DISTILLATION COLUMN DYNAMICS

AND CONTROL

A thesis presented for the degree of

Doctor of Philosophy in Chemical Engineering

in the

University of Canterbury

Christchurch, New Zealand

by

GRANT WILSON

1979

ABSTRACT

A pilot plant scale, atmospheric pressure, sieve plate distillation column was constructed and fully instrumented. Novel speed controllable pumps were used to control liquid flows. A microcomputer was constructed to provide local and hierarchical control of the column. The microcomputer included an operator console, a 16 channel data acquisition unit, a 4 channel control output unit, and a hardware arithmetic processor. A software development system was assembled by linking the microcomputer to a minicomputer. Software written for the development system included a cross-assembler, a transfer program, and a microcomputer control program.

A binary steady state distillation column model was developed, solved on a digital computer, and verified against experimental data using a binary mixture of methanol and water.

Two control schemes were investigated using only the microcomputer resources. A multi-loop system using digital PI controllers was found to give excellent control within the accuracy of the instrumentation.

An adaptive feedforward controller was proposed and verified using a steady state model, and experiments. The results were good, but because of the relatively simple dynamics of the experimental column, the feedforward controller was no better than the feedback controllers.

A microcomputer control system has been shown to be an effective replacement for conventional analog control on a distillation column. The computing power of the microcomputer has enabled a sophisticated control scheme to be implemented at low cost.

ACKNOWLEDGEMENTS

Many thanks to Dr. R.M. Allen for his help and advice during this work.

Thanks also to all the technicians of the Chemical Engineering Department for their efforts in constructing equipment.

This work was financed by a University Grants Committee grant (C76/25). Glassware was kindly donated by Ivon Watkins-Dow Ltd.

The author acknowledges financial support from a B.P. Postgraduate Scholarship, a University Grants Committee Postgraduate Scholarship, and a University of Canterbury Teaching Fellowship.

Thanks to Mrs. D.E. Ball for typing, and my parents for proof-reading this thesis from my illegible writing.

Finally, to my parents, who provided endless support and help, my most grateful thanks.

DISTILLATION COLUMN DYNAMICS AND CONTROL

CONTENTS

CHAPTER ONE

INTRODUCTION

Distillation processes have been extensively studied because they occur frequently in the chemical processing industries, are large consumers of energy, and are often critical in determining product purities. The objectives of previous studies have been to improve the understanding of the dynamics of distillation columns, and consequently to improve performance by maximising production and minimising costs. This work represents a continuation of this theme with the use of new technology, and an alternative control system.

A range of approaches have been adopted in the literature including complex computer simulations, simple black box models, and experimental tests using analog and digital hardware. Each approach may have its merits, but there is no single approach which can be applied to all distillation processes, in fact the control problems of a particular distillation column may be unique. The overriding philosophy behind all investigations of the performance of various control schemes will ultimately be profit - is the extra expense of a more complex control system justified?

The major problems of distillation control are:

(i) the interactions between control loops;

(ii) the non-linearities of the process;

(iii) the time variation of process parameters;

(iv) the slow dynamics of the process;

(v) the use of slow and inaccurate sensors and analysers;

(vi) the large load changes which may occur.

Solutions to these problems have been proposed and examined using feedback control, combined feedforward/feedback control, multivariable control, and optimal control strategies. In some cases, the techniques have been successful, but in others the process non-linearities, and the measurement difficulties have led to poor control.

1.1 OBJECTIVES

The objectives of this work were fourfold:

(i) to design and construct a pilot scale distillation scheme;

(ii) to instrument the column;

(iii) to construct a computer based control system;

(iv) to investigate computer control of the column.

The criteria for the column and instrumentation were:

(i) redesign and reconstruct a 225 mm diameter atmospheric pressure sieve tray column of conventional form using components from an existing column;

(ii) provide temperature sensors on all trays, level sensing in the reboiler and reflux accumulator, overall pressure drop sensing, and product composition analysers;

(iii) provide flow control systems on the distillate, reflux, bottoms and steam flows;

(iv) provide flexible operation of the column.

The choice of a digital control system was made from among the following:

(i) microprocessors acting as analog controller replacements under minicomputer control;

(ii) a microcomputer/minicomputer system with substantial microcomputer independence;

(iii) a microcomputer standalone system with an operating system and peripherals;

(iv) a minicomputer system.

Significant technological developments in microprocessors, and their support hardware, coupled with a drop in price for semiconductor integrated circuits prompted a shift in emphasis from using a minicomputer to a microcomputer. A minicomputer would have been under utilised on a single distillation column, but if used as the centre of a development system for a microcomputer, it could be released for other uses when not required. Option (ii) was therefore adopted, and a microcomputer system developed with interfaces to the process, the operator, and a nearby minicomputer. A development system was created in the minicomputer to enable rapid and easy programming of the microcomputer.

Several control strategies were investigated, and compared on the pilot scale column. These were based on

(i) feedback control;

(ii) feedforward/feedback control.

The feedforward controller was based on the shortcut column design method of Gilliland (1940). A steady state computer model was developed to predict the column performance, and to test the feedforward controller.

The overlying objective of this work was to create a system that approximated what might be found in the industrial world. In all instances, attempts were made to use the simplest solutions to all problems, provided that the results were satisfactory. This was particularly true in the design and tuning of the controllers used. The microcomputer software was designed to be compact yet flexible with respect to control configuration and to incorporation in a wider control scheme.

CHAPTER TWO

REVIEW

Process control has been extensively studied and the investigation of the behaviour and control of distillation columns has been no exception. Every conceivable control strategy has been applied to controlling simulated and experimental distillation columns, but production units have in general relied upon the less exotic control strategies such as feedforward/feedback control.

The introduction of minicomputers into process control produced a centralisation of control functions over the previously used distributed analog systems. The big failing of the minicomputer system was the need to provide analog backup in case of a computer failure (Bruce and Fanning (1964), Guisti et al (1962), Rosenbrock et al (1965)). The current trend is back to distributed control schemes with microcomputers providing dedicated control on one piece of plant and perhaps connecting with a hierachical control scheme (Tao et al (1977)). The major control equipment vendors are now marketing distributed control systems, e.g. Honeywell TDC2000 system.

The rapid development of the microprocessor and its peripherals, coupled with a drop in the price of semiconductors has made sophisticated microcomputer-based control schemes possible. It is now feasible to provide intelligent control systems in places where a minicomputer could not be justified (Skrokov(1976)). These dedicated microcomputers can be further adapted to process control by providing special instruction sets (Cummings and Miller (1977)) and special programming languages (Gillespie (1977), Claggett (1977)).

The use of microprocessors can be extended into the instrumentation area. These devices can be used to operate instruments and to provide sophisticated signal conditioning (Garelick (1977)). An increase in the use of discrete process analysers such as gas chromatographs and mass spectrometers can be expected as microcomputers are programmed to operate these instruments and interpret the results (Bailey (1978)).

Improvements are being made in sensing variables with solid state transducers. National Semiconductor's (1974) pressure and temperature sensing integrated circuits are good examples. The price and performance of these systems make it possible to build more sophisticated control systems at reasonable prices.

While the advances in control computers have been spectacular, the improvements in DDC algorithms have been much lower key. Various controller algorithms using different design criteria have been proposed (Mosler et al (1967), Kurz and Isermann (1977)), but the industry standard, the PID controller, has been shown to be the best general purpose controller (Bristol (1977), Unbehauen et al (1976)). The more complex algorithms can be beneficial on some processes where long lags and deadtimes exist, or where interactions occur. Multivariable techniques are best applied to systems where the process models are simple but accurate over the operating range, and can lead to conventional PID controllers (Shih (1970)).

Distillation Column Dynamics

Distillation column dynamics have been extensively investigated, and several excellent reviews of the literature have been made by Archer and Rothfus (1961), Williams (1963) and Rademaker et al (1975). A study of dynamics requires a study of the steady state problem in order that any simulation may have a valid starting point.

Several approaches have been used to solve for the steady state conditions in a distillation column. The methods of McCabe and Thiele (1925) and Sorel (Gilliland and Robinson (1950)) involved plate by plate calculations to find the number of trays required for given operating conditions. Attempts by Martin (1963), Singh (1966), Sarkarny et al (1970) and others to find analytical solutions have involved simplifying assumptions such as a linear vapour/liquid equilibrium relationship, and hence have limited applicability. The shortcut methods of Brown et al (1939), Gilliland (1940), Mason (1959) and Erbar et al (1961) have merit in that they are invaluable for preliminary design work, but are inaccurate for more exacting requirements (Van Winkle and Todd (1971)). The problem of determining the column compositions and flows for a given geometry has been tackled by solving the simultaneous heat and mass balances which can be written around each stage (Amundson and Pontinen (1958), Holland (1963), Wang and Henke (1966)). These methods involve an iterative solution for the flows and compositions, and can be applied to multicomponent and binary systems. Other factors such as the tray efficiency can also be included.

The determination of the dynamic response of a binary distillation column involves the solution of a set of ordinary differential equations (for a plate column). The majority of the literature on column dynamics deals with the assumptions inherent in deriving these equations and the method of solution. Early researchers used analog computers to solve the equations (Lamb et al (1961)), but more recent workers have used digital computers. Svrcek (1967) showed that the differential equations for a binary eight plate column could be solved in 5% of realtime (assuming that the liquid flow dynamics were fast enough, so the liquid flow differential equations could be replaced by steady state balances). A number of other methods of solution are summarised by Rademaker et al (1975). The consequences of the assumptions made in some of the models (e.g. negligible liquid and vapour holdup) can be serious in large columns where liquid holdups are significant. However, in smaller pilot scale columns, such effects can generally be ignored.

Rademaker et al (1975) have summarised the more important experimental results. Most of the work on experimental columns, using step tests, shows good agreement between the suggested theory and the experimental measurements. The number of trays used in such columns is generally much less than in production columns, and hence the response will be different.

Simple transfer function models (lags and deadtime) have been used to describe column responses (Luyben and Gerster (1964), Jafri et al (1965), Wood and Berry (1973), Krishnamoorthy and Edgar (1977)). The fitted models were used to design controllers. Approximate models for large distillation columns have been suggested to avoid the need to solve large systems of differential equations. These models can be determined from steady state data using the concept of inventory time which can be defined as the ratio of the total change in storage for the whole column to the total change in flow out of the column (i.e. a pseudo first order time constant for the complete holdup of the whole column in terms of a single component). Improvements on the initial work of Moczek et al (1965) have been made by Wahl and Harriot (1970), and Weigand et al (1972), but the results have only been compared with simulations, and generally only apply to large columns with many trays.

Distillation Column Control

There is a vast quantity of material on distillation column control in the literature. Rademaker et al (1975) have summarised the major contributions in this field up to the late 1960's. Control schemes have generally been examined theoretically on experimental columns or practically on production columns and there is a large gap between these two groups. Experimental columns are, in general, pilot scale atmospheric pressure binary columns, and hence avoid a lot of the problems of large production columns, e.g. liquid flow lags, and the need for pressure controls. Much work needs to be done in bridging this gap and extending the scope of industrial control schemes using economic criteria to justify the changes.

The major practical contributions to distillation control have been summarised by Shinskey (1967) using single loop controllers with variations. These schemes include the direct and indirect material balances, pressure control loops, and feedforward schemes based on a constant product to feed ratio. Systems based on these approaches have been the basis of most industrial distillation control.

Many researchers have looked at the application of multivariable techniques to distillation columns. An excellent review of this approach is given by Edgar and Schwanke (1977). Some of the simple multivariable techniques such as decoupling have been studied but generally only on experimental columns. The effect of decoupling single loop controllers operating on product compositions has been shown to produce improvements, but this is to be expected because of the lags and delays caused by the large liquid holdups in the reflux accumulator and reboiler, and the consequently slow responses of the composition loops. Indirect composition control by controlling internal tray liquid temperatures is superior even without decoupling because of the faster loop dynamics. Other schemes using optimal multivariable controllers require process models which are generally inaccurate because of linearisation, model order reduction, incomplete measurements, and changing operating conditions (Schwanke et al (1976)). The results of these inaccuracies are sub-optimal and in some cases unstable control. Some work on the use of adaptive controllers (Sastry et al (1977)) has shown promising results for a single loop, but the overall performance is no better than can be achieved by feedforward/feedback techniques.

A gap between control theory and practice still exists in distillation control. Exotic control schemes have been proposed on the basis of small experimental columns and simulations without production scale testing. The introduction of sophisticated multivariable techniques into distillation control will continue to be hampered by poor models. The explosion in the field of microcomputers indicates that the commonly used feedforward/feedback schemes will be improved and implemented more widely now that expensive minicomputers are not required. The future appears to be in a dedicated control computer on each distillation column, with the possible connection to an overall plant control system.

CHAPTER THREE

DISTILLATION COLUMN HARDWARE

CONTENTS

CHAPTER THREE

DISTILLATION COLUMN HARDWARE

3.1 INTRODUCTION

A pilot plant scale, atmospheric pressure distillation column had been used in the Department of Chemical Engineering for teaching in undergraduate laboratories. For the reasons noted below, this column was unsatisfactory both in its performance and in its possible application to this project:

(1) reboiler capacity was excessive;

(2) condenser capacity was inadequate;

(3) reflux/distillate split was by a swinging bucket reflux divider giving pulsed flows;

(4) poor tray efficiency (< 30% overall);

(5) insufficient and ineffective control equipment;

(6) no on-tray sampling and temperature measurement;

(7) no composition measurement;

(8) no centralised control station.

The column was completely dismantled and rebuilt using such parts of the original as were required. The reboiler capacity was reduced, and the condenser capacity increased. The trays were redesigned and rebuilt to allow feed/drawoff and temperature measurement. The reflux divider was replaced by a reflux accumulator. A control system was devised constructed and installed using speed controllable motors driving positive displacement vane pumps for liquid flows, and a flow control loop for the reboiler steam supply. The distillation column was instrumented with temperature, pressure, level and composition sensors, and interfaced to a microcomputer-based control system.

