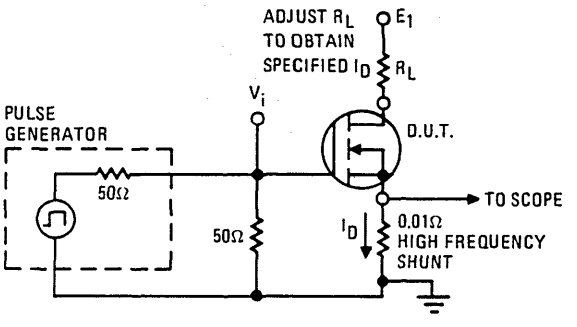
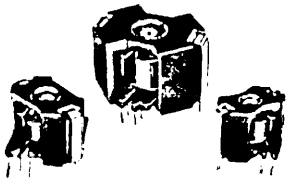


Fig. 13 – Switching Time Test Circuit

APPENDIX 5B.

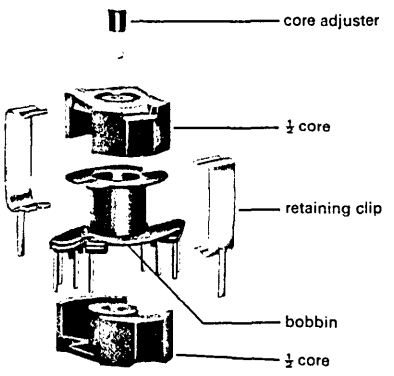
RM SERIES FERRITE CORES.



A range of 5 of the most popular p.c.b. mounting ferrite cores covering three sizes. Of square design which allows maximum board utilisation, this series enables transformers or inductors to be constructed to meet exact customer requirements. The core material is equivalent to the commonly known grades: A13–Q3–N28. Each core is supplied in kit form and consists of the following: one pair of matched half cores, one single section bobbin with integral pins on an 0·1 in grid, one pair of retaining clips with earth spikes and one core adjuster.

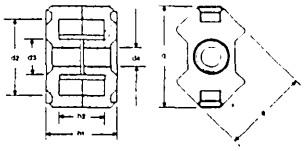
size	A_L value
RM6	160
RM6	250
RM7	250
RM10	250
RM10	400

exploded view of kit

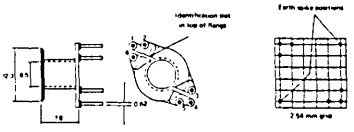


core dimensions

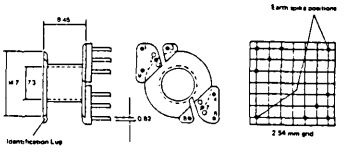
	RM6	RM7	RM10
a (max.)	14·7	17·2	24·7
d2 (min.)	12·4	14·76	21·2
d3 (max.)	6·4	7·24	10·9
d4 (min.)	3·05	3·05	5·4
h1 (max.)	12·5	13·5	18·7
h2 (min.)	8·0	8·93	12·4
q (max.)	17·9	20·3	28·5



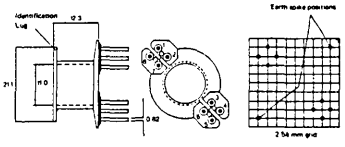
Core dimensions (see table)



RM6 Bobbin



RM7 Bobbin



RM10 Bobbin

To determine the number of turns required for a particular inductance use the following formula:

$$\text{No. turns} = \sqrt{\frac{L}{A_L}}$$

Where L=inductance in nH (10^{-9}H).

properties of core assemblies at 25 °C

		core size				
	symbol	*RM6*	*RM6*	*RM7*	*RM10*	*RM10*
Inductance Factor (nH/Turns²)	A_L	160 ±2%	250 ±2%	250 ±2%	250 ±2%	400 ±2%
Turns Factor (turns for 1mH)	α	79·06 ±1%	63·25 ±1%	63·25 ±1%	63·25 ±1%	50·00 ±1%
Effective Permeability	μ_e	109·5	171·1	146·0	99·67	159·5
Temp. Coeff. of μ_e (+25 to 55 °C) ppm/°C		51 min. 154 max.	80 min. 241 max.	73 min. 219 max.	50 min. 149 max.	80 min. 239 max.
Adjuster Range		+20%	+14%	+15%	+17%	+20%
Max. Residual plus Eddy Current Core Loss Tangent tan δ_{r+f} at 30 kHz		$0{\cdot}34\times10^{-3}$	$0{\cdot}53\times10^{-3}$	$0{\cdot}47\times10^{-3}$	$0{\cdot}32\times10^{-3}$	$0{\cdot}51\times10^{-3}$
at 100 kHz		$0{\cdot}58\times10^{-3}$	$0{\cdot}91\times10^{-3}$	$0{\cdot}82\times10^{-3}$	$0{\cdot}60\times10^{-3}$	$0{\cdot}96\times10^{-3}$
Recommended Frequency Range (kHz)		5·5 to 800	3·5 to 700	3 to 650	2 to 650	1·2 to 500
Energy Storage Capability (mJ)	LI^2sat	0·383	0·245	0·406	1·731	1·082
B_{sat}	mT	250	250	250	250	250

magnetic properties of cores

	symbol	*RM6*	*RM7*	*RM10*
Effective Path Length	l_e	26·9 mm	29·6 mm	41·7 mm
Effective Path Area	A_e	31·3 mm²	40·3 mm²	83·2 mm²
Effective Volume	V_e	840 mm³	1190 mm³	3470 mm³

maximum turns accommodated on bobbin

wire dia. (mm)	*RM6*	*RM7*	*RM10*	*wire dia. (mm)*	*RM6*	*RM7*	*RM10*
0·2	205	306	612	0·56	25	36	87
0·224	160	250	484	0·71	19	33	59
0·25	127	209	402	0·8	13	19	44
0·315	87	131	246	1·0	9	11	25
0·4	47	76	160	1·25	4	9	19
0·5	36	50	98	1·5	3	7	11

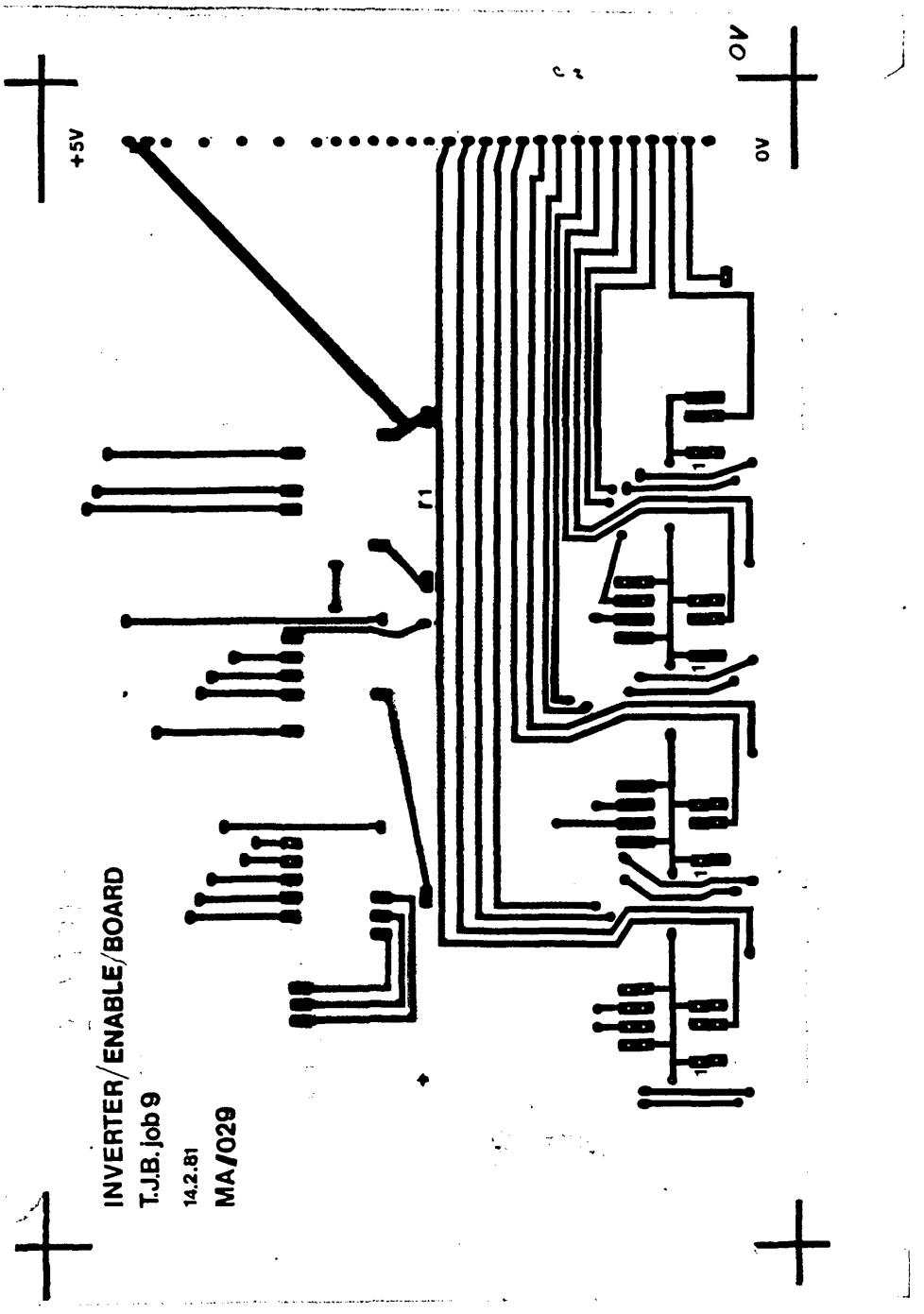
All dimensions are in mm unless otherwise specified

(Note: RM10 Core With A_l = 400 Used In Thesis Circuits)

APPENDIX 5C.