3.2 REBOILER AND CONDENSER MODIFICATIONS

The reboiler originally consisted of a shell and tube heat exchanger with ninety-six 19 mm OD type 316 stainless steel tubes 1220 mm long. The overall heat transfer coefficient was measured to be 1600 $Wm^{-2}K^{-1}$ based on the internal heat transfer area of 5.8 m^2. Assuming saturated steam in the reboiler chest at 70 kPa, and water on the tube side at 100°C, the maximum energy transfer rate was 140 kW. Calculation of the maximum likely column performance showed a reboiler capacity of 100 kW to be sufficient. Consequently, twenty-seven of the ninety-six tubes were blanked off using two stainless steel plates clamped across the tube bundle by tie rods through the now unused tubes. This reduced the internal heat transfer area to 4.2 m^2 and gave an energy transfer rate of 100 kW based on the conditions given above. The reboiler was fitted with two safety valves. That on the process fluid side was set at 70 kPa, and piped outside building in a 50 mm diameter galvanised line. The safety valve on the steam jacket of the reboiler was set at 175 kPa.

The condenser system on the original column comprised one QVF HEU9 and one QVF HEM9 glass heat exchanger with a total heat transfer area of 3 m^2. This system was altered by replacing the HEM9 with two QVF HE9 glass heat exchangers giving a total heat transfer area of 7 m^2. For a methanol/water mixture condensing at 70°C, and using cooling water at 20°C, the maximum rate of energy transfer based on the QVF heat transfer coefficient of 280 $Wm^{-2}K^{-1}$ (QVF (1970)) was 98 kW . Each exchanger was fitted with a valve in its cooling water line to allow manipulation of the condenser capacity to suit the column operating requirements.

The heat transfer equipment in summary was:

(a) Reboiler - shell and tube heat exchanger
- 69 tubes 19 mm OD x 1220 mm 316 S.S.
- all parts in contact with the process fluid are 316 S.S.

- internal heat transfer area 4.2 m^2
- overall heat transfer coefficient 1600 $Wm^{-2}K^{-1}$

(b) Condensers - QVF glass coil heat exchangers
- two HE9, one HEU9 units
- heat transfer area 7 m^2
- overall heat transfer coefficient 280 $Wm^{-2}K^{-1}$.

3.3 TRAY DESIGN

The column trays were redesigned to improve the overall tray efficiency, and to provide sample/feed ports and temperature measurement. The sample/feed ports were located directly beneath the downcomers bringing liquid on to the tray, in a small chamber below the actual active tray. The chamber was opened to the tray by two 30 mm diameter and two 17 mm diameter holes. The liquid holdup in the chamber was measured to be 160 cm^3. The temperature probes were screwed into the top ring of the tray section, and sealed with PTFE tape. The sensing heads were located as close as possible to the exit weir on the tray to give a close estimate of the temperature of the liquid leaving the tray. The trays were assembled between 225 mm diameter x 300 mm standard glass QVF sections giving a tray spacing of 390 mm.

The tray layout is shown in figure 3-1, and plate 3-1.

Summary of Tray Details:

(i) active area - 223 x 3 mm D holes on 12 mm triangular spacing
- total active area = 1.6 x $10^{-3} m^2$

(ii) free area - 225 mm D
- area = 4.0 x $10^{-2} m^2$

(iii) downcomers - two, 35 mm D
- area = 2.0 x $10^{-3} m^2$

(iv) weirs - 19 mm high
- 122 mm apart centrally located.

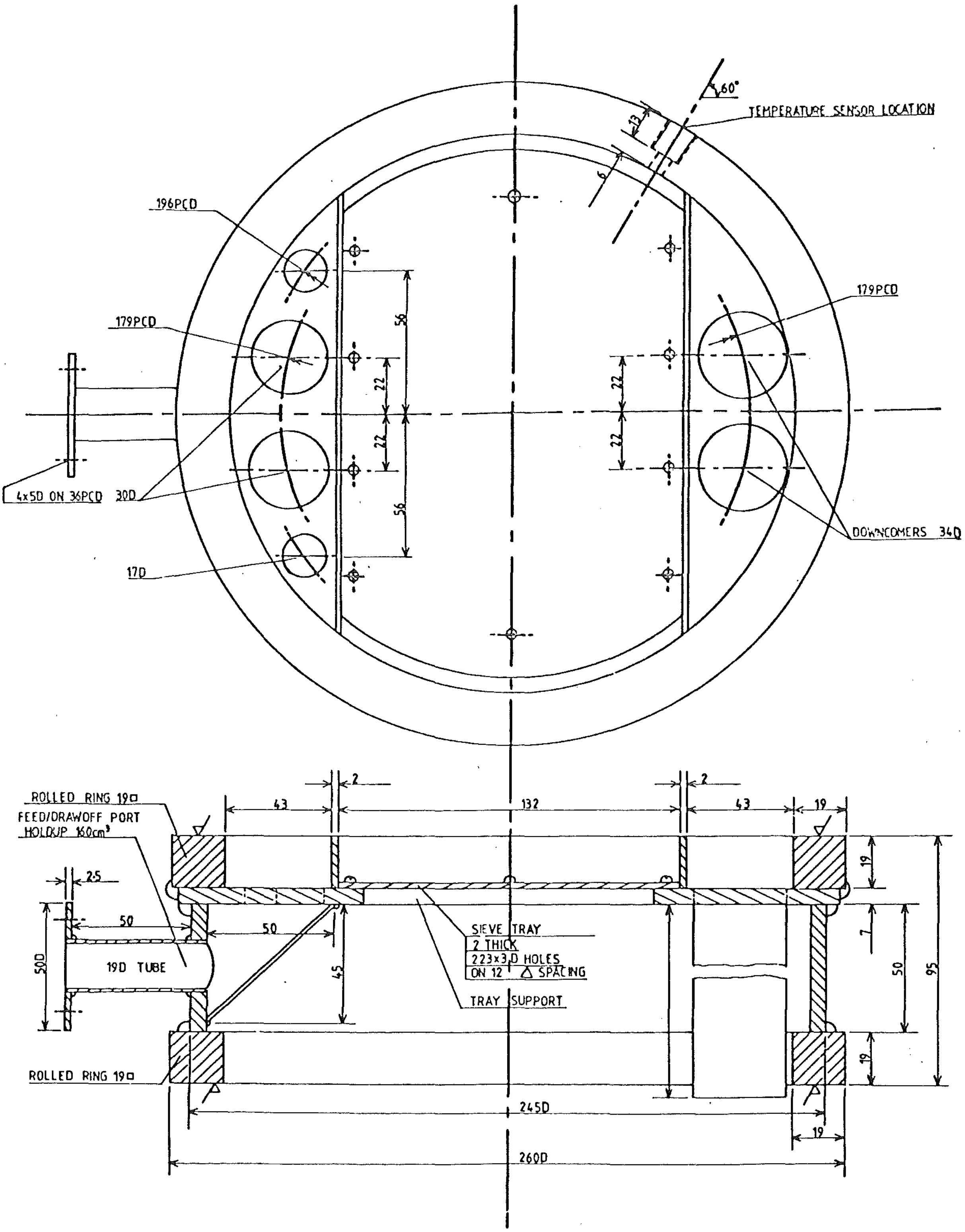


FIGURE 3-1 SIEVE TRAY DETAILS

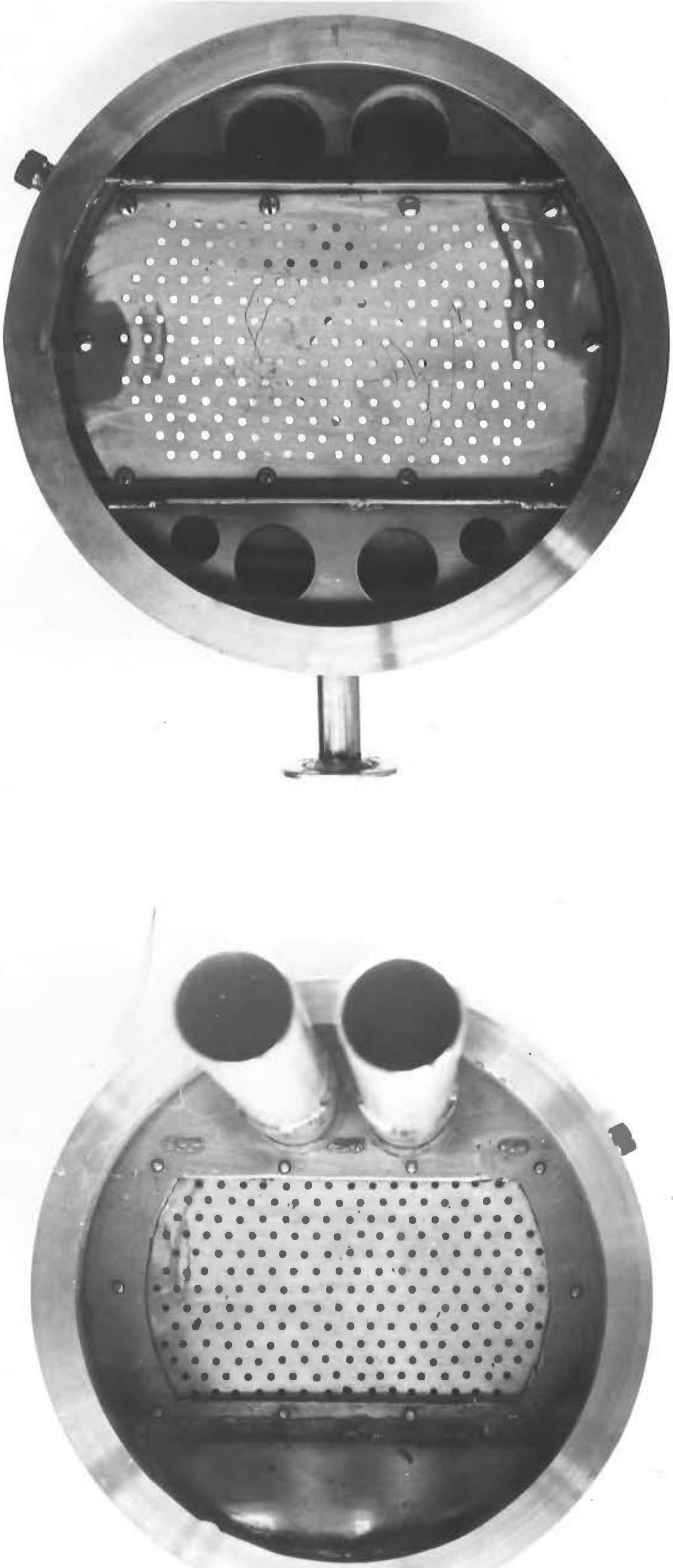


PLATE 3-1 SIEVE TRAY

3.4 REFLUX ACCUMULATOR

The original column incorporated a swinging bucket reflux divider, which inevitably led to pulsed reflux flows, and in some situations to column trays running dry. To correct this situation and to bring the column in to line with standard industrial practice, a reflux accumulator was incorporated. The accumulator was constructed from a standard QVF glass section 600 mm x 225 mm diameter with a type 316 stainless steel cap to seal the base of the accumulator, to support the accumulator and the condensers, and to provide the pipework connections for liquid flows and a temperature sensor. The accumulator provided a maximum liquid holdup of 25 litres.

Incorporated in the reflux drum was the atmospheric vent for the column to allow air to escape during startup, and to ensure that the column remained close to atmospheric pressure. The 50 mm diameter PVC vent line was piped outside the building for safety.

The reflux accumulator and vent are shown in figure 3-2.

3.5 CONTROL EQUIPMENT

The previous column operated under the following controls:

(i) steam pressure control in the reboiler jacket;

(ii) reboiler level control by bottoms flow (pneumatic PI controller and d/p cell);

(iii) reflux ratio control by a QVF swinging bucket.

These controls were all subject to a number of problems including a steam pressure setpoint very close to atmospheric pressure, airlocking in the bottoms flow loop and pulsed flow for the reflux divider. To overcome these problems, the original controls were replaced with individual controls on all liquid flows, and flow control on the reboiler steam supply.

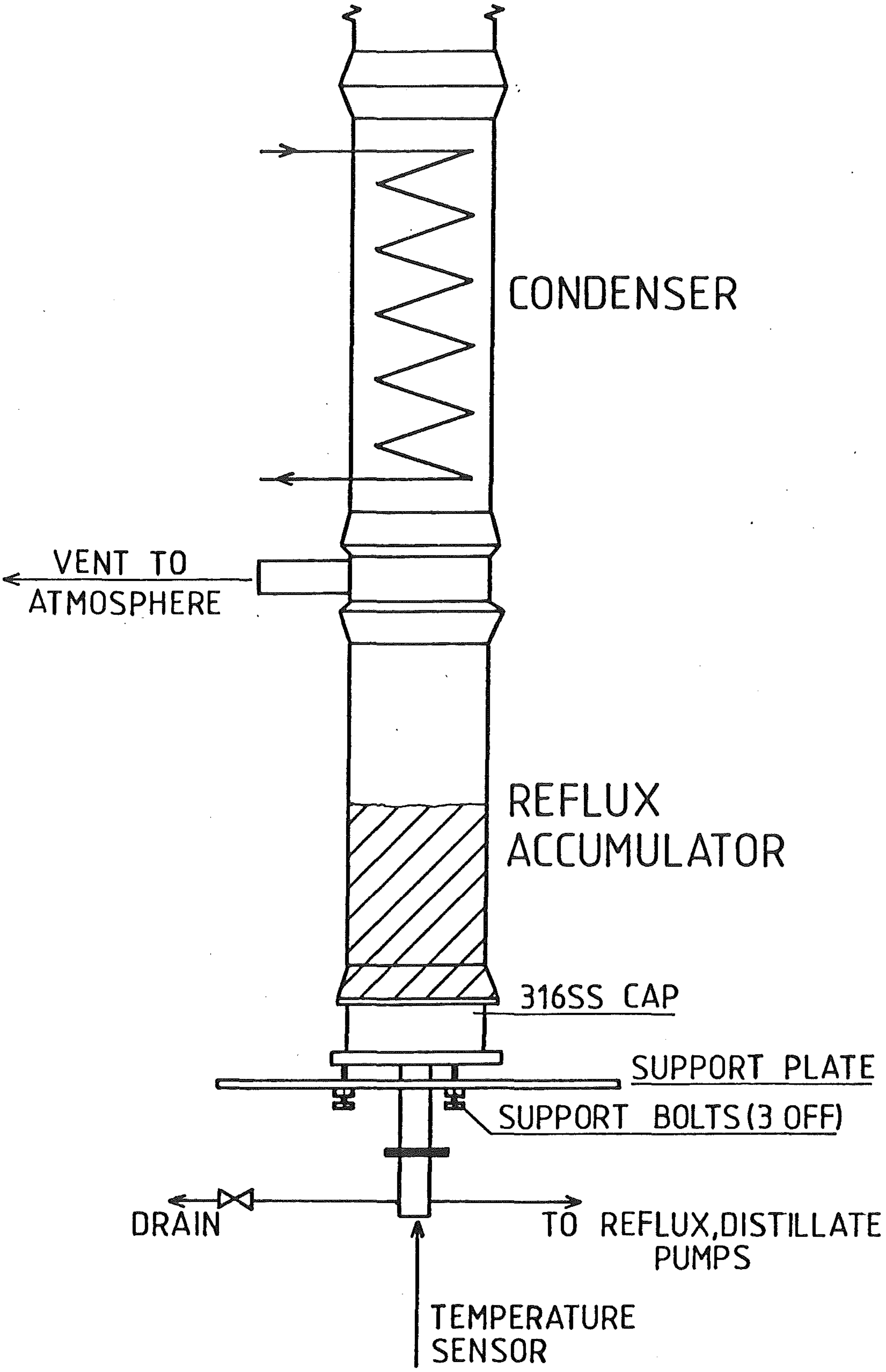


FIGURE 3-2 REFLUX ACCUMULATOR

3.5.1 Liquid Flow Control

The standard practice for controlling liquid flow is to use a flow sensor, a controller and a valve in a feedback loop. For the column used in this project, the maximum flowrate of any liquid stream was estimated to be 4.5 ℓmin^{-1}. To get good turndown on the flow loops a flow sensor was required to handle the range 0.05 - 4.5 ℓmin^{-1} . Devices to achieve this accuracy are available but are very expensive, e.g. turbine meters. To have implemented three conventional flow control loops would have been expensive and beyond the project budget.

Speed controllable positive displacement pumps were considered as replacements for the conventional flow loops. The major problem was to find a controllable prime mover to drive the pumps. A positive displacement vane pump, driven by a variable speed motor was selected as suitable based on considerations of cost and ease of construction.

3.5.2 Variable Speed Motor

A survey of currently available equipment showed a lack of a suitable motor in the 200 W, 0.5 Nm range. Small stepping motors of very limited torque are available, and larger induction motor controls are available, but in between there is a gap. The problem was solved by designing and constructing a stepping motor around a standard Bosch automotive alternator, driven by a variable frequency three phase inverter.

The alternator motor incorporated a number of novel features:

(i) An optically isolated DC signal provided control of the inverter frequency and hence motor speed. A 0-10V dc signal gave a speed range of 0-2000 rpm. The control voltage was conditioned by the input circuit to ensure that the rate of control voltage increase did not exceed 0.2 Vs^{-1}. If this rate was exceeded, the protection circuit would remove the control signal from the inverter, and then slowly ramp the inverter control up to the desired level. This action was necessary to ensure that the motor did not lose synchronisation due to a rapid change in inverter frequency.

(ii) Sequencing of the inverters was performed using CMOS digital logic.

(iii) A current limiting circuit was incorporated in each phase to prevent overloading of the drive transistors.

(iv) Field excitation of the alternator rotor was achieved using a 12V dc supply feeding 2A through the alternator slip rings.

A circuit diagram of the inverter appears in Appendix I.

The combination of the inverter and alternator produced a very useful motor with good low speed torque characteristics, as required by a variable speed positive displacement pump. The torque/speed and power/speed characteristics are shown in figure 3-3.

The use of slip rings to supply the rotor excitation current provided an ignition source, and an explosion risk. This was eliminated by installing each motor in a vented case, continuously purged with air from a fan located outside the building. The continuous flow of fresh air prevented a buildup of explosive vapours within the motor case and helped in cooling the motor.

To ensure that these motor systems were intrinsically safe, an interlocking safety system was installed. A pressure switch in the air purging line ensured that the motors could not operate without the purging system. Further, for a period of 50 seconds after power on, the motors were disabled to ensure that the motor cases were fully purged before applying the excitation supply. Details of this interlocking system are shown in Appendix I.

3.5.3 Positive Displacement Pump

A suitable pump at a reasonable cost could not be found on the local market. A vane pump was designed, this being the simplest type of positive displacement pump to construct.

The pump was designed with a capacity of 4.5 ℓmin^{-1} at 400 rpm to make best use of the low speed torque characteristics of the motor described in the previous section. It was initially constructed using a

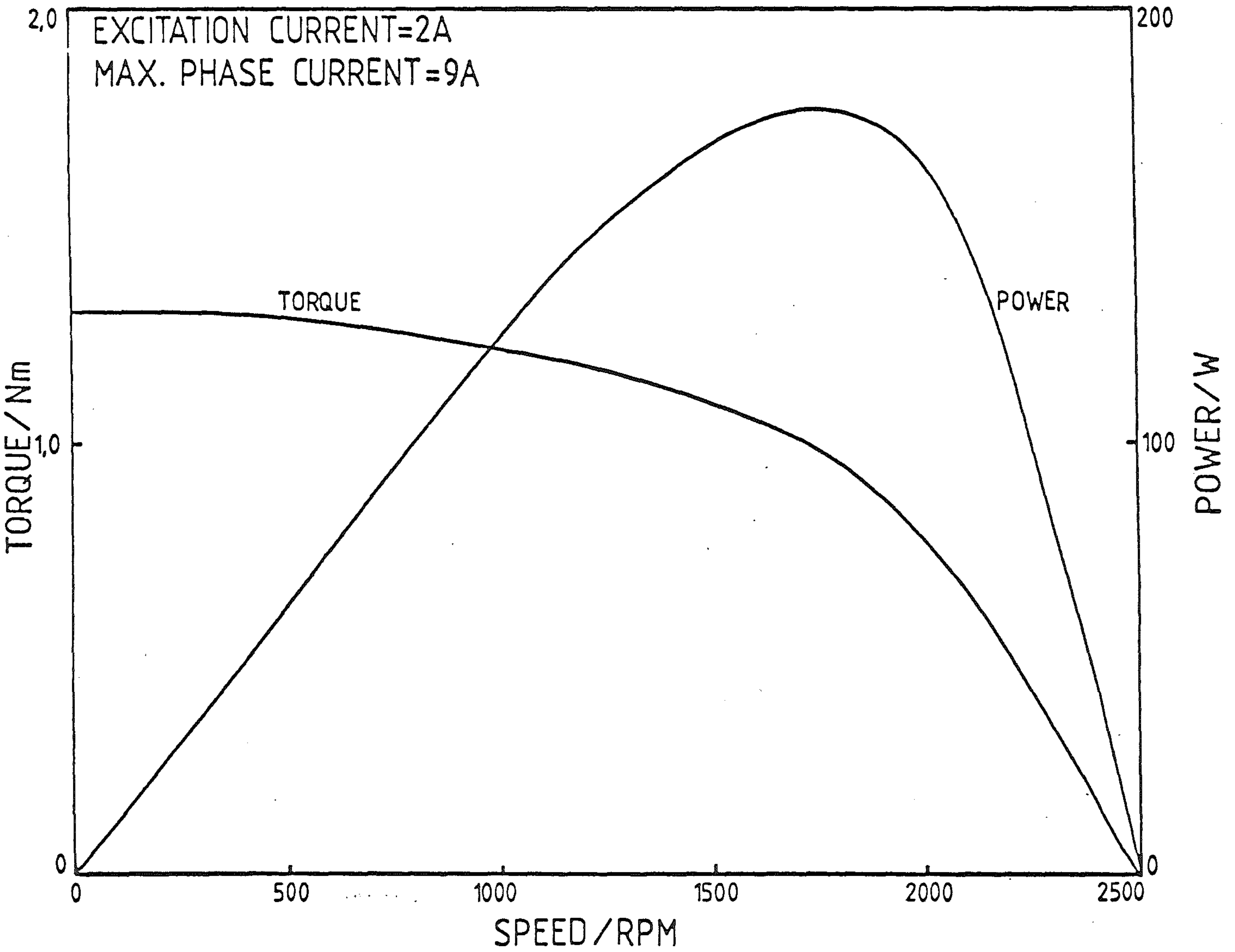


FIGURE 3-3 ALTERNATOR-MOTOR CHARACTERISTICS

type 316 stainless steel body and rotor, with PTFE vanes and cams. Vane movement was by the action of the cams alone; no springs were used. Initial tests showed the pumps to be satisfactory in most circumstances but some problems did arise.

When operated at elevated temperatures (100°C) the PTFE vanes and cams began to creep and expand causing increased frictional forces and eventually seizing. The solution was to provide larger clearances in the pump when assembling it at room temperature. This solved the seizing problem, but the elevated temperatures also accelerated the wearing of the vanes and the cams. The cams acted as bearings and wore rapidly due to some side loading of the shaft from the pumping action. These problems were overcome by fitting two ball races in a housing outside the pump body to absorb the side loading, and by fitting graphite vanes in place of PTFE.

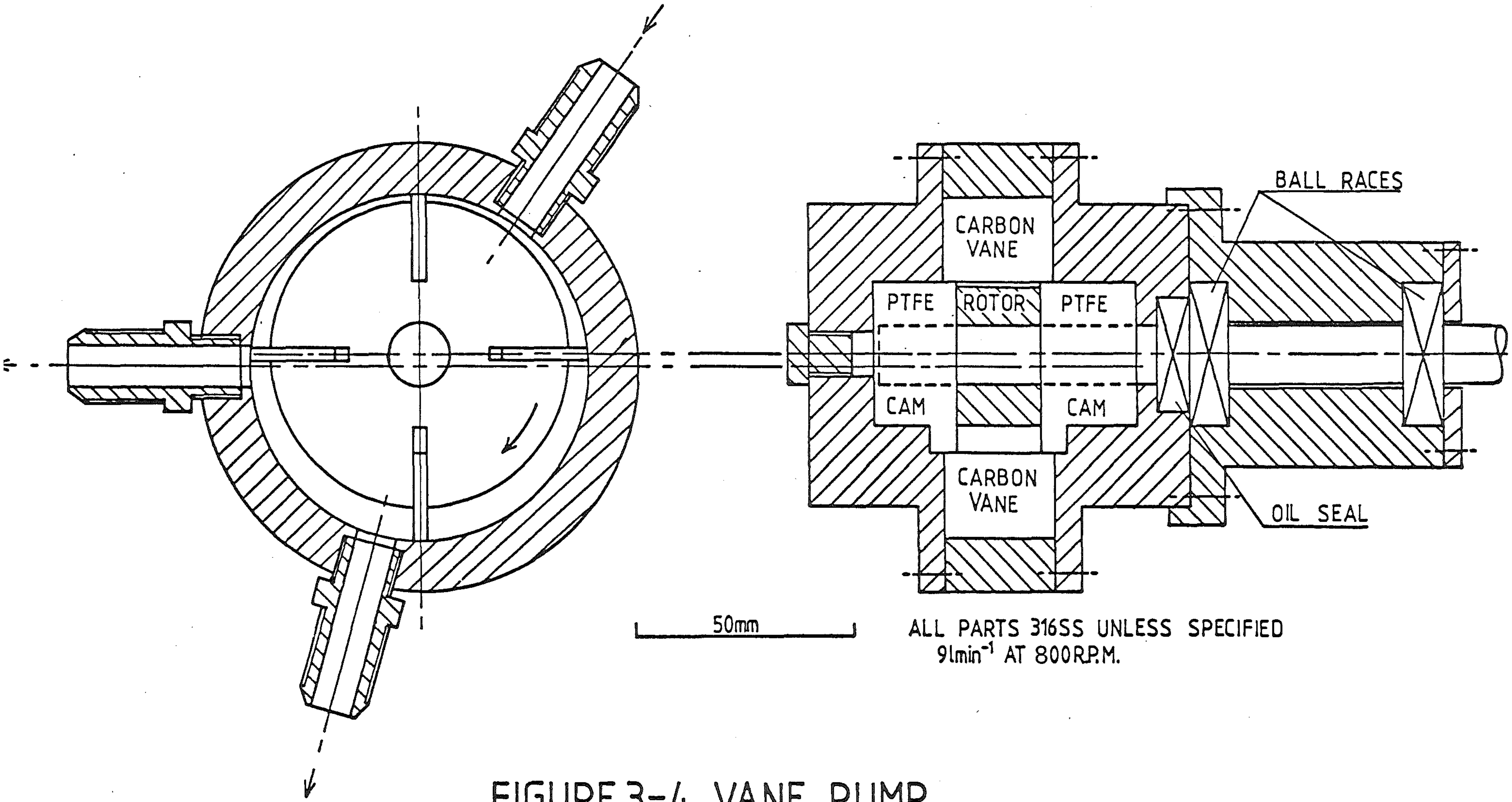
This final design has proved satisfactory provided the pumps are properly set up for the conditions of operation, i.e.

- the bottoms pump was set up for liquids at 100°C,
- the distillate and reflux pumps were set up for liquids at 20°C.

The final design is shown in figure 3-4. The pump was connected to the motor described by a flexible coupling. The completed unit is shown in plate 3-2.