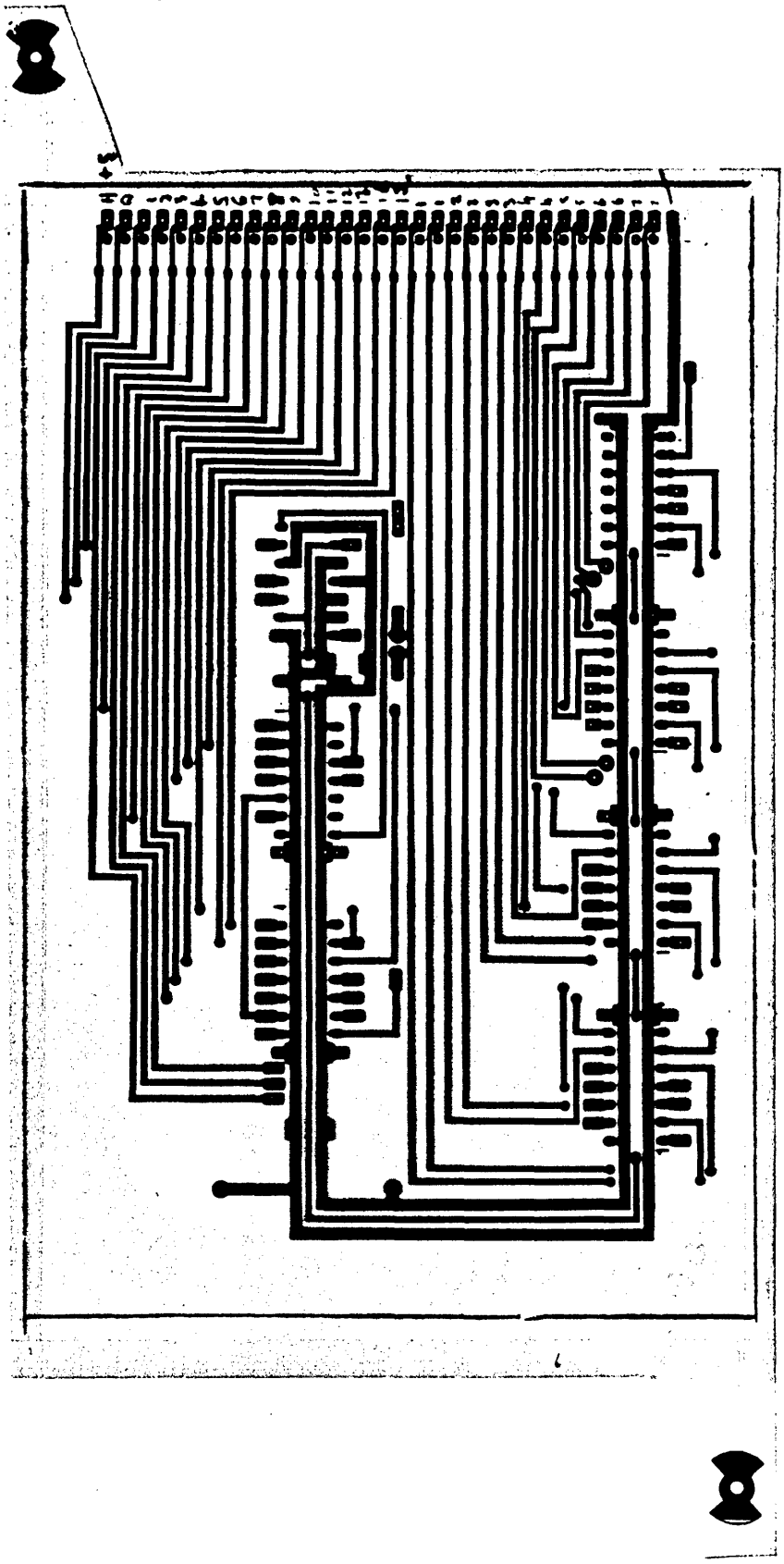
LAYOUT DETAILS OF THE PRINTED CIRCUIT BOARDS USED IN THE SEVEN PHASE MOSFET INVERTER.

1) The Inverter Enable Board: track layout on the component side of the board (viewed from component side).



APPENDIX 5C CONTINUED.

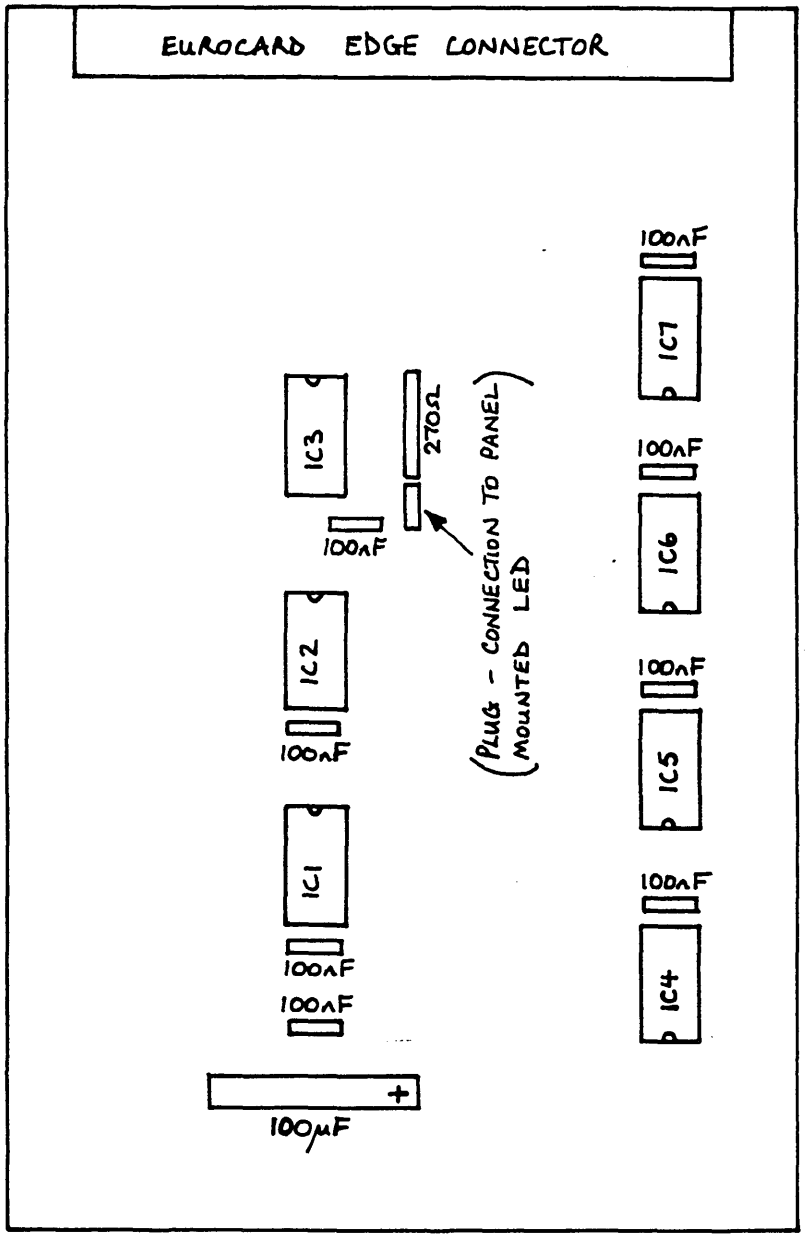
2) The Inverter Enable Board: track layout on the non-component side of the board (viewed from component side).



APPENDIX 5C CONTINUED.

3) The Inverter Enable Board: component positions (viewed from component side of board).