3.5.4 Pump Installation

The three pumps (distillate, reflux, bottoms) were installed to operate under flooded suction and a positive pressure head. To achieve a positive head the product streams (distillate, bottoms) were raised above their respective inlet levels and discharged into a siphon breaker to flow under gravity back into the feed tank. The siphon breaker was constructed from a 2 m length of QVF 25 mm diameter glassware, vented outside the building. The vent ensured that under no conditions could



ALL PARTS 316SS UNLESS SPECIFIED
9lmin^{-1} AT 800R.P.M.

FIGURE 3-4 VANE PUMP

liquid siphon through the pumps. In addition an 18X rotameter with a koranite float was installed in the delivery line of each pump for easy monitoring of liquid flow rates.

3.5.5 Steam Flow Control

Control of the steam to the reboiler was changed from a pressure to a flow loop so that the loop setpoint could be easily related to the actual steam flow. The layout of the loop is shown in figure 3-5.

The orifice plate was non-standard in both its diameter (19 mm) and in the placement of the pressure tappings, 9 mm either side of the orifice plate, requiring calibration.

Equipment:	Controller	Foxboro model 58P4-WV (P+I)
	Control Valve	12 mm D Saunders Valve
	d/p Cell	Foxboro 0-625 mm W.G.
	i/P Converter	Foxboro model 69TA-1.

3.6 AUXILARIES

The distillation column was connected to the service lines in the Department of Chemical Engineering and supplied with dry saturated steam at 550 kPa, compressed air at 550 kPa, and cooling water at 400 kPa and 15-18°C. Miscellaneous sections of the column are described in the following sections.

3.6.1 The Feed System

The distillation column feed system is shown in figure 3-6. A 270 litre feed tank was installed so that fluctuations in the feed composition caused by changes in the column holdup would be minimal (the average column holdup was 61 litres). The feed was drawn from the feed tank, through a filter, the feed pump, a double pipe preheater and into the column through one of the tray feed ports as described in section 3.3. The fifth tray from the top of the column was used as the feed tray in

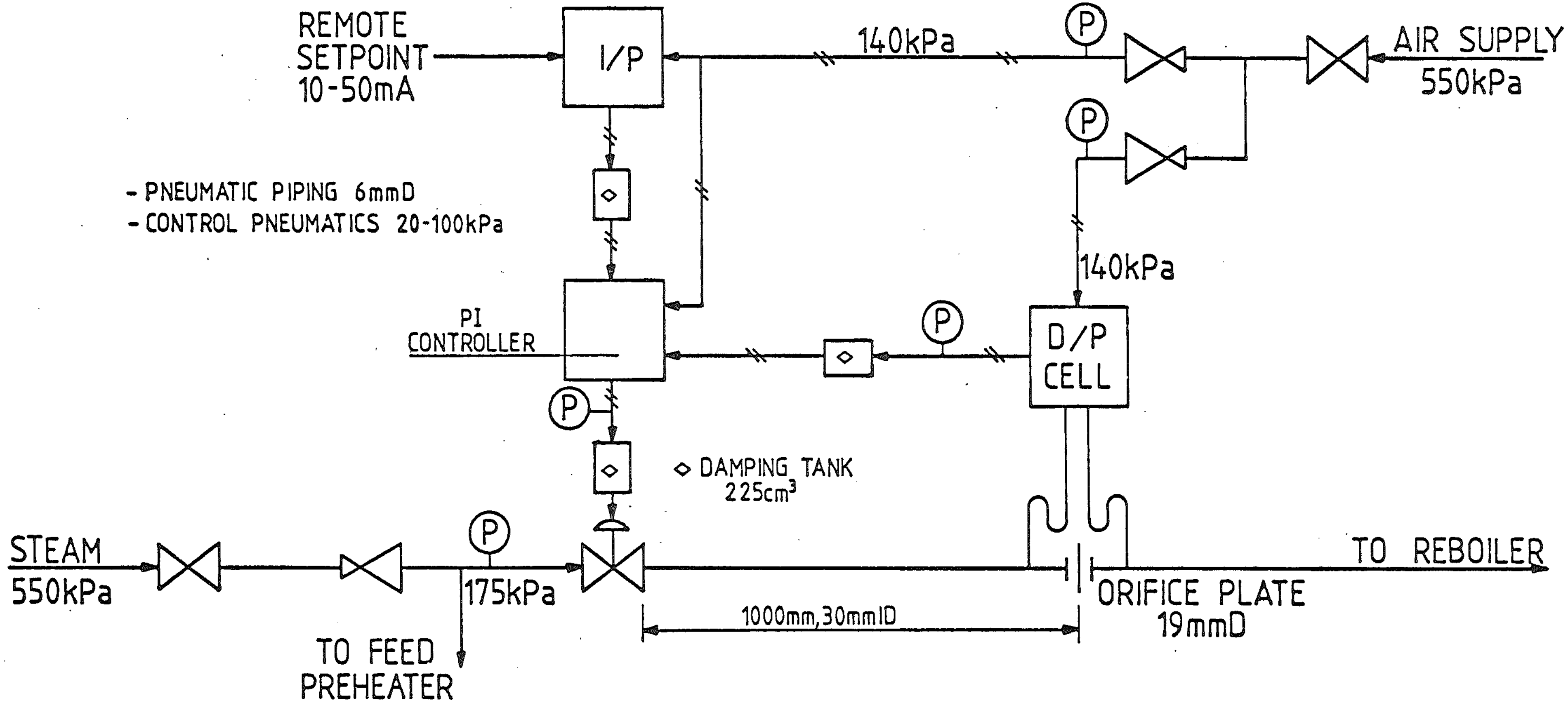


FIGURE 3-5 STEAM FLOW CONTROL SYSTEM

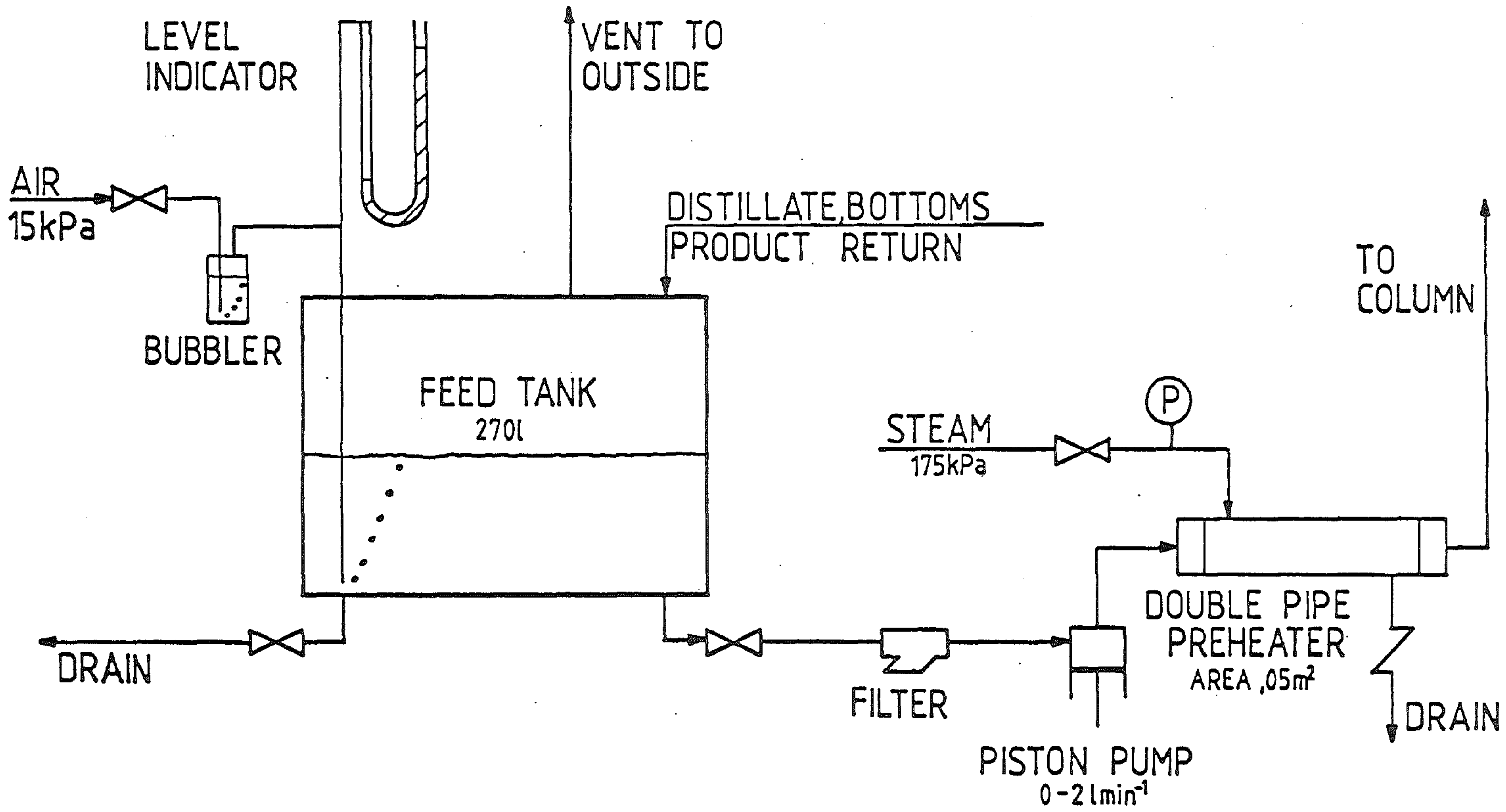


FIGURE 3-6 FEED SYSTEM

this work. Liquid from the siphon breaker for the product pumps was returned to the feed tank via a 25 mm diameter line. The tank was also fitted with a vent which was piped outside the building, and a liquid level indicator comprising a bubbler and Utube manometer. Details of the feed system:

Feed pump - Candy variable stroke piston pump
- capacity 0-2 $\ell\mathrm{min}^{-1}$
- all 316 SS in contact with fluid

Feed Tank - 270 litres
- all 316 SS construction

Feed Preheater - double pipe exchanger
- heat transfer area = 530 cm^2 (internal)
- control : manual adjustment of steam pressure in outer pipe
- 316 SS in contact with working fluid.
- tube dimensions: outer 44 mm ID x 1280 mm
  inner 23 mm ID x 1280 mm

Filter - contains 1 mm gauze to remove solid matter.

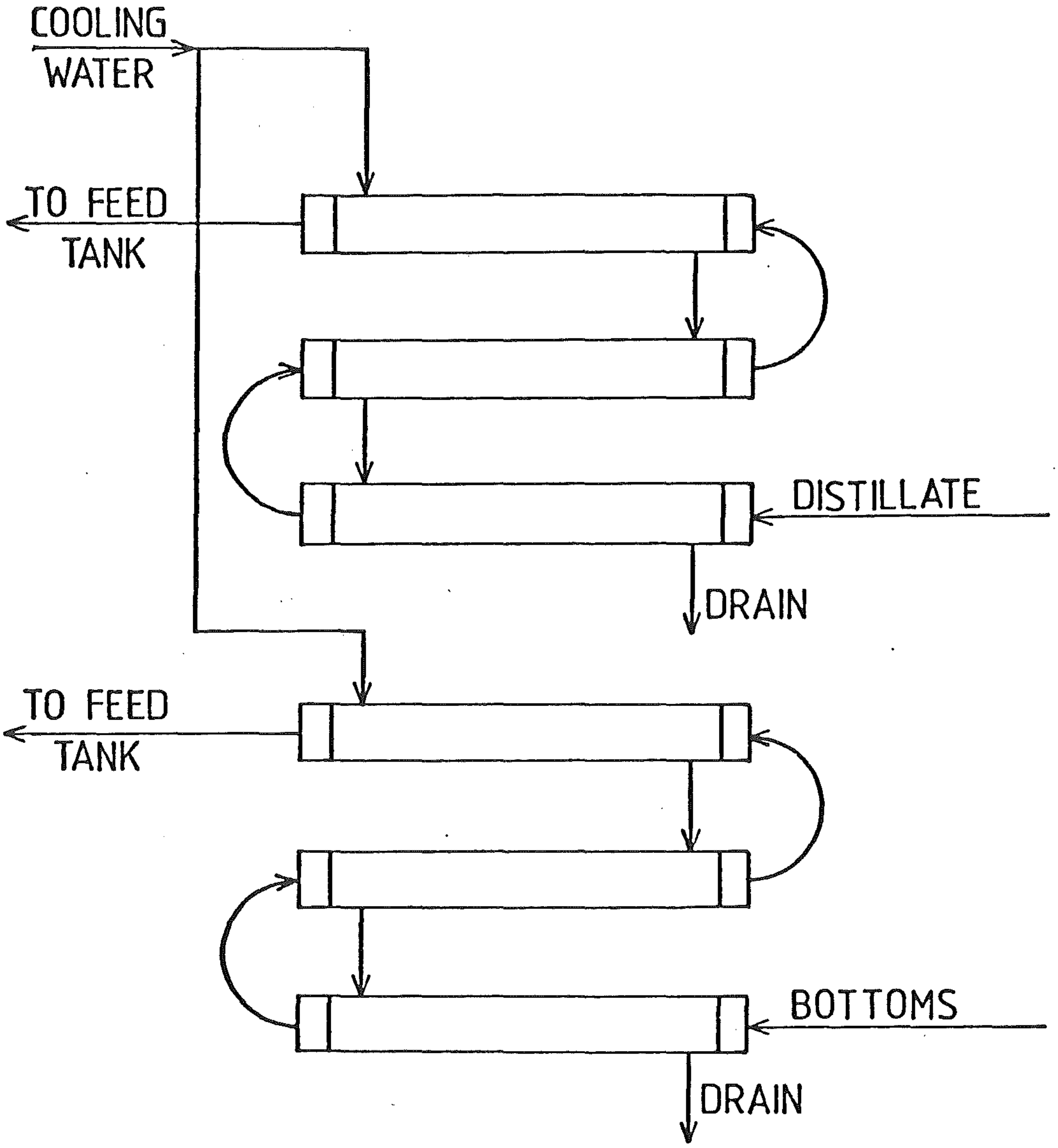
3.6.2 Product Coolers

The bottoms product was removed from the reboiler at its boiling point, and cooled before its return to the feed tank. The temperature of the distillate product was set by manipulating the condenser cooling water flows. Each stream was passed through a three stage countercurrent cooling system as shown in figure 3-7. The exchanger details are

- tube dimensions : outer 44 mm ID x 1280 mm
  inner 23 mm ID x 1280 mm
- heat transfer area = 530 cm^2 per exchanger stage.

3.7 DISTILLATION COLUMN CONSTRUCTION

The previous sections in this chapter have outlined the component



ALL EXCHANGERS-DOUBLE PIPE (0.05m^2)

FIGURE 3-7 PRODUCT COOLERS

parts of the distillation column. The column was mounted in a frame of 50 mm square box section steel and supported on steel plates bolted to the frame under the base of the column, the reboiler, and the reflux accumulator as shown in figure 3-8. Provision was made on these support plates for adjustment of the column attitude so that the column trays could be set horizontal. The column was only sitting in position, restrained from moving sideways by the QVF spring loaded fittings mounted on the QVF flanges to allow thermal expansion on the startup.

The column was constructed with the condensers and the trays in separate vertical columns because of a lack of headroom. Differential thermal expansion, on startup, caused problems in joining the two sections; the tray section was calculated to expand 3 mm, while the condenser section expansion was negligible. A bellows was fitted to each section to absorb the stresses generated by the thermal expansion.

The bellows were designed using the axial symmetric bending of an annular plate formulae given by Flügge (1962), and constructed from six type 316 stainless steel 22 gauge annuli sections alternately welded on the inner and outer edges. A stainless steel section was fabricated to connect the two column sections using the bellows and hydraulically tested to 100 kPa before installation. The completed bellows assembly in place can be seen in plate 3-3. A 60 mm thickness of polystyrene foam plastic was fitted to reduce heat losses through the bellows section. A deflection of 2.5 mm was measured in the tray section on startup, but was accommodated by the bellows.

The major column components were plumbed together using 12 mm diameter type 316 stainless steel tubing for the process fluid lines. All flanges in the process fluid lines were gasketed with rubber protected by PTFE sheaths. Figure 3-9 is a schematic of the column piping, and plates 3-4 and 3-5 show the completed column.

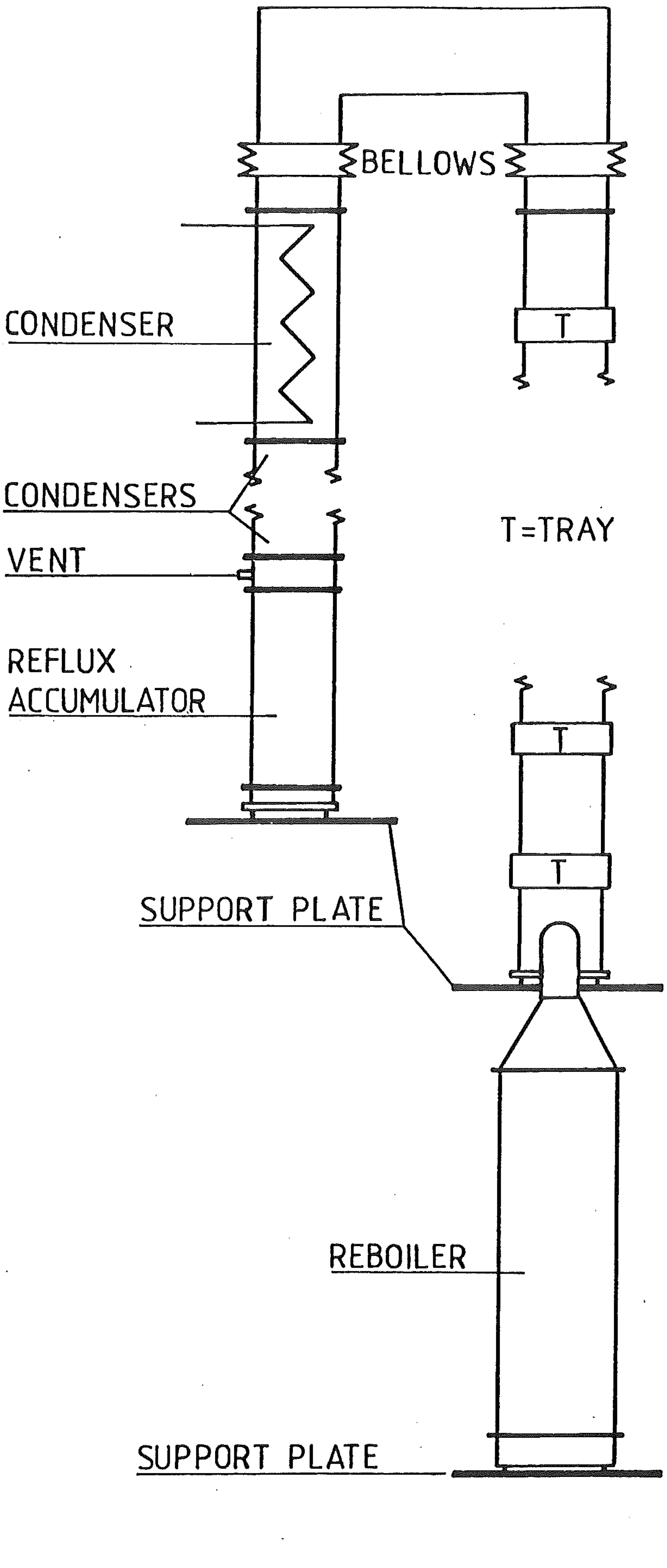


FIGURE 3-8 COLUMN ASSEMBLY

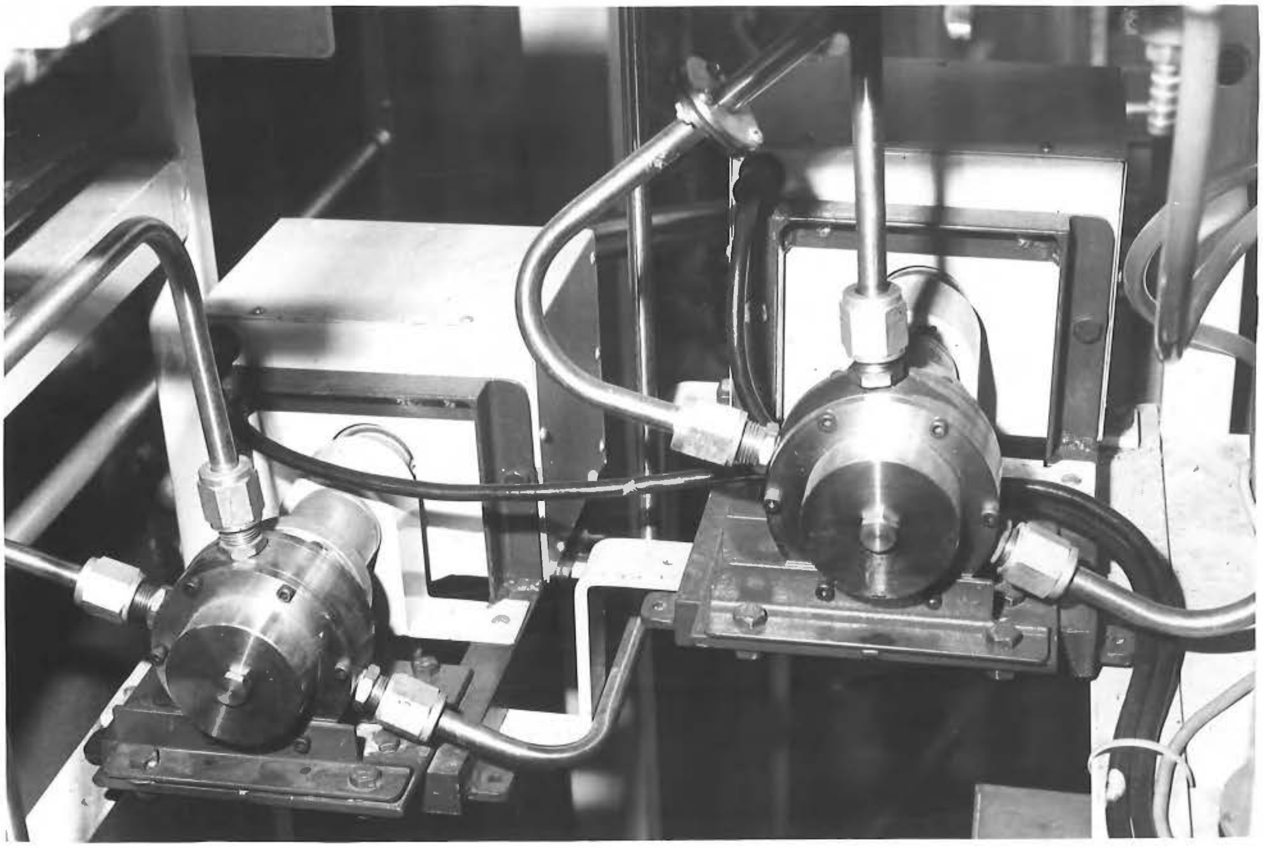


PLATE 3-2 CONTROL PUMPS

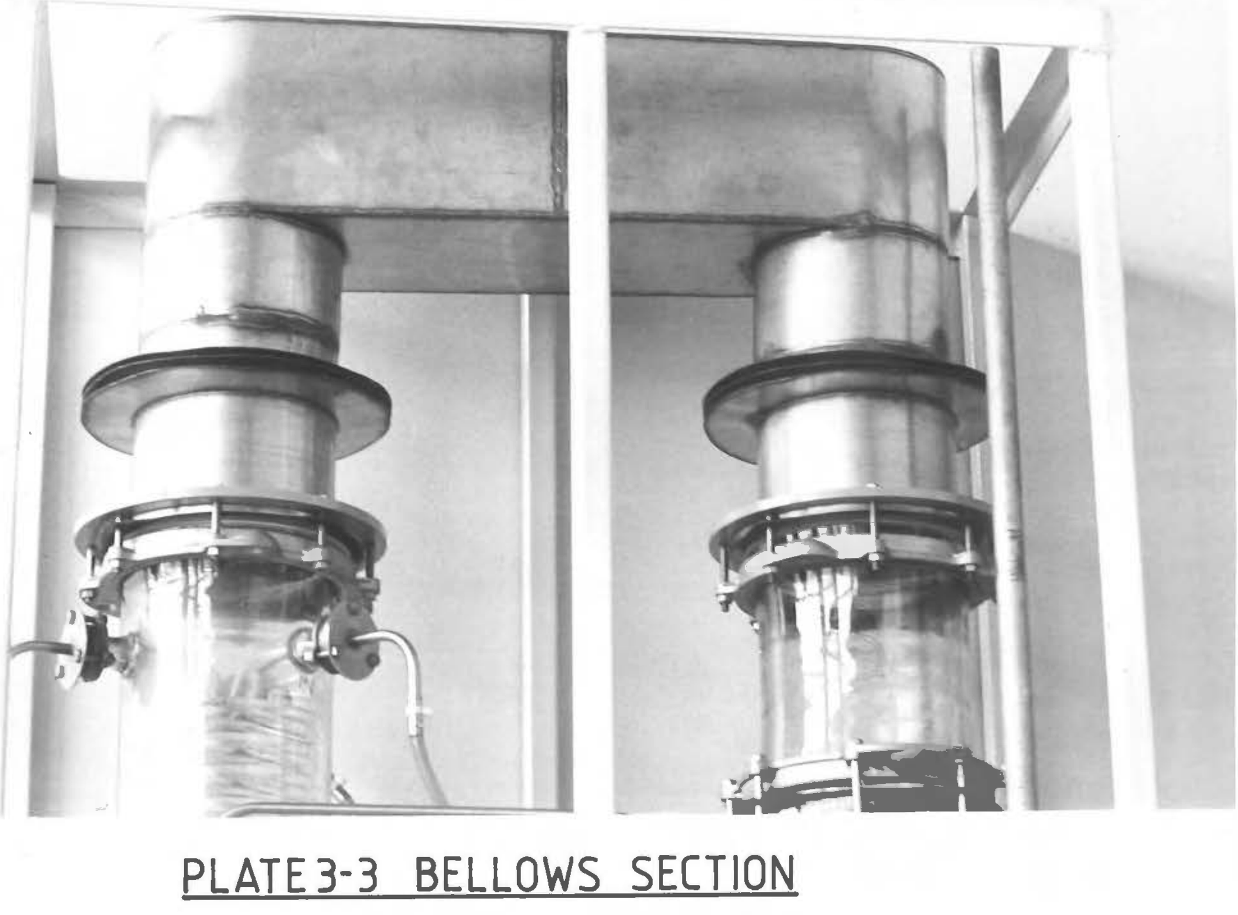
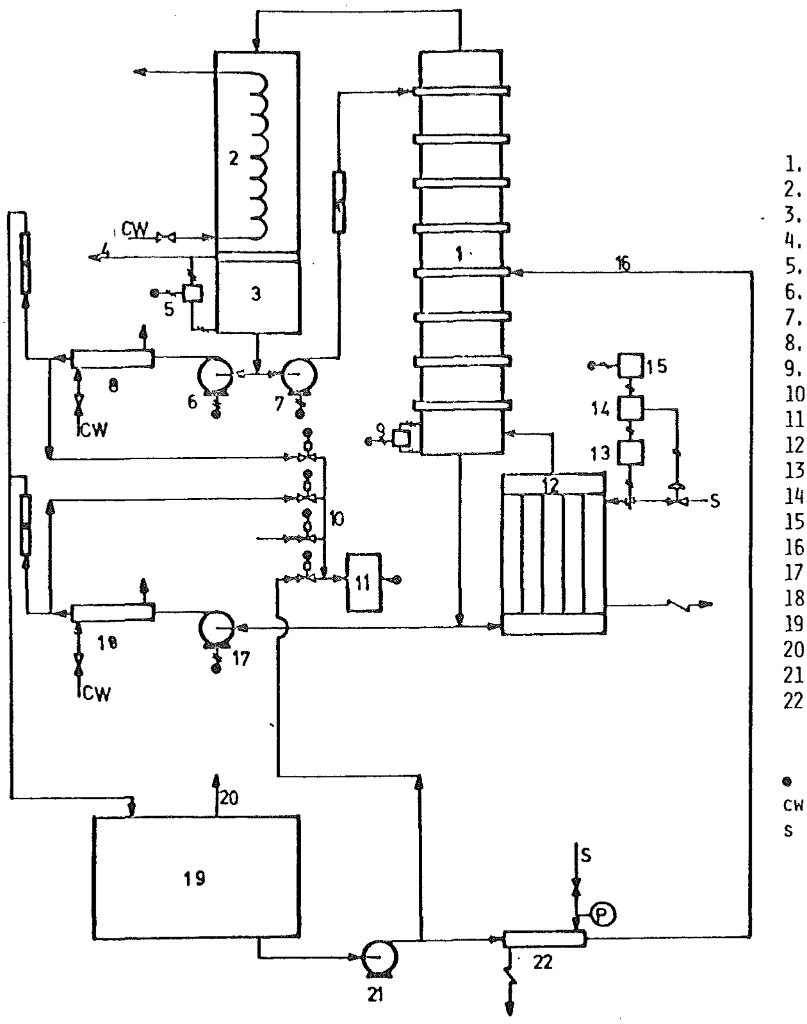