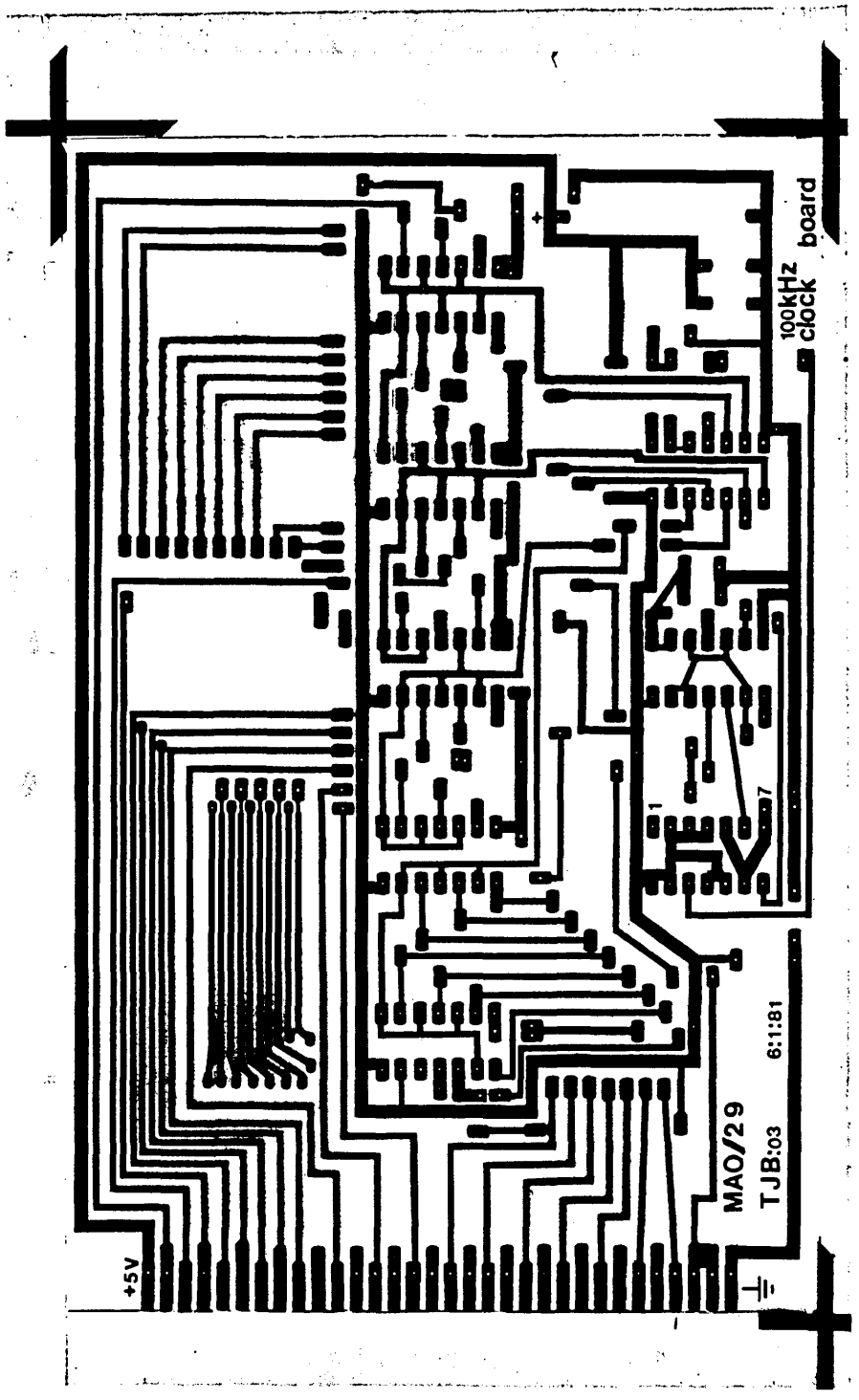
Refer to fig. 5.6(a) for circuit diagram.



NOTE: WIRE LINKS NOT SHOWN.

APPENDIX 5C CONTINUED.

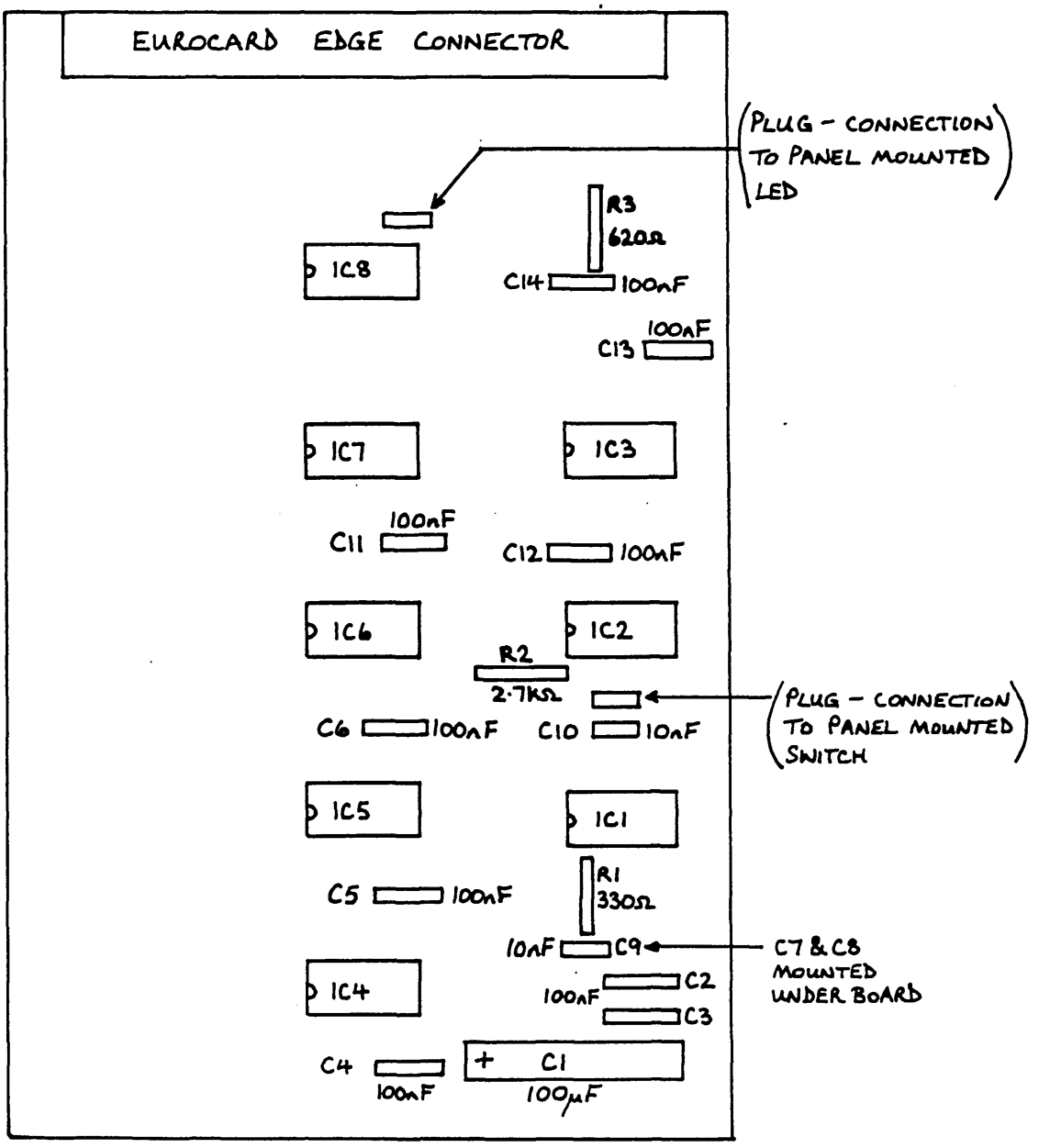
4) The 100 kHz Clock Signals Circuit Board: track layout on the non-component side of the board (viewed from the non-component side).



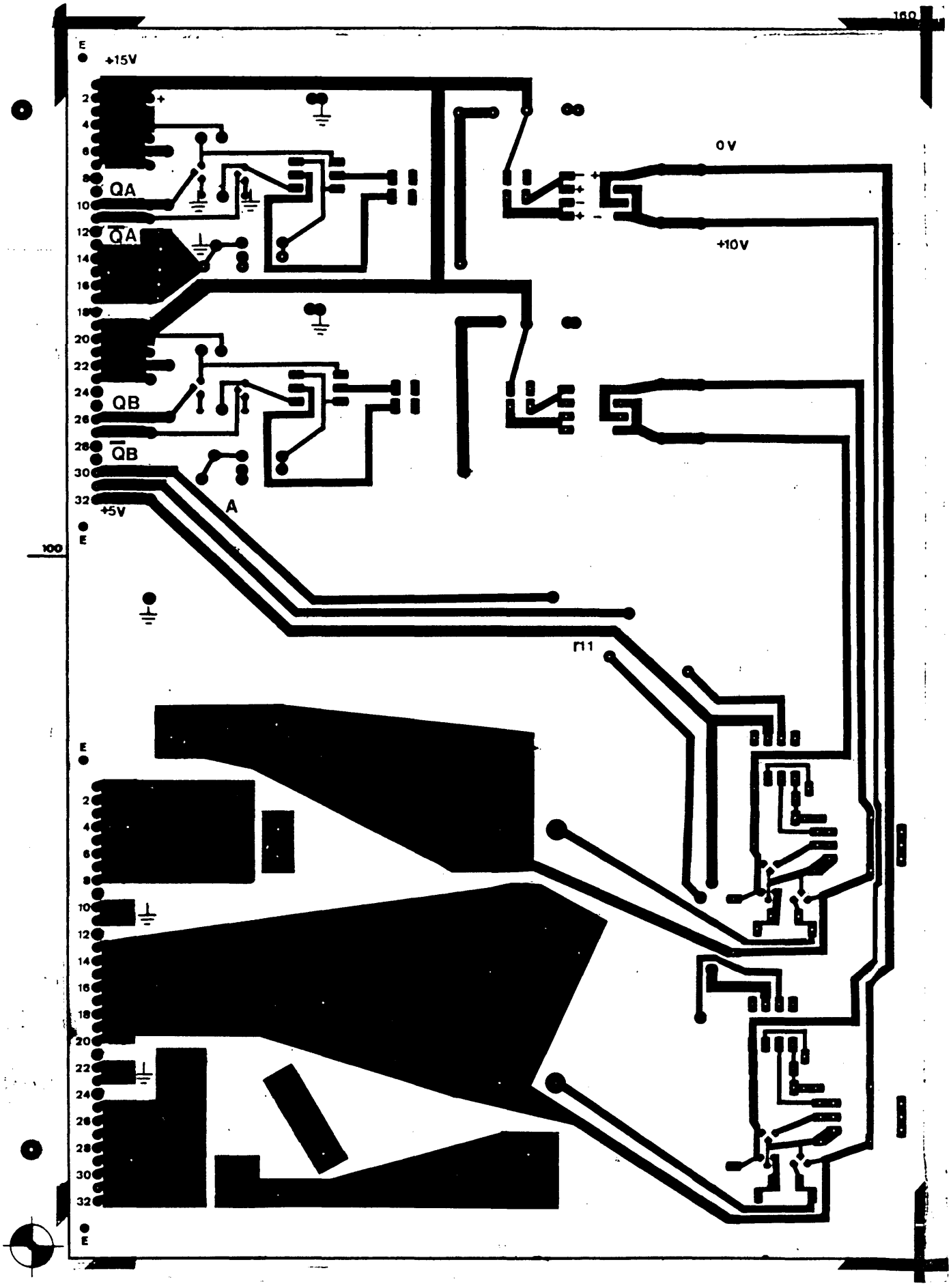
APPENDIX 5C CONTINUED.

5) The 100 kHz Clock Signals Circuit Board: component positions (viewed from component side of board).

Refer to fig. 5.13 for circuit diagram.
For details of wire links see photograph fig. 5.32.

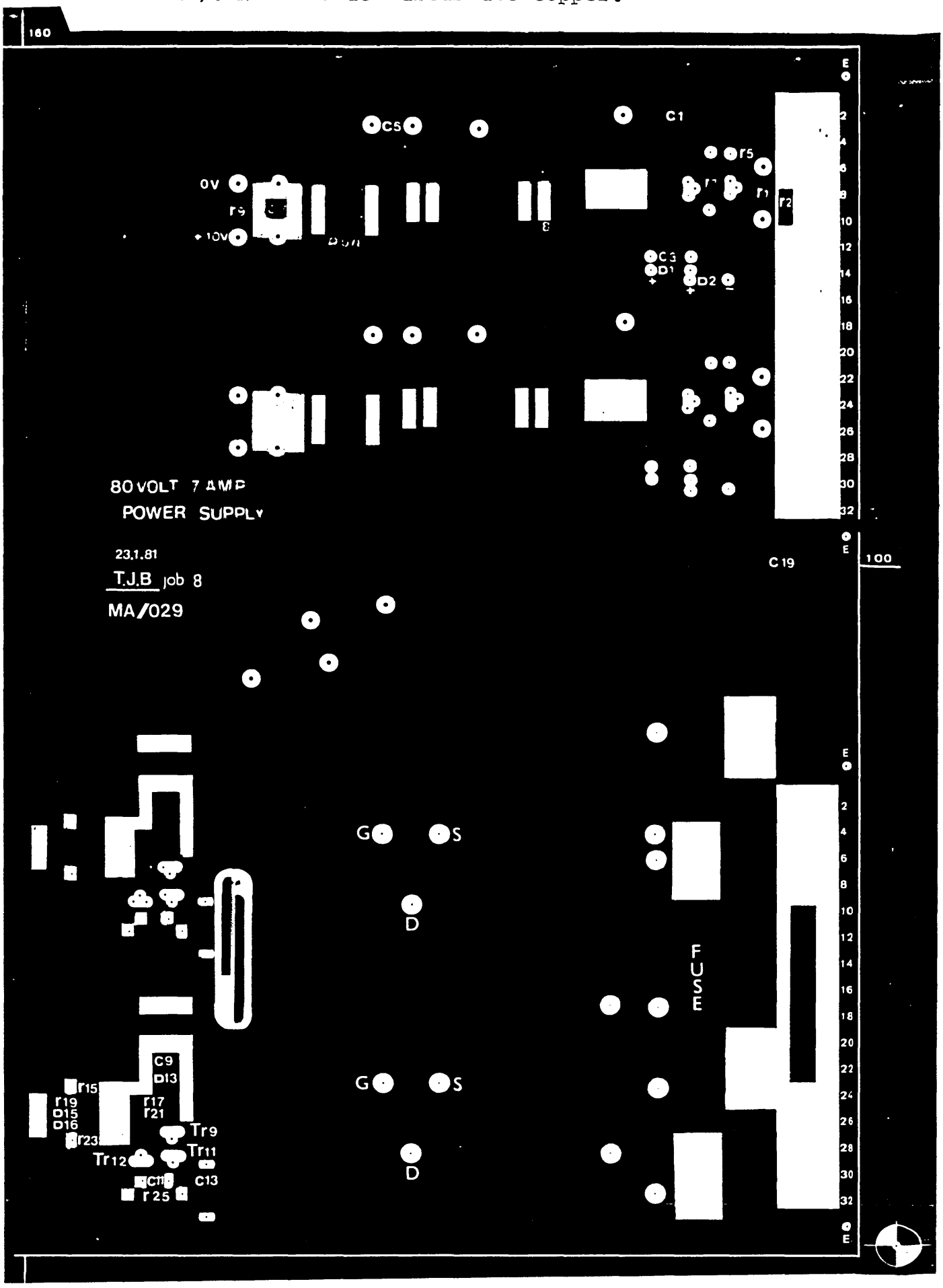


6) The Hexfet Switch Circuit Board: track layout on the non-component side of the board (viewed from the non-component side).



APPENDIX 5C CONTINUED.

7) The Hexfet Switch Circuit Board: track layout on the component side of the board (viewed from the component side). Note- black areas are copper.



APPENDIX 5C CONTINUED.

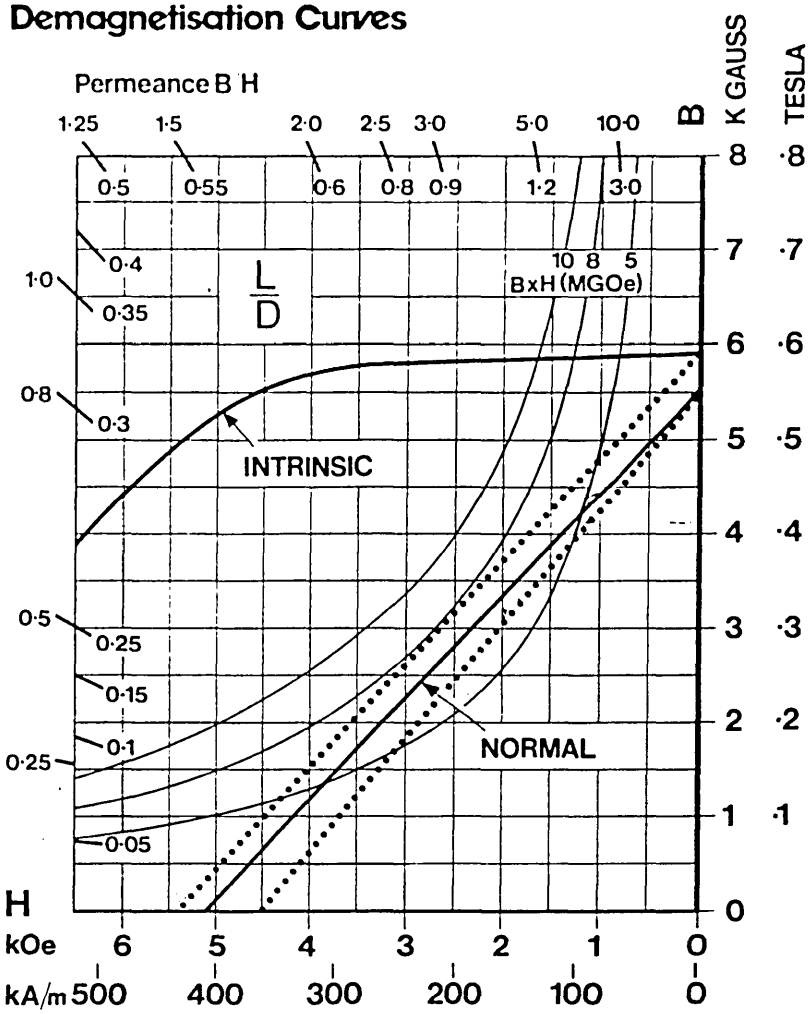
8) The Hexfet Switch Circuit Board: component positions.

Refer to fig. 5.9 and fig. 5.11 for basic circuit diagram (two sets of circuits are mounted on each hexfet switch circuit board).

Fig. 5.22 indicates the arrangement of the circuit blocks on the hexfet switch circuit board and a photograph of the component side of the board is shown in fig. 5.23.

Component positions can be determined by reference to the photograph and by studying the printed circuit board track layout in conjunction with the circuit diagrams (fig. 5.9 and fig. 5.11).

Demagnetisation Curves



Typical Magnetic Properties

Energy Product (BH Max)	M G Oe	7-8
	kJm^{-3}	56-64
Remanence (Br)	Gauss	5500-5900
	Tesla	0.55-0.59
Coercive Force (Hc)	Oersteds	4500-5200
	kAm^{-1}	360-416
Intrinsic Coercive Force (Hc_i)	Oersteds	7500
	kAm^{-1}	600
Recoil Permeability (Hr)		1.05-1.1
Reversible Temperature Coefficient % per °C (-40°C to +75°C)		-0.04
Temperature Range		-40°C to +125°C
Maximum Continuous Operating Temperature		+60°C

Typical Physical Properties (Test Method BS 476)

Density	5.1-5.2g cm^{-3}
Tensile Strength	1.8kg mm^{-2} 2500 psi
Compressive Strength	3.2kg mm^{-2} 4500 psi
Flexural Strength	5.6kg mm^{-2} 8000 psi
Hardness (Brinell $\frac{5mm\ ball}{25kg\ wt}$)	4
Coefficient of Linear Expansion	47×10^{-6} $°C^{-1}$
Electrical Resistivity	0.25-15 ohm cm
Thermal conductivity	2×10^{-3} cal s^{-1} cm^{-1} $°C^{-1}$

An Isotropic version of HR8 is available with the following properties

Energy Product (BH Max)	M G Oe	2.25
	kJm^{-3}	18
Remanence (Br)	Gauss	3000
	Tesla	300
Coercive Force (Hc)	Oersteds	2800
	kAm^{-1}	224

HR8

POLYMER BONDED RARE EARTH MAGNETS

Magnet Design

For optimum performance redesign may be necessary to utilise the full capabilities of HR8. Apart from the extremely costly sintered rare-earth magnets no other material offers the field strength, coercive force or energy product of HR8. A recoil permeability of 1.1 with the knee of the intrinsic curve well into the 3rd quadrant of the normal demagnetisation curve gives a high resistance to demagnetisation and makes HR8 an ideal material for recoil applications.