PLATE 3-3 BELLOWS SECTION



KEY

1. COLUMN TRAYS WITH TEMPERATURE SENSORS
2. CONDENSERS
3. REFLUX DRUM
4. ATMOSPHERIC VENT LINE
5. REFLUX DRUM LEVEL SENSOR
6. DISTILLATE PUMP
7. REFLUX PUMP
8. DISTILLATE COOLER
9. REBOILER LEVEL SENSOR
10. REFRACTOMETER SOLENOID VALVES
11. REFRACTOMETER
12. REBOILER
13. PNEUMATIC D/P CELL ON ORIFICE PLATE
14. PNEUMATIC PI CONTROLLER
15. CURRENT/AIR CONVERTER
16. FEED TO TRAY 5
17. BOTTOMS PUMP
18. BOTTOMS COOLER
19. FEED TANK
20. FEED TANK VENT
21. FEED PUMP
22. FEED PREHEATER

● INTERFACE TO CONTROL MICROCOMPUTER
CW - COOLING WATER
S - STEAM

FIGURE 3-9 COLUMN INSTRUMENTATION

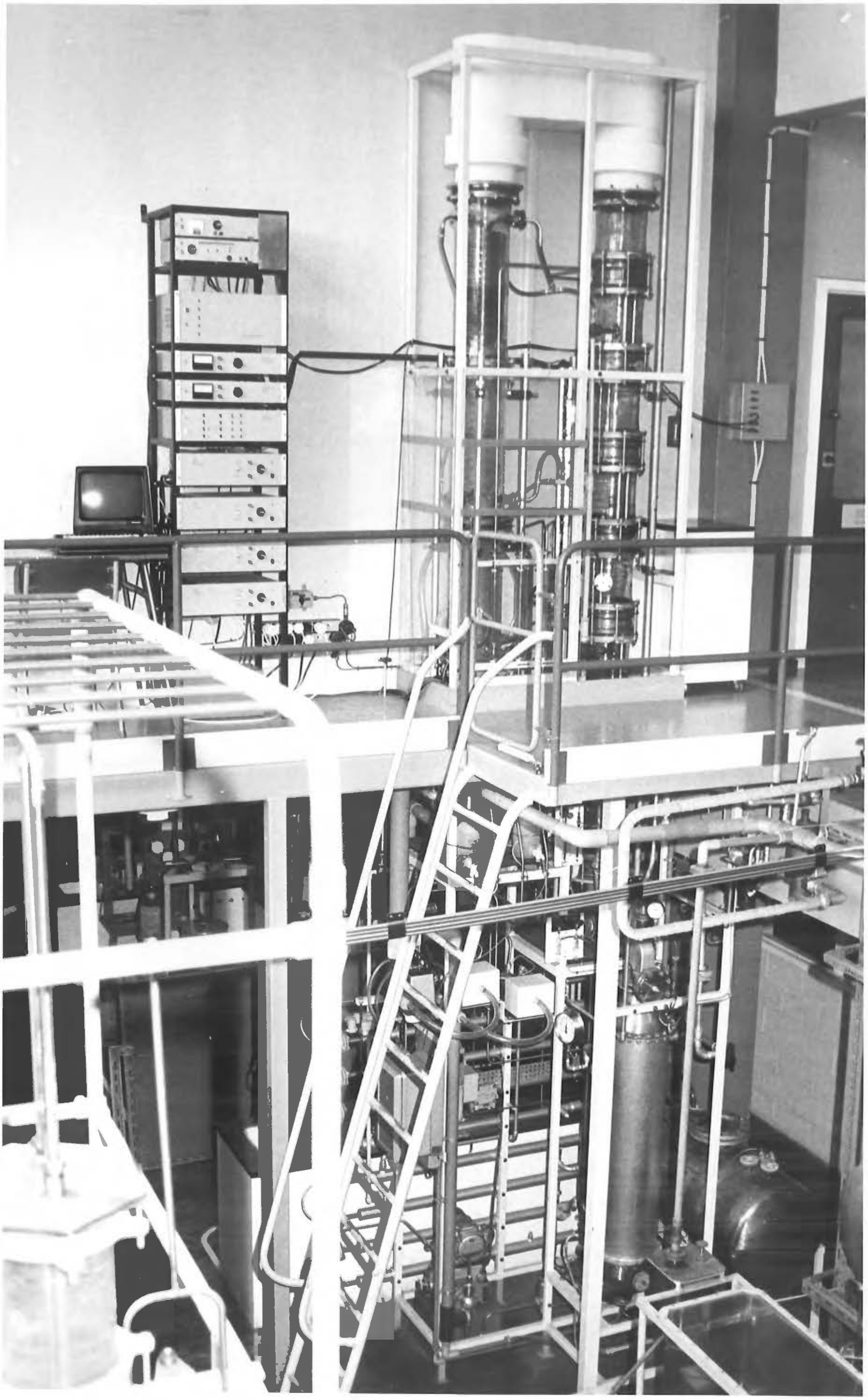


PLATE 3-4 COLUMN

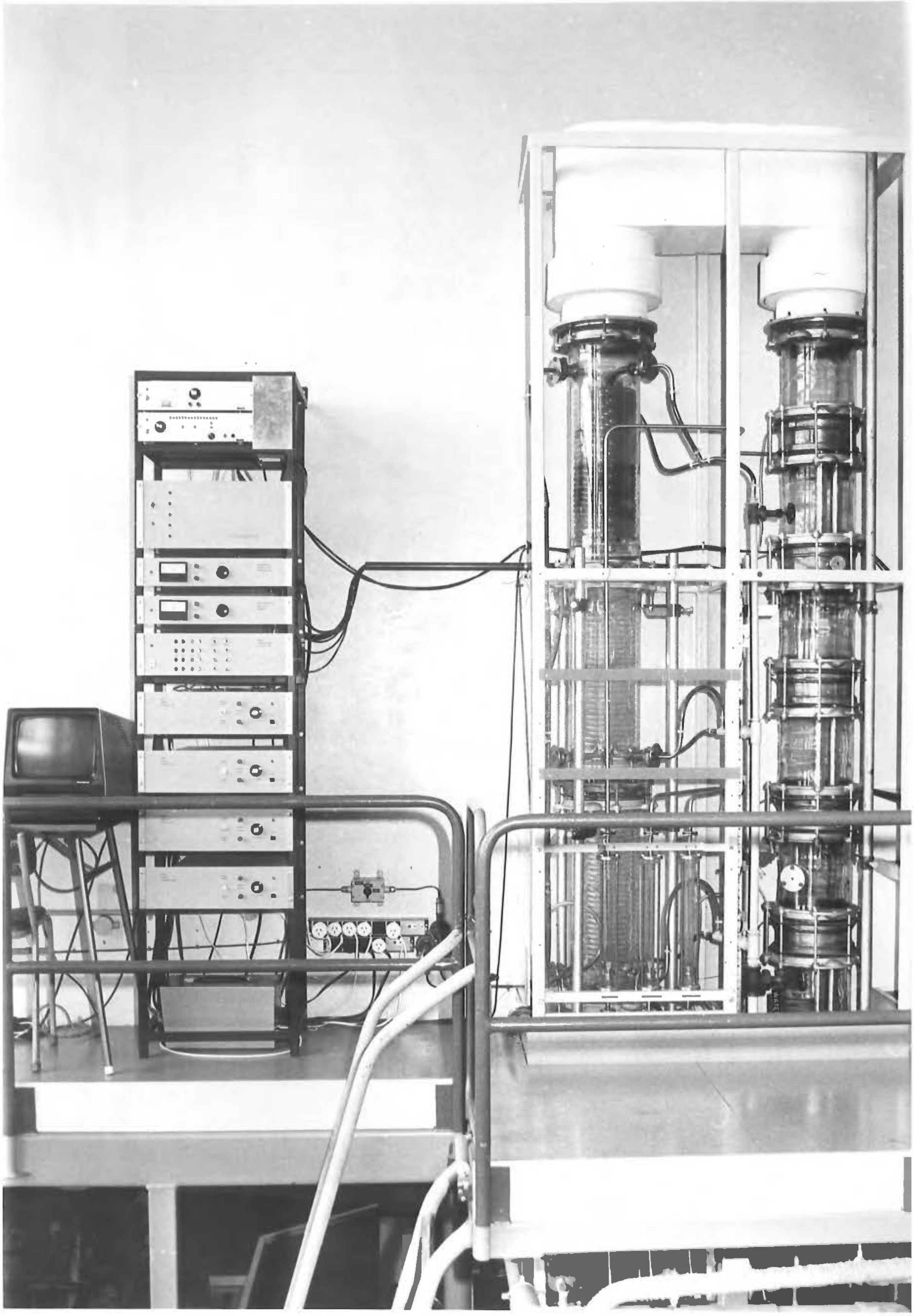


PLATE 3-5A COLUMN-UPPER LEVEL

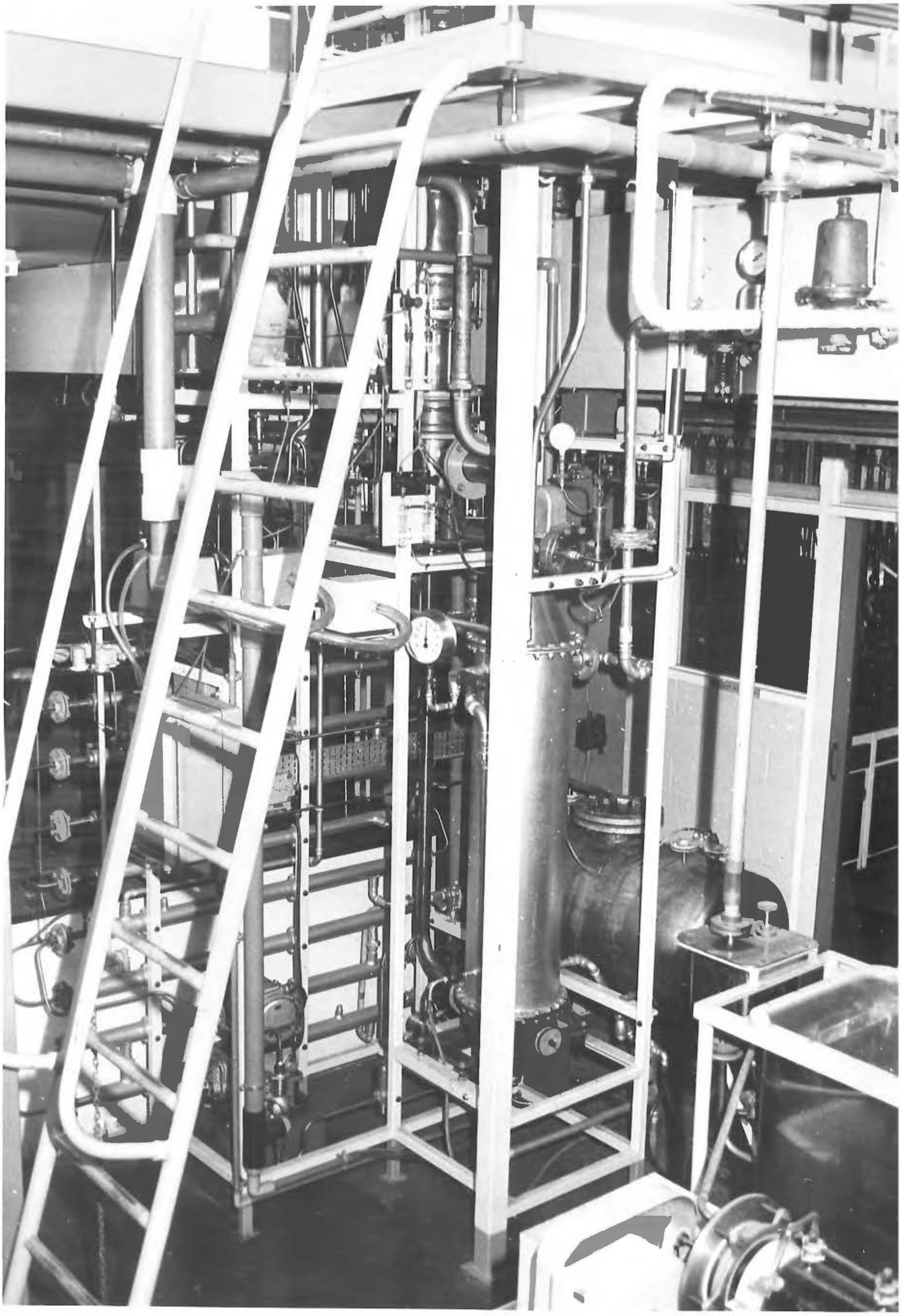


PLATE 3-5B COLUMN-LOWER LEVEL

3.8 INSTRUMENTATION

The distillation column was heavily instrumented to allow flexibility in configuring control strategies. All the instrumentation was constructed to meet an interface standard of 0-10V. In some cases this required the provision of signal conditioning circuits.