At BH max significant volume and weight savings can be made over anisotropic ferrites and alnicos.

	HR8	Anisotropic Ferrite	Alnico 5
Density	5.1	4.7	7.35
Bc (Tesla)	0.275	0.19	1.02
Hd (kAm^{-1})	216	137	42
Lm	1	1.57 $(\frac{216}{137})$	5.14 $(\frac{216}{42})$
Am	1	1.45 $(\frac{275}{190})$	0.27 $(\frac{275}{1.02})$
Vm	1	2.28	1.39
Wt	5.1	10.72	10.22

The normal magnetic axis is axial and unless specified the magnets are supplied magnetised. No keeper is necessary. Bolts or clamps should lie parallel to the magnetic axis and should be non-magnetic.

Diametric magnetisation is possible but with some loss in Br.

Mechanical Properties

Mechanical properties are lower in value than ferrites and metal magnets but more than adequate for most applications. Where necessary surface hardness and scuff resistance can be improved by coating or metallising but it is recommended that HR8 is not used in highly abrasive conditions. Impact strength is equivalent to ferrites and vastly superior to sintered rare-earths.

Dimensional Tolerances/Shape

Tolerances are a function of tool design. With low cost tooling tolerances of plus/minus 1% can be held throughout a production run. With sophisticated tooling tolerances can be better than plus/minus 0.5% without costly secondary finishing operations.

Plastic moulding technology enables the manufacture of complex shapes, blind holes and the moulding of inserts.

Machining

HR8 can be easily machined using normal hand and machine tools, making the production of prototypes and short runs easy.

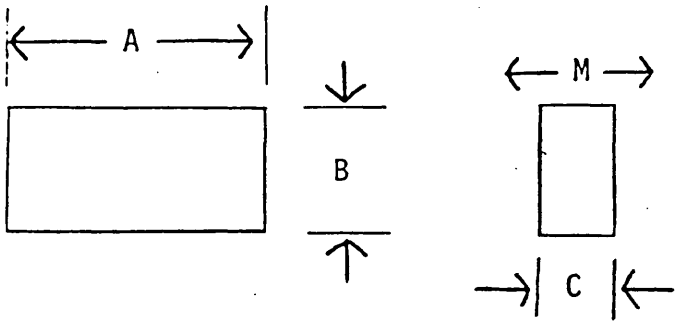
Grinding is not necessary. Normal coolants must be applied to prevent the formation of dust that may induce temporary skin or respiratory irritation.

HR8 can be drilled and tapped.

APPENDIX 6A CONTINUED.

STANDARD MAGNETS

Bars



Tolerances ± 1% or ± 0.3mm whichever is greater

Magnet Type Number	Dimensions in mm			Approx Wt. (g)
	A	B	C	
M027	76	76	10	300
M027/1	76	76	5	150
M014	68	48	4	67.5
M016	68	48	3.2	54
M013	68	48	2	34
M028	50	50	8	104
M012	48	48	12.2	146
M009	48	40	27	269
M051	60	10	15	47
M015	50	12	10	31
M022	30	30	10	47
M026	17	12	4	4.2
M033	10	10	4	2.0
M049	6	6	8	1.5

APPENDIX 6B.

August 1980
Data Sheet GI 8

GRAFIL CARBON FIBRES
GENERAL INFORMATION

INTRODUCTION

Courtaulds GRAFIL high performance carbon fibres are manufactured from a polyacrylonitrile (PAN) precursor using a process developed by Courtaulds from that invented in 1963 at the Royal Aircraft Establishment, Farnborough, England. The acrylic precursor fibre which made such an important contribution to the original breakthrough was specially developed by Courtaulds and is used by other carbon fibre manufacturers as well as in the GRAFIL process.

The property levels of PAN based carbon fibres are determined by the process temperature operated during manufacture and it is possible to produce a family of fibres with a combination of properties suited to different end use requirements. The GRAFIL fibres family consists of three performance types:

GRAFIL A — high strain
GRAFIL XA — high performance
GRAFIL HM — high modulus

MANUFACTURING PROCESS

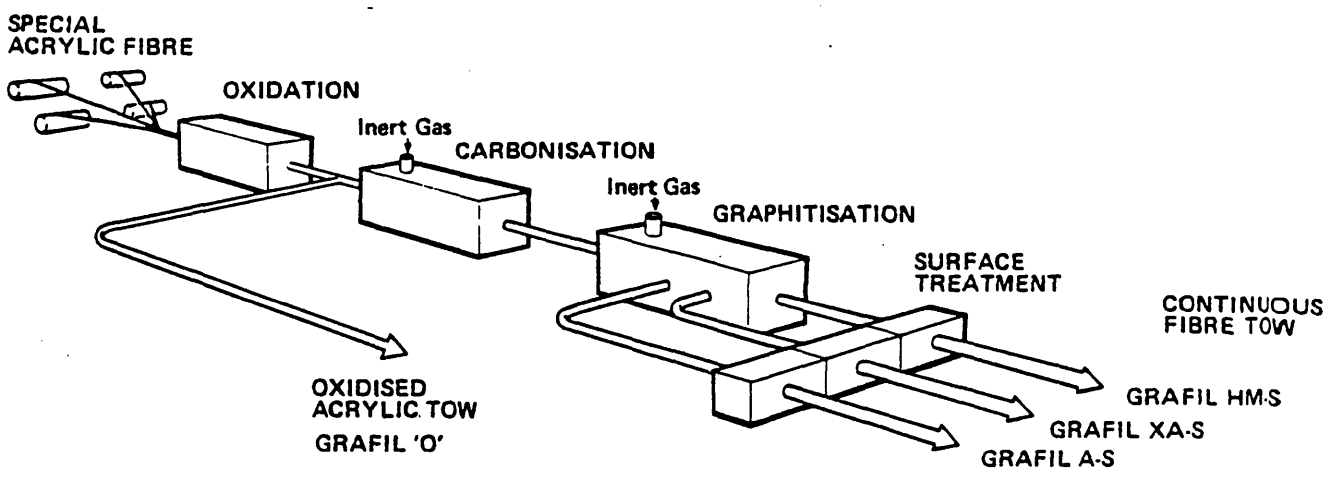


Figure 1. GRAFIL Manufacturing Process

The manufacturing process is made up of three stages; oxidation, carbonisation, and graphitisation taking place at temperatures up to 3 000°C. The fibres produced comprise extremely fine parallel filaments grouped together in tow bundles, which are converted into useful mechanical and structural materials by combining them with a suitable matrix to form a composite. The choice of matrix materials is wide and carbon fibres can be combined successfully with plastics, glass, ceramics, metals and also other forms of carbon to produce high performance fibre composites. Plastics provide a convenient matrix and by far the largest use of carbon fibres is in combination with thermosetting or thermoplastic polymers.

Fibre conversion to composite can be achieved by a variety of techniques, often requiring the use of an intermediate form of material, e.g. prepreg, woven fabric, etc. A comprehensive product range has therefore been developed to meet widely differing processing requirements.

Continued overleaf

APPENDIX 6B CONTINUED.

PRODUCT RANGE

Continuous fibre tow forms the primary fibre production route from which other products can be derived by secondary operations. Tows can be produced with several filament counts ranging in certain discrete steps from 3 000 to 12 000 filaments/tow. Larger tows are also being developed to meet volume conversion requirements. The GRAFIL product range therefore offers a wide choice of materials to suit all types of applications and fabrication processes. The range includes GRAFIL 'O', oxidised acrylic fibre, produced from the same precursor fibre by a process involving a controlled oxidation reaction, basically the first stage of carbon fibre manufacture.

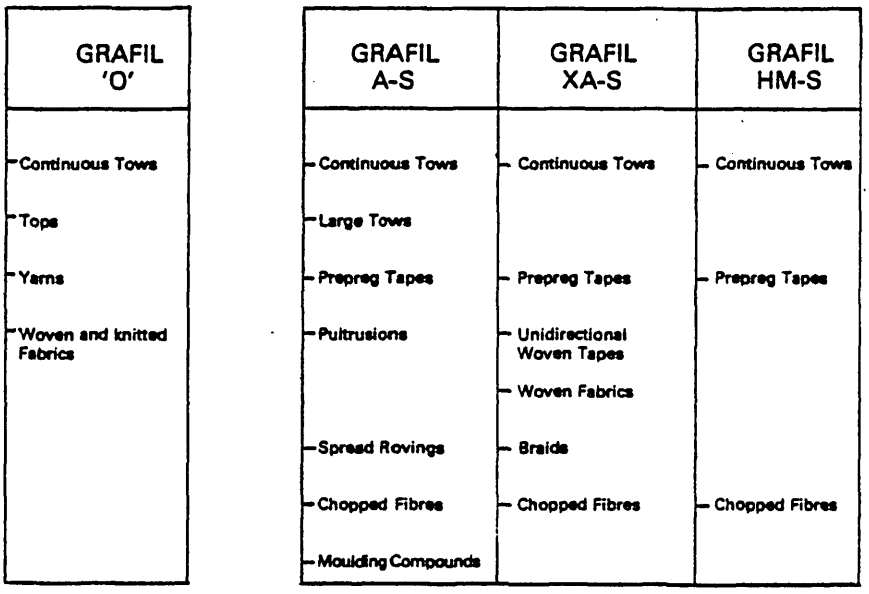


Figure 2. The GRAFIL Product Range.

PERFORMANCE

GRAFIL carbon fibre products can be used with a variety of matrix materials, both thermosetting and thermoplastic and generate excellent performance in composites.

Stiffness/strength

Carbon fibre composites are extremely stiff, strong and lightweight and specific performance is superior to metals and other structural materials. The properties of GRAFIL fibre unidirectional composites are compared with those of several other materials in Figure 3.

Material	*Specific Gravity*	*Ultimate Tensile Strength*	*Young's Modulus*	*Specific Ultimate Tensile Strength*	*Specific Young's Modulus*
Units		*GPa*	*GPa*	*GPa*	*GPa*
GRAFIL A-S	1·5	1·5	110	1·00	74
GRAFIL XA-S	1·5	1·9	125	1·27	83
GRAFIL HM-S	1·6	1·5	190	0·94	119
GRP	2·0	1·0	42	0·50	21
Steel	7·8	1·0	210	0·13	27
Titanium DTD5173	4·5	0·96	110	0·21	25
Aluminium L65	2·8	0·47	75	0·17	26

Figure 3. Comparison of Material Properties.

APPENDIX 6B CONTINUED.

Stress/strain

The stress/strain relationship for GRAFIL fibre composites is illustrated in Figure 4. Unlike most metals, composites are perfectly elastic to failure and exhibit no yield or plasticity region. Failure strains are relatively small compared to steel for instance, however the elastic behaviour means that a far higher proportion of the ultimate strength of the material can be utilised in practice. XA-S fibres with up to 1·5% elongation to failure are therefore highly competitive and are capable of meeting high strain design requirements. Composite materials are anisotropic and mechanical performance is optimum in the direction of fibre reinforcement. Properties transverse to the fibre are significantly lower, largely determined by those of the matrix. Multi-directional performance can be achieved by using fibres in more than one direction, carefully placed to meet the applied stress in application.

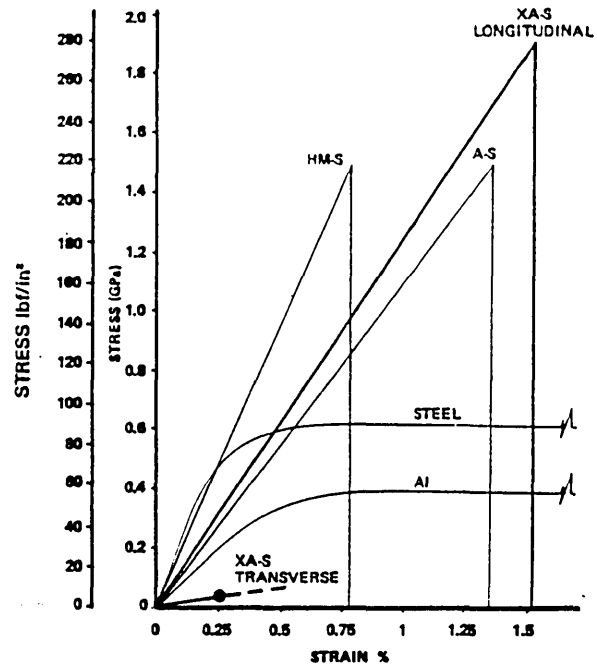


Figure 4. Stress/Strain Relationship.

Fatigue Characteristics

Carbon fibre composites exhibit excellent fatigue performance when subjected to an alternating stress. Figure 5 compares the fatigue characteristics of a variety of engineering materials and illustrates the significant advantage held by carbon fibre/epoxy composite over metals particularly aluminium. The excellent fatigue resistance means that a much higher ratio of working stress to ultimate stress can be utilised practically.

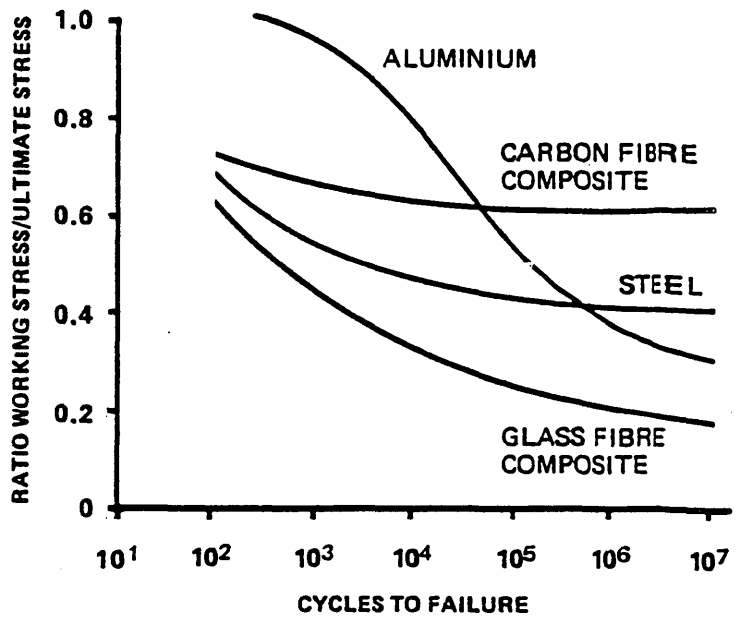


Figure 5. S–N Fatigue Curves.

Vibration Damping

Carbon fibre composites have good built-in damping properties, as illustrated in Figure 6 by the strain amplitude versus time graphs. Compared to steel (as illustrated) and other materials, carbon/epoxy composites damp out vibration more rapidly, a feature that is utilised in fishing rods to achieve longer casting lengths. The vibration in the rod damps out sooner and as a result there is less frictional resistance between line and guides.

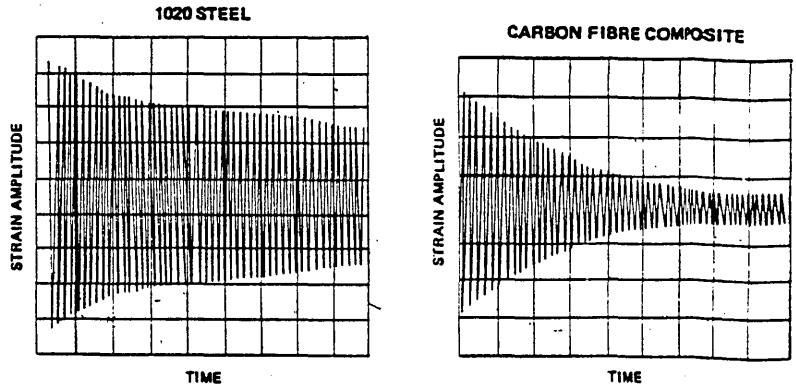


Figure 6. Comparison of Damping

Continued overleaf

APPENDIX 6B CONTINUED.

Creep

Creep in a unidirectional carbon composite loaded in the fibre direction is extremely small. The major part of the load is carried by the fibres which are perfectly elastic unlike many plastics which are viscoelastic. Creep strain in a high strength CF/epoxy composite is only of the order of 0·04% after 3200 hours at a stress of 360 MN/m².

Impact Properties

Elastically carbon fibre composites can absorb impact energy until the maximum strain is reached. Most other materials continue to absorb energy after elastic deformation by plastic deformation. When the maximum strain is exceeded in a carbon fibre composite failure can occur either by interlaminar shear, parallel to the fibres, or cracking transverse to the fibres.

At very low impact velocities the fibres and composite absorb the energy by elastic deformation then return to their original shape.

High impact velocities tend to fracture the carbon fibres transversely, punching a clean hole through the composite. The majority of surrounding fibres are unaffected and a reasonably high level of mechanical performance remains in the composite.

Impact at intermediate velocities causes the maximum amount of damage as the composite fails in interlaminar shear over a wide area surrounding the point of impact. The strength remaining in the composite for intermediate impact velocities is very low.

The higher the strain to failure of the composite the higher the impact resistance, therefore XA-S fibre has a higher impact resistance than A-S or HM-S.

Thermal and Electrical Properties

Along the fibre direction carbon fibres and composites are both thermally and electrically conducting. The level varies with fibre type increasing as the fibre becomes more graphitic.

Thermal Conductivity		*A-S Epoxy*	*HM-S Epoxy*	*Steel*	*Aluminium*
Longitudinal	W/mK	10–17	34	50	205
Transverse	W/mK	1–5	0·80		

Electrical Resistivity		*A-S*	*XA-S*	*HM-S*	*Aluminium*
Fibre tow	μΩm	26	16	10	—
Composite:					
Longitudinal	μΩm	30 – 50			0·03
Transverse	μΩm	7×10^4			

Thermal Expansion

The coefficient of thermal expansion (CTE) for carbon fibre composites varies according to fibre type, fibre orientation and temperature testing range. It should be noted that this variation is non-linear. Typical values for CTE, over the temperature range 100K to 400K (−173°C to 127°C) are given in the table below. It is possible for a designer to tailor the CTE to his requirements by varying the fibre orientation angle within the component. Components with very low or negative CTE's can be produced.

Coefficient of Thermal Expansion × 10^{-6} K^{-1}	*Longitudinal*	*Transverse*
HM-S/Epoxy	−0·25 to −0·60	20 to 65
High Strength fibre/Epoxy	0·30 to −0·30	20 to 65

APPENDIX 6B CONTINUED.