3.8.1 Temperature Measurement

Temperatures were sensed on all trays, in the reflux accumulator and in the feed line using National Semiconductor (1974) LX5700 integrated circuits. These devices have an output of 10 mV $°C^{-1}$, and are quoted as having a repeatability of ± 0.2°C and a non-linearity of 1.8°C. To conform to the 0-10V interface standard, the transducer output was shifted and scaled so that 0°C = 0V and 100°C = 10V with the circuit shown in Appendix I.

The integrated circuits were soldered to 316SS probes as shown in figure 3-10 and plate 3-6, and were electrically insulated from the column tray by a polyethylene bush.

Each sensor was individually checked in a water bath, and found to exhibit a slightly different characteristic within the manufacturer's specifications. The calibrations showed that the linearity error could be reduced to a maximum of 0.7°C in the range 60-100°C. Further accuracy could be obtained by using a calibration for each individual transducer. A typical time reponse of a probe to a step change in fluid temperature of 50°C is shown in figure 3-11. The transfer function can be approximated by

$$G(s) = \frac{1}{\tau s + 1} \quad \text{where} \quad \tau = 2.6s$$

3.8.2 Level Sensors

National Semiconductor (1974) integrated circuit differential pressure sensors were selected to sense liquid levels. The devices used

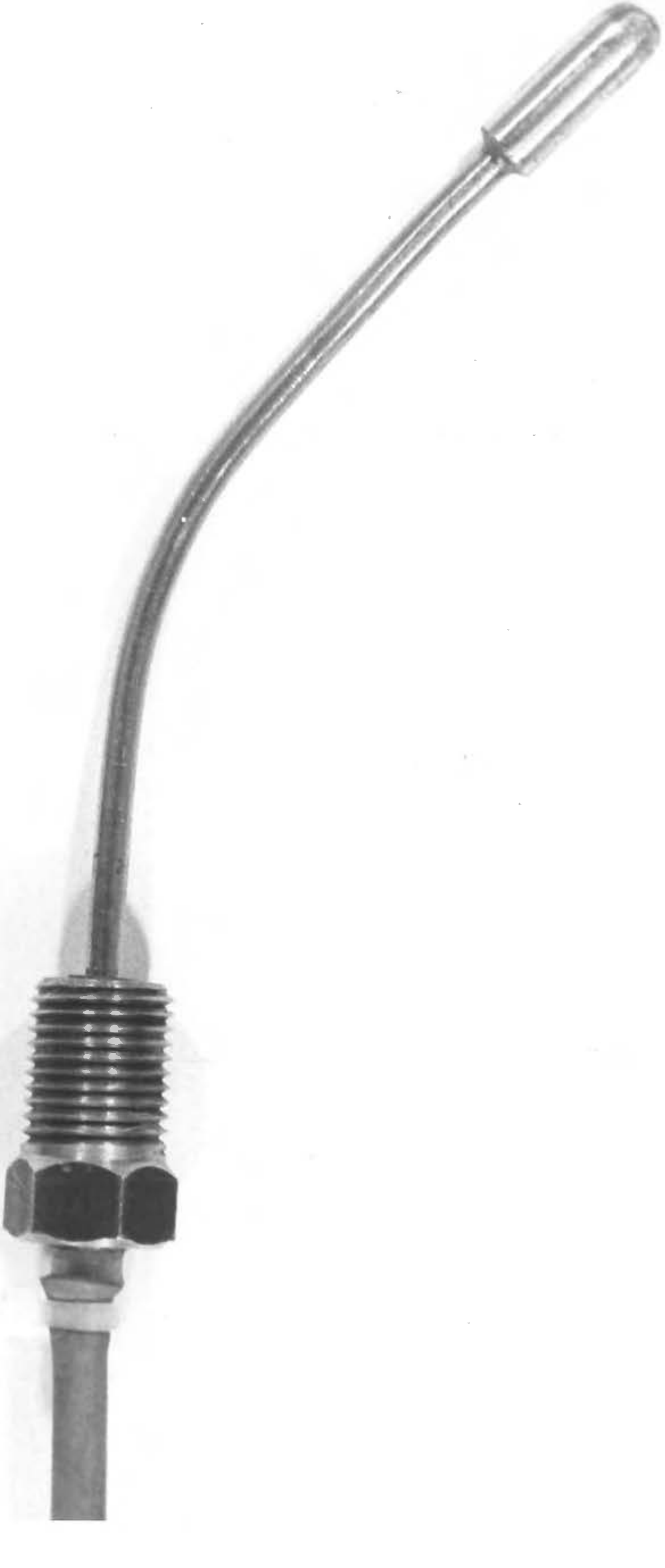


PLATE 3-6 TEMPERATURE PROBE

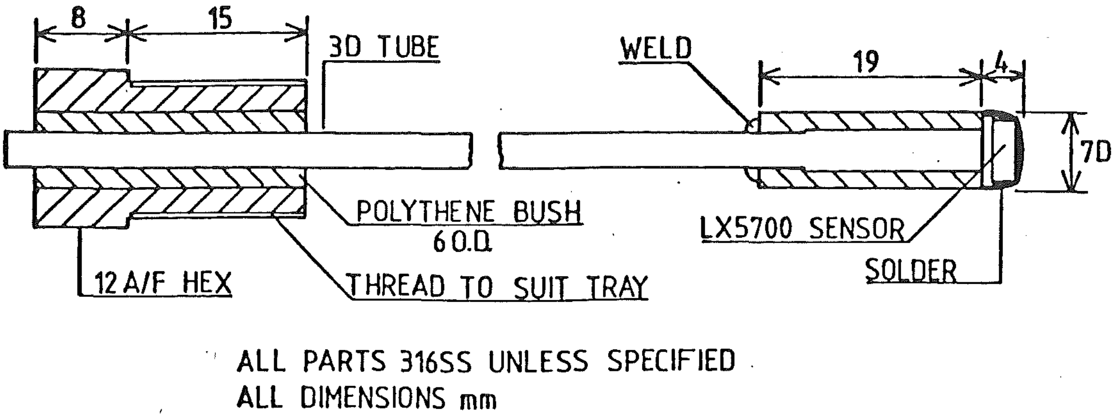


FIGURE 3-10 TEMPERATURE PROBE

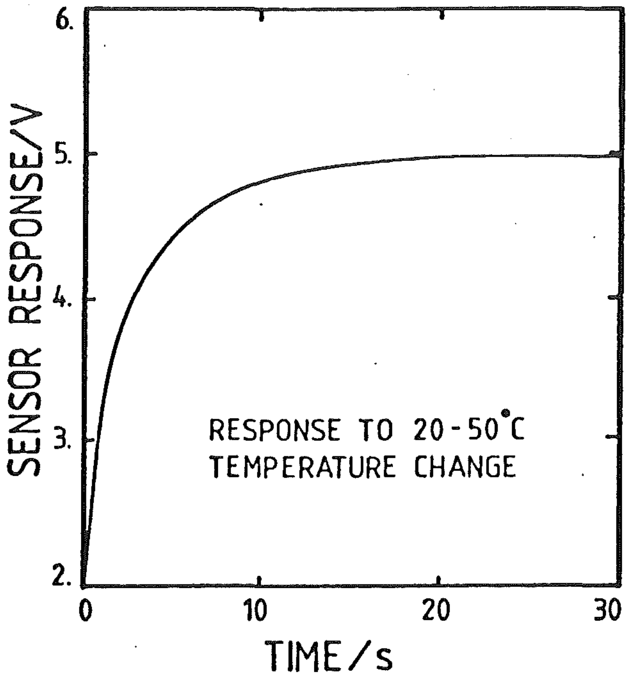


FIGURE 3-11 TEMPERATURE PROBE RESPONSE

were LX1601DF sensors fitted with rubber socks filled with silicone oil to prevent contact between the sensing diaphragm and the process fluids. Further, to allow remote siting of the sensors away from the process, conventional pneumatic bubblers were used.

The outputs of the LX1601DF devices were scaled to the range 0 - 10V by the circuit shown in Appendix I. The response of the sensor was reported as 2.5 - 12.5V for ± 35 kPa by the manufacturer (National Semiconductor (1974)) with a repeatability of ± 2% of the span. The sensitivity of the sensor can be calculated as:

$$\begin{aligned} \pm\ 35\ \text{kPa} &= \pm\ 3.57\ \text{m W.G.} \\ \pm\ 2\%\ \text{of span} &= \pm\ .02 \times 7.14\ \text{m W.G.} \\ &= \pm\ 0.14\ \text{m W.G.} \end{aligned}$$

the temperature coefficient of the LX1601DF was negligible. However in practice, the LX1601DF was found to be more sensitive than the worst case figure of ± 0.14 m W.G. A short term resolution of ± 0.02 m W.G. was observed with a longer term drift in the range ±0.07 m W.G. occurring over a period of 8 - 10 hours. This long term drift could prove to be a problem in circumstances where tight level control is required.

The sensor on the reboiler level was operated in true differential form, while the reflux accumulator level sensor used atmospheric pressure as a reference. The level sensors arrangement is shown in figure 3-12.

3.8.3 Pressure Sensor

Overall column pressure drop was sensed from the upper pressure tapping point of the reboiler level sensing system to atmospheric pressure. A National Semiconductor differential pressure sensor, the LX1601DF, was used, and the same comments made for the level sensors apply. This sensor was used to detect malfunctions in the column operation such as loss of liquid on the trays, due to lack of reflux flow or excessive vapour rates, and flooding. The location of the pressure sensor is shown in figure 3-12.

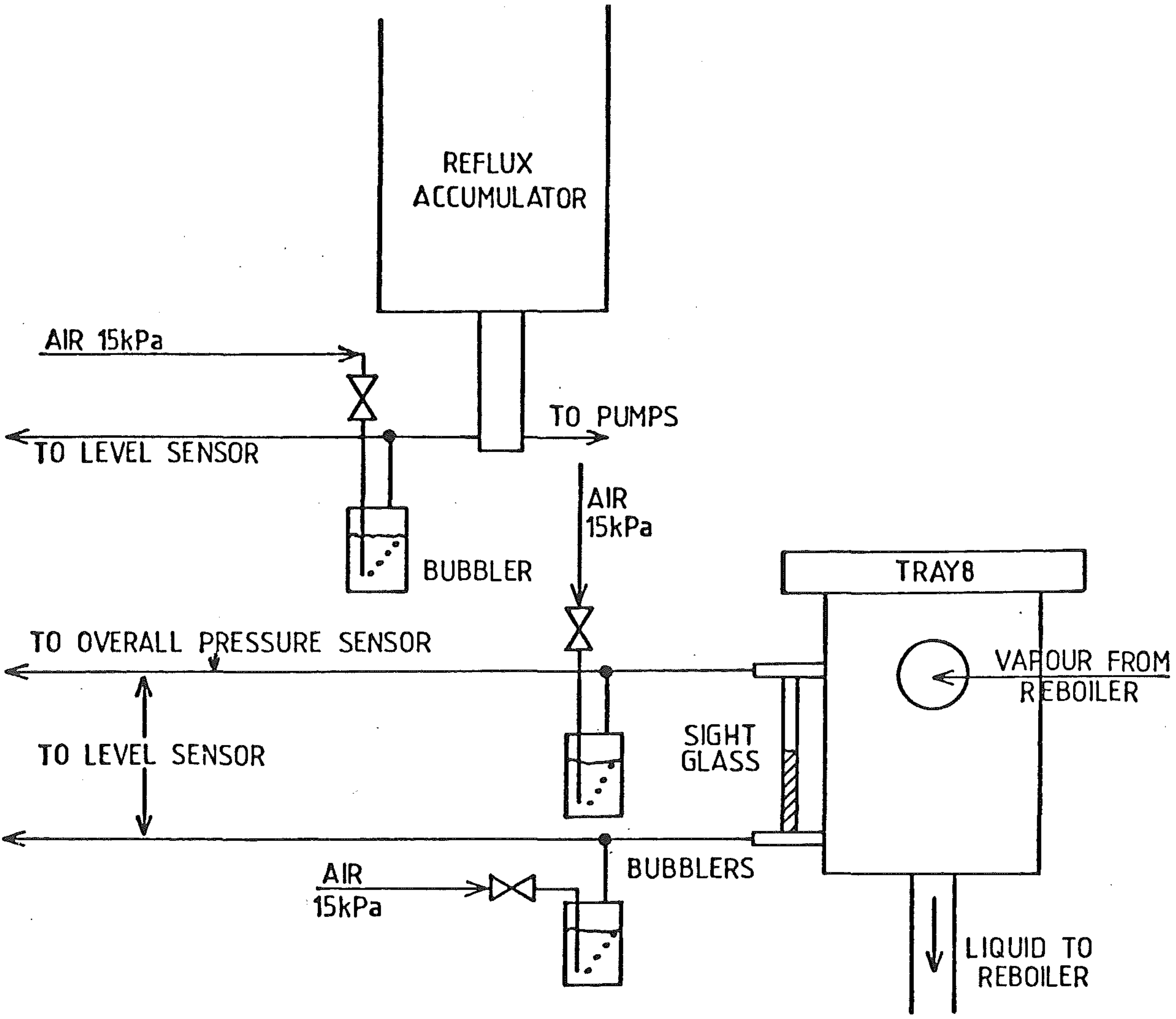


FIGURE 3-12 LEVEL AND PRESSURE TAPPINGS

3.8.4 Composition Sensor

A Phillips OKOMETER R industrial in-line refractometer was installed on the original column to monitor distillate product composition. The refractometer was modified and recommissioned to measure the composition of four methanol/water liquid streams.

The OKOMETER R was an in-line instrument and operated as follows: two parallel light beams of equal intensity were chopped with a rotating segmented plate and passed through the sampling cell and the reference cell respectively. A condenser lens focussed both beams on to a phototube; if the beams were of equal intensity there was no ac output from the phototube. If, however, there was some absorption in the sample cell, the intensity of the two beams was no longer equal and the resulting alternating current produced by the phototube was amplified and rectified to give a dc output signal. Within the sample cell, the light beam was bent so as to strike a sample liquid/glass prism interface at a constant angle. At the interface some of the light was reflected, and some was refracted into the sample liquid and absorbed. The change in light intensity of the beam leaving the cell was a measure of the amount of refracted light (and hence the refractive index of the sample liquid), and was measured by comparison with the light beam passing through the sample cell. The liquid sample stream passed through a heat exchanger which controlled the sample temperature. A thermostatically controlled water bath supplied the heat transfer medium to the heat exchanger. The interior of the instrument containing the cells, optics and electronics was also maintained at a constant temperature by a temperature controller with a fan for air circulation. Further details of the OKOMETER R can be found in the Phillips instruction manual.

A number of modifications were made to the refractometer to match it to the requirements of the distillation column. A manifold of four solenoid valves was added to the refractometer sample input. The solenoids were selected by the states of two TTL logic signals through the interface shown in Appendix I. The output of the refractometer amplifier (0-200μA d.c.) was scaled to 0-10V d.c. by the amplifier shown in Appendix I. Additional filtering was introduced into this amplifier to reject low frequency noise in the refractometer output signal due to bubble formation in the liquid sample streams, and the optical chopper. The filter used was a first order lag with a corner frequency of approximately 0.15Hz.

The optical system of the OKOMETER R was found to be unsatisfactory for the methanol/water system used in the distillation column. The reference cell contained a coloured filter (green) which absorbed too much of the light passing through the reference cell. In order to balance the cell, it was necessary to almost completely block off the sample cell with the zero diaphragm, and this in turn caused problems with poor focusing of the light beams on the phototube. Removing the filte: from the reference cell and placing a green perspex filter in both the sample and reference light beams allowed better focusing on the phototube and de-sensitised the zero diaphragm mechanism.

Temperature control of the sample stream was found to be very important as shown in Appendix II. The sample stream was cooled/heated in a heat exchanger comprising 1.7 m of 3 mm ID stainless steel tubing. The cooling/heating was controlled by water from a thermostatically controlled water bath pumped over the heat exchanger tubing. The water bath was controlled by an on/off solid state relay switched by a mercury/glass thermometer. The sample temperature was measured by an LX5700 integrated circuit sensor mounted in the outlet line of the refractometer. This device used a signal conditioning circuit similar to that described in Appendix I.

The construction of the refractometer and its auxilaries is shown in figure 3-13, and the installation of the refractometer is shown in figure 3-9.

The response time of the modified refractometer was measured so

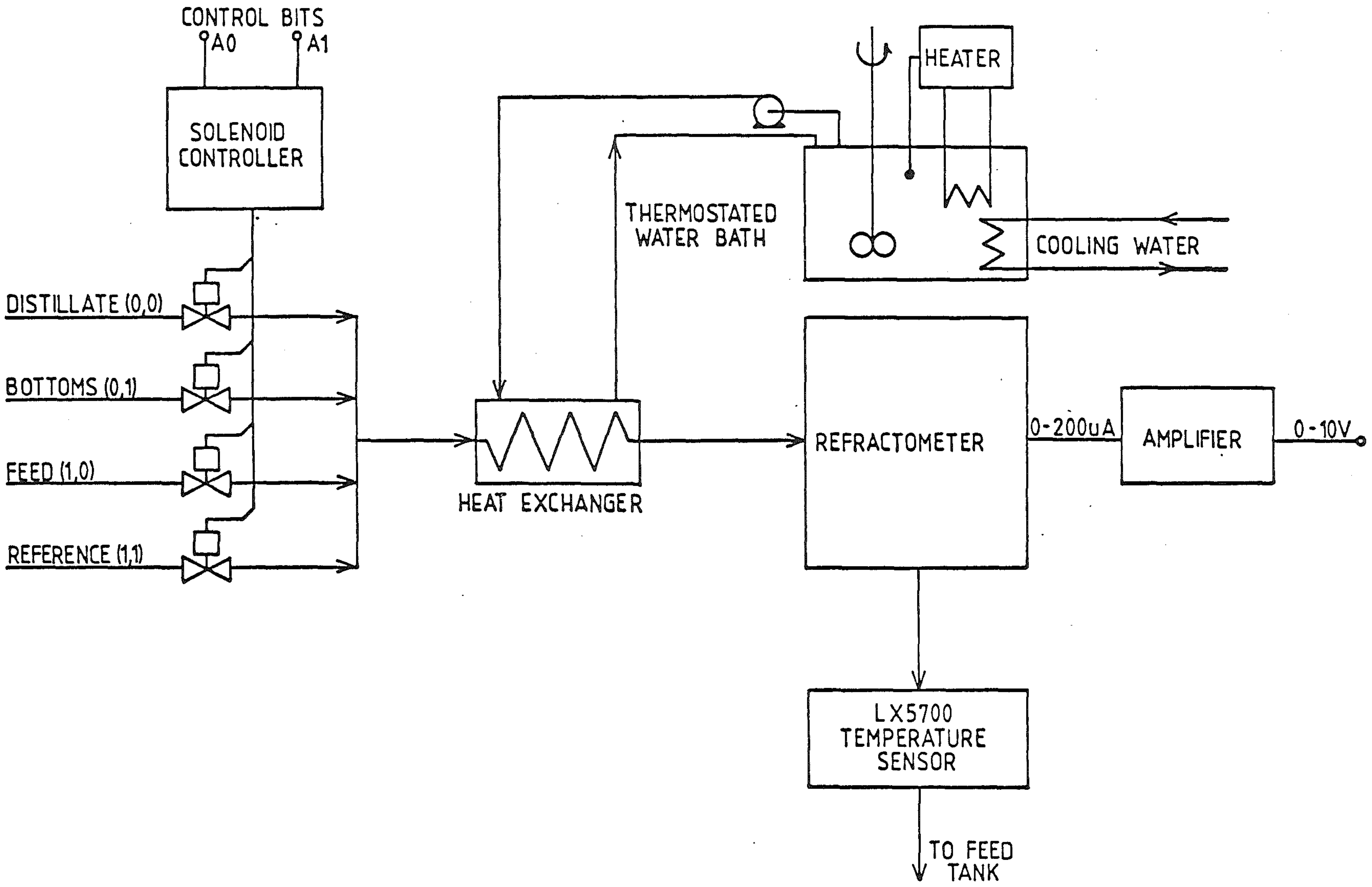


FIGURE3-13 ON-LINE REFRACTOMETER

that a sampling schedule could be formulated. A typical response of the refractometer to a channel change is shown in figure 3-14. Using the method of Sundaresan and Krishnaswamy (1978), a transfer function was fitted to this response:

$$G(s) = \frac{Ke^{-15.6s}}{3.7s + 1} \qquad \text{(in seconds)}$$

The minimum time between channel changes was chosen to be 60 seconds to allow for the dynamics of the sampling system.

3.9 SELECTION OF A BINARY SYSTEM

A number of important factors were considered in selecting a binary system for the distillation column. Methanol and water were chosen on the basis of the following points:

(i) Safety - a low toxicity, low risk of explosion system was required because the column was located in a hazardous area, and because the column would be operated by undergraduate students in the future. Methanol/water while having some toxic effects is regarded as safe. A Pye gas detector was set to alarm when the concentration of methanol in the air reached 1.5% (25% of the lower explosive limit of methanol in air).

(ii) Properties - methanol/water mixtures do not exhibit constant relative volatility nor constant molal latent heat of vaporation and can be described as non-ideal. Methanol is completely miscible in water and does not form an azeotrope.

(iii) A number of other researchers have used methanol/water with similar columns, and by using the same binary system, some comparisons were possible between published results and experimental work. The separation of methanol from water is performed commercially by distillation to recover methanol e.g. in the manufacture of formaldehyde.

Industrial grademethanol supplied by Imperial Chemical Industries (ICI) to the specifications given in Table 3-1 was used in the column.

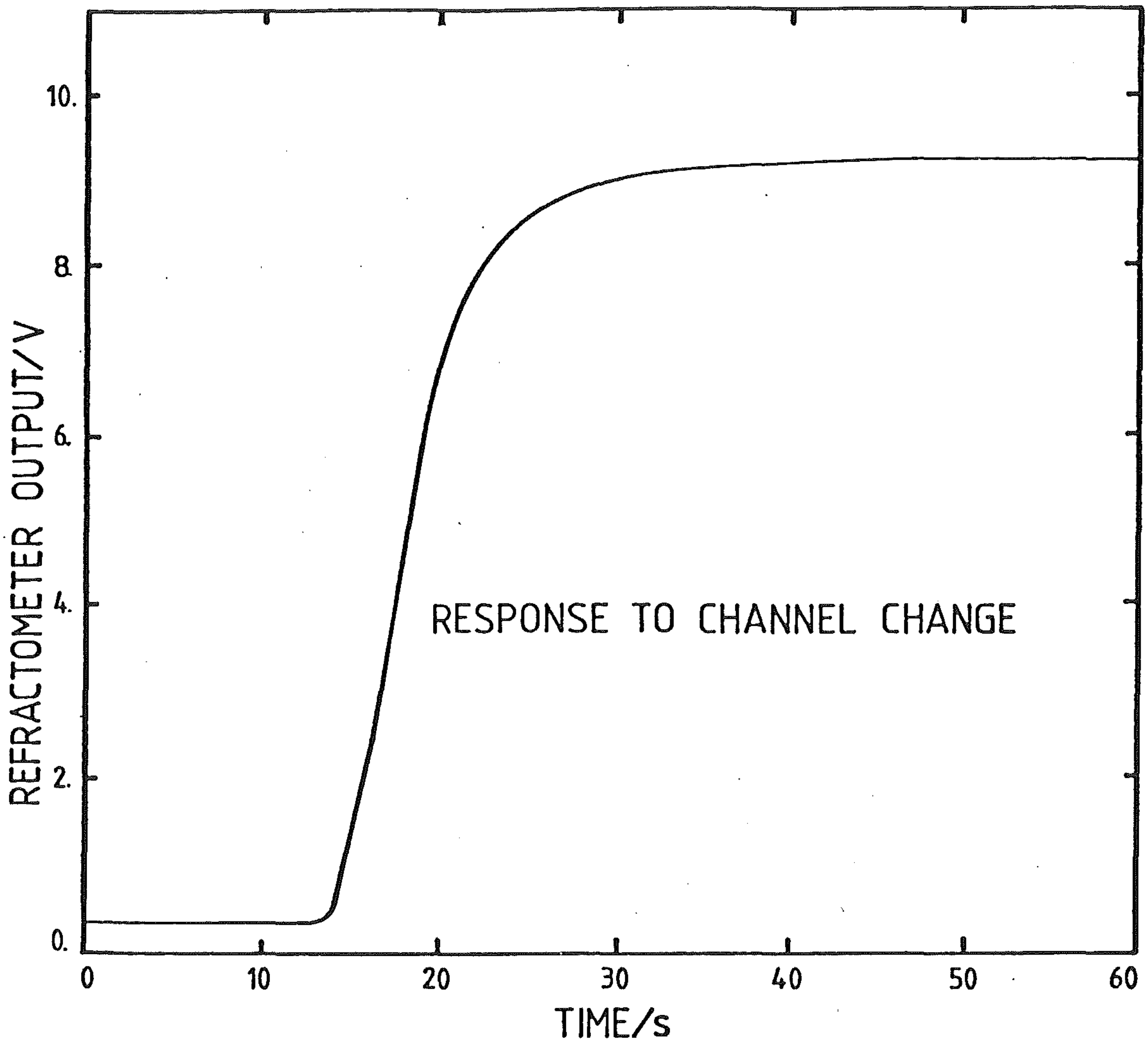


FIGURE 3-14 REFRACTOMETER RESPONSE

TABLE 3-1

SPECIFICATIONS OF METHANOL/WATER

Methanol:

Appearance	- clear, colourless, free from suspended matter and sediment.
Purity	- water < .05% wt
	- aldehydes and ketones < .003% wt as acetone
	- alkalinity < .0005% wt as ammonia
	- acidity < .003% wt as acetic acid
	- sulphur < .0005% wt as sulphur

Water: Distilled in a Manesty still.

3.10 COMMISSIONING

Prior to assembly of the column components, all sections were thoroughly washed in a trichloroethylene bath, and scrubbed with boiling water to remove all traces of cutting oil and any remaining residues from the old column. Once assembled the column was twice filled with distilled water and operated at total reflux. During this operation the flanges and joints were checked and the operation of the bellows section described in section 3.7 was checked.

The column was then loaded with 87.5 litres of methanol (70 kg) and 40 litres of distilled water (40 kg). Commissioning of the instruments and controls then commenced.

3.10.1 Temperature Sensors

Calibration of the sensors described in 3.8.1 was carried out in a stirred water bath using mercury-in-glass reference thermometers. An ice bath was used to set the zero point of the scale. National Semiconductor (1974) specifications quoted the LX5700 device with a maximum of 1.2°C non-linearity, however the devices used in this work in general were better than 0.7°C. The results from each probe were fitted with a second order polynomial which gave a maximum discrepancy between the fitted line and the data points of 0.2°C. This small deviation could be attributed to the errors involved measuring the true temperature during the calibration procedure and was within the manufacturer's repeatability specification. The calibration data, and the fitted quadratic correlations are given in Appendix II.

3.10.2 Pressure and Level Sensors

These devices were installed and operated as described in sections 3.8.2, 3.8.3. The desired operating levels were set up in the column and the level sensor readings were noted and used as setpoints in the level control loops. Calibration of the level sensors over a wide range was not necessary. A similar procedure was adopted to find the upper and lower limits on the overall column pressure drop to be placed in the alarm detecting software. The limits were chosen based on the overall column pressure drop existing when the column was operating at its desired state.

3.10.3 Composition Sensors

The OKOMETER in-line refractometer was installed and checked out according to section 3.8.4. This