CONTINUOUS FIBRES

GRAFIL high performance PAN-based carbon fibres are available as continuous tows in standard performance types XA-S and HM-S for use as reinforcements in composite materials. The fibres are suitable for a wide range of conversion and fabrication processes, including prepregging, woven fabric manufacture, filament winding, pultrusion, etc., and can also be chopped for incorporation into moulding compounds.

FIBRE PROPERTIES

GRAFIL		*XA-S*	*HM-S*
Tensile Strength Batch Mean Minimum	GPa	2·90	2·10
Tensile Modulus: Limits	GPa	215–245	320–355
Density	g/cm³	1·81	1·86

GRAFIL fibre tensile properties are determined using an impregnated tow test (Test Reference 103.21). The minimum value for **Batch Mean Ultimate Tensile Strength** and the limits for **Tensile Modulus** represent a **basic specification** for each fibre type.

COMPOSITE PROPERTIES

GRAFIL fibres generate excellent performance in composite illustrated by the typical values obtained in epoxy resin system Shell Epikote 828/MNA/K61B.[1]

Property[2]		*XA-S*	*HM-S*
0° Tensile strength	GPa	1·90	1·69
0° Tensile modulus	GPa	128	189
Interlaminar shear strength (5 : 1)	MPa	99[3]	62

(1) Shell Epikote 828/MNA/K61B is a 170°C cure resin used in a 'leaky mould' technique for preparation of Q.C. specimens (Test Method Ref. 201.15).
(2) Values are expressed at 60% by volume fibre.
(3) These values for ILSS were obtained in Ciba Araldite MY720/MNA/K61B.

TOW CHARACTERISTICS

A range of GRAFIL continuous tows is available. Selection of fibre type and number of filaments/tow can be made to suit the end use and conversion technique employed.

Filaments/tow

E/XA-S	12 000, 10 000, 6 000,	filament tows
E/HM-S	10 000 filament tow	

Filament Shape

Individual filaments are round in section and range in diameter from approximately 7–9 micrometres.

Twist

Standard tows are produced completely twist-free and care is taken during manufacture to maintain parallel filament alignment.

Continued overleaf

APPENDIX 6B CONTINUED.

CONTINUOUS FIBRES
GRAFIL XA-S

GRAFIL E/XA-S is a high performance PAN-based carbon fibre available as 12 000 6 000 filament continuous twist-free tows. The fibres are surface treated during manufacture to promote a good fibre/matrix bond in composite, and are sized with an epoxy resin to assist handling. GRAFIL E/XA-S is suitable for use in composite materials where high strength characteristics are required. XA-S composites exhibit a good balance of mechanical properties and achieve excellent levels of failure strain (up to 1·5%).

FIBRE PROPERTIES

Property		*Units*	*Specification*	*Typical Value*	*Test Method*
Tensile Strength*		GPa	Min. 2·90	3·43 (7·6% c.v.)	103.22
Tensile Modulus*		GPa	215–245	237 (2·2% c.v.)	103.22
Elongation		%		1·44	103.22
Density		g/cm³		1·81	104.23
Mass/Unit Length:	12K	mg/m		800	105.14
	6K	mg/m		400	105.14
Size Content:	12K	% mass		0·7	106.11
	6K	% mass		1·5	106.11
Twist Level		tpm		Zero	—
Filament Shape		—		Round	—
Filament Diameter		$\times 10^{-6}$ m		6·8	—

* GRAFIL fibre tensile properties are determined using an impregnated tow test (Test Reference 103.22). The minimum value for Batch Mean Ultimate Tensile Strength and the limits for Tensile Modulus represent a basic fibre specification.

AVAILABLE FORMS

Twist free tows: E/XA-S 12K — 12 000 filaments

E/XA-S 6K — 6 000 filaments

Untwisted tows can be processed successfully by a variety of techniques including prepregging, pultrusion and filament winding.

PRESENTATION

Sizing: E/XA-S 12K tows are available with 0·7% by mass size as standard and 1·5% by mass size to special order.
E/XA-S 6K, sized with 1·5% by mass resin as standard.

Package: E/XA-S 12K fibres with 0·7% by mass size can be supplied either in tubs or on spools.
All E/XA-S tows with 1·5% size are supplied on spools only.

Splices: Splices where necessary are used on the basis of a maximum of two splices/kilogram. In practice splices are infrequent and usually limited to one per package. Typically a production fibre batch contains less than 10% of spliced tows.

COMPOSITE PROPERTIES

GRAFIL E/XA-S fibres achieve excellent performance in composite. Typical values expressed at 60% fibre by volume for two epoxy resin systems are listed in the table:

Resin System	*Tensile Strength GPa*	*Tensile Modulus GPa*	*Interlaminar Shear Strength (5:1) MPa*
Epikote 828/MNA/K61B	1·90	128	99*
GRAFIL 3501-5	2·10	130	126
Test Method	402.14	401.14	406.14

* ILSS Specimen prepared using MY 720/MNA/K61B.

APPENDIX 6B CONTINUED.

August 1980
Health and Safety Information Sheet

HEALTH AND SAFETY INFORMATION

Carbon fibres comprise carbon in multifilament fibrous form. The individual filaments are extremely fine with diameters in the range 7–9 micrometers (0·27–0·32 × 10^{-3}in). All carbon fibres and derived products present a number of potential hazards under use and certain precautions are necessary and processing practices are recommended to ensure the safe handling of these materials.

PRODUCT WARNING LABEL

Small filament particles may arise when carbon fibres are handled or processed and every container or package of GRAFIL carbon fibre products carries a clear warning drawing the user's attention to the hazards that can arise:

Handle with Care

1. May cause minor skin irritation.
2. Fibres are conductive — Protect from electrical equipment.
3. Do not incinerate.

DUST INHALATION

Fine lightweight carbon fibre particles are easily circulated into the atmosphere and can be inhaled by operators handling the material. The use of protective masks is recommended where severe dust generation is likely to occur. Whilst the particles and dust generated may create some temporary discomfort, the particle size of the fibre fragments is considered to be too large to represent a health risk to the respiratory system.

SKIN IRRITATION

The fibre particles can cause transient skin irritation. The irritation usually appears as itching and may also cause a rash. In most cases the irritation is only temporary and disappears after a short time. The more sensitive areas of the skin are affected, e.g. between the fingers, backs of the hand, and also wrists and neck where the problem can be aggravated by the rubbing of tight clothing.

A high standard of cleanliness is recommended and the use of protective workwear can reduce the risk of fibres getting into normal clothing. Barrier skin creams and talcum powder may also be helpful in eliminating any discomfort.

Certain resins (particularly epoxies) and curing agents can produce quite severe skin reactions. Contact with these materials will occur during the handling of fibres sized with epoxy resin (note: the resin size used for GRAFIL fibres does not contain curing agent or other cross linking constituent) or prepreg products and in the preparation of composites.

ELECTRICAL HAZARDS

Carbon fibre fly is electrically conductive and can disturb electrical equipment installations and switchgear causing short-circuiting. Equipment should be protected by sealing to prevent the ingress of fibres or by isolating in clean air conditions. The provision of filtered ventilation maintained at a slightly positive pressure is recommended for particularly sensitive components. Specific advice on electrical protection is available on request from Courtaulds Limited, Carbon Fibres Division.

FIBRE DISPOSAL

Waste carbon fibres, derived products and carbon composite materials should not be incinerated for disposal. Collection and packaging for burial as landfill is recommended. Fibre particles can be released during incineration which are easily circulated in the atmosphere representing a hazard to electrical installations as outlined in the previous section.

Continued overleaf

APPENDIX 6B CONTINUED.

RECOMMENDATIONS FOR SAFE HANDLING

It is important that users are aware of the potential hazards associated with handling carbon fibres and procedures are recommended aimed at minimising the hazard risk and ensuring the safe use of this product.

1. Handling and processing operations should be as mild as possible. Where vigorous handling is unavoidable, e.g. chopping fibre or weaving, the generation of dust and particle fly must be expected. The use of sized fibre is recommended. Note: Fibres are sized by the application of a small amount of resin binder normally applied from solvent solution or emulsion. The size aids handling by providing lubrication and reducing broken filaments.
2. Handling and work areas should be well ventilated with efficient dust extraction particularly at the points where broken filaments are likely to arise.
3. Where dust generation is severe operatives should wear face masks and protective clothing. A check on particle concentration in the atmosphere should be carried out regularly.
4. Adequate protection of electrical installations is essential.
5. High standards of cleanliness and good housekeeping are recommended and the use of specially impregnated mats at plant entrances and exits considerably reduces the risk of transfer of fibres outside the handling area.
6. Fibre storage and handling areas should be protected from draughts.
7. The use of gloves and protective overalls when handling prepreg products and resin materials is necessary to minimise the risk of contact with the skin. The work should be carried out in clean well ventilated conditions. The skin should be thoroughly cleansed with soap and water after finishing work. Resin removing creams are commercially available. The use of solvent for resin removal should be avoided.
8. Waste products containing carbon fibres should not be incinerated. Disposal by burial is recommended.

REFERENCES IN VOLUME TWO

CHAPTER 5

5.1 Issawi, A.M.: 'A Variable Frequency, Variable-Waveform, Three-Phase Converter For Independent Thrust And Levitation Control Of Linear Induction Motors', Ph.D Thesis, Imperial College, London University, 1978, Chapter 3.

5.2 Humphrey, A.J.: 'Constant Horsepower Operation Of Induction Motors', IEEE Transactions on Industry and General Applications, Vol IGA-5, Sept/Oct 1969, pp.552-557.

5.3 Chalmers, B.J.; Pacey, K.; Gibson, J.P.: 'Brushless D.C. Traction Drive', Proc. IEE, Vol 122, No 7, July 1975, pp.733-738.

5.4 Slemon, G.R.; Dewan, S.B.; Wilson, J.W.A.: 'Synchronous Motor Drive With Current-Source Inverter', IEEE Transactions on Industry Applications, Vol IA-10, No 3, May/June 1974, pp.412-416.

5.5 Hurley, N.V.; Bolton, H.R.: 'A Novel Current-Source Inverter For The Supply Of A Brushless DC Machine', in "Power Electronics - Power Semiconductors And Their Applications", IEE Conference Publication 154, Sept 1977, pp.54-57.

5.6 Proceedings of International Conference on Electrical Machines, Athens, 1980, ("PE" sessions).

5.7 Bose, B.K. (Editor): 'Adjustable Speed AC Drive Systems', (IEEE Press, 1980).

5.8 Conference Record, Industry Applications Society, IEEE-IAS-1981 Annual Meeting, Philadelphia, 5-9 Oct 1981.

5.9 Frankl, G.: 'The Performance In Service Of A Special Commutatorless D.C. Machine', in "Small And Special Electrical Machines", IEE Conference Publication 202, 22-24 Sept 1981, pp.178-181.

5.10 Frankl, G.: 'The Multiphase Commutatorless D.C. Machine - A Special Converterfed Machine', Proc. of International Conference on Electrical Machines, Athens, 1980, pp.609-616.

5.11 Frankl, G.; Walzel, G.: 'The Digital Control Of A Commutatorless D.C. Machine For An Electric Vehicle', Proc. of International Conference On Numerical Control Of Electrical Machines (CONUMEL), Lyon, France, 1980, pp.I70-I79.

5.12 McMurray, W.: 'Selection Of Snubbers And Clamps To Optimise The Design Of Transistor Switching Converters', IEEE Transactions on Industry Applications, Vol IA-16, No 4, July/Aug 1980, pp.513-523.

5.13 McMurray, W.: 'Optimum Snubbers For Power Semiconductors', in "Power Semiconductor Applications, Vol 1 1972", (IEEE Press, Editors: Harnden, J.D., & Golden, F.C.), pp.33-41.

5.14 Rice, J.B.: 'Design Of Snubber Circuits For Thyristor Converters', in "Power Semiconductor Applications, Vol 1 1972", (IEEE Press, Editors: Harnden, J.D., & Golden, F.C.), pp.21-24.

5.15 Balthasar, P.P.: 'Fast Commutation Of High Inductive Currents', IEEE-IAS Annual Meeting Conference Record, 1975, pp.55-58.

5.16 Regan, P.: 'V-M.O.S. Improves Efficiency Through Switching Performance', IEE Electronics & Power, September 1979, pp.629-633.

5.17 Baliga, B.J.: 'Switching Lots Of Watts At High Speeds', IEEE Spectrum, December 1981, pp.42-48.

5.18 Severns, R.: 'Using The Power MOSFET As A Switch', Electronic Product Design, May 1981, pp.49-53.

5.19 Holmes, L.: 'Micro-Compatible Power MOSFETs', IEE Electronics & Power, April 1980, p.290.

5.20 Hewlett Packard Ltd.: 'Gate Drive Circuits For Power MOSFETs', Electronic Product Design, Dec 1980, pp.27-28.

5.21 Severns, R.: 'The Power MOSFET As A Rectifier', Powerconversion International, Mar/Apr, 1980, pp.49-50.

5.22 Clemente, S.; Pelly, B.; Ruttonsha, R.: 'Apply A Few Design Rules To Avoid Destroying Power FETs', EDN, May 13, 1981, pp.126-130.

5.23 Taylor, B.: 'High Efficiency Power Switching With Cascoded HEXFETs And Bipolars', Electronic Engineering, November 1981, pp.61-69.

5.24 Collins, H.W.; Pelly, B.: 'HEXFET, A New Power Technology, Cuts On-Resistance, Boosts Ratings', Electronic Design, Vol 27, No 12, June 7, 1979, pp.36-40.

5.25 Stein, K.U.: 'Present Status And Trends In MOS Power Devices', 4th European Conference On Electrotechnics (EUROCON '80), Stuttgart, 1980, pp.355-359.

5.26 Pelly, B.: 'Applying International Rectifier's Power MOSFETs', International Rectifier Power MOSFET Application Notes, Number AN-930/A.

5.27 'VMOS Power FETs Design Catalog', Siliconix Inc., January 1979.

5.28 'POWER MOS Field-Effect Transistors', (Theory, Cross Reference and Data Sheets), Motorola Semiconductor Products Inc., 1981.

5.29 'SIPMOS Power Transistors', Siemens Data Book, November 1980.

5.30 'Power MOSFETs And Their Applications', IEE Discussion Meeting at Savoy Place, London, 10th February, 1982.

5.31 Snelling, E.C.: 'Soft Ferrites', (Iliffe Books Ltd., 1969), Chapter 11.

5.32 Snelling, E.C.: 'Soft Ferrites', Iliffe Books Ltd., 1969), pp.341-343.

5.33 Eisler, P.: 'The Technology Of Printed Circuits', (Heywood & Co. Ltd., 1959), pp.169-171.

5.34 'Transistorgleichstromstellar Für Vorschubantriebe', (Bild 22b), Siemens Technical Publication, Number GWE647/015.

5.35 Houldsworth, J.A.: 'Purpose Designed Ferrite Toroids For Isolated Current Measurement In Power Electronic Equipment', Mullard Technical Publication M81-0026.

CHAPTER 6

6.1 Jahns, T.M.: 'Improved Reliability In Solid-State A.C. Drives By Means Of Multiple Independent Phase-Drive Units', IEEE Transactions on Industry Applications, Vol IA-16, No 3, May/June 1980, pp.321-331.

6.2 Enslin, N.C.; Malengret, M. (Cape Town University, South Africa): 'Flux And Current Distributions In The 9 Phase Inverter Driven Induction Motor', Private communication to Imperial College, Dept of Electrical Engineering, July 1978.

6.3 Stephens, R.C.: 'Strength Of Materials', (Edward Arnold, 1970), Chapter 14.

6.4 Stephens, R.C.: 'Strength Of Materials', (Edward Arnold, 1970), Chapter 1, pp.4-6.

6.5 Roters, H.C.: 'Electromagnetic Devices', (John Wiley & Sons Ltd., 1st Ed., 1941), pp.130-139.

6.6 Say, M.G.: 'Alternating Current Machines', (Pitman Publishing, 1978), pp.105-106.

6.7 Carter, G.W.: 'The Electromagnetic Field In Its Engineering Aspects', (Longman, 2nd Ed., 1967), pp.199-200.

6.8 Russell, R.L.; Norsworthy, K.H.: 'Eddy Currents And Wall Losses In Screened Rotor Induction Motors', Proc. IEE, Vol 105A, 1958, pp.163-175.

6.9 Boules, N.: 'Impact Of Slot Harmonics On Losses Of High-Speed Permanent Magnet Machines With A Magnet Retaining Ring', Electric Machines and Electro-mechanics, Vol 6, 1981, pp.527-539.

6.10 Carter, F.W.: 'Air Gap Induction', Electrical World And Engineer, Vol 38, 1901, pp.884-888.

6.11 Say, M.G.: 'Alternating Current Machines', (Pitman Publishing, 1978), pp.37-39.

CHAPTER 7

7.1 Rieke, B.: 'The Microprocessor Control Of A Four Phase Star-Connected Multi-Pole Reluctance Motor', Proc. of International Conference on Electrical Machines, Athens, 1980, pp.394-401.

7.2 'American National Standard Code For Information Interchange', (ASCII), American Standards Institute Publication X3.4-1968.

7.3 Lewis, J.: 'Remote Keyboard', Practical Electronics, September 1981, pp.24-27.

7.4 'Count Display I.C. ZN1040E', RS Components Ltd., Data Sheet 2848, March 1983.

CHAPTER 8

8.1 Finney, L.W.; Davis, J.H.: 'Precision Measurement Of Angular Speed And Acceleration', Conference Record, Industry Applications Society Annual Meeting, IEEE-IAS-1975, pp.369-372.

8.2 Kühnlein, H.: 'Measurement Of The Elapsed Time During One Rotation Of The Rotor Of A Brushless DC Motor', United States Patent, No 3903462, Sept 2nd, 1975.

8.3 Woodward, F.I.: 'Digital Frequency Comparator For Motor Control', Electronic Engineering, August 1976, p.13.

8.4 Török, V.; Valis, J.: 'High Accuracy And Fast Response Digital Speed Measurement For Control Of Industrial Motor Drives', IFAC Symposium on Control in Power Electronics and Electrical Drives, 2nd Dusseldorf, 1977, (Editor: Leonhard, W.), pp.721-732.

CHAPTER 10

10.1 Forsythe, J.B.: 'Paralleling Of Power MOSFETs For Higher Power Output', Conference Record, Industry Applications Society Annual Meeting, IEEE-IAS-1981, Oct 5-9, 1981, pp.777-796.

10.2 Clemente, S.; Pelly, B.R.; Isidori, A.: 'Understanding Power MOSFET Switching Performance', Conference Record, Industry Applications Society Annual Meeting, IEEE-IAS-1981, Oct 5-9, 1981, pp.763-776.

10.3 'Hexfet Databook', International Rectifier Corp., Databook HDB-1, 1981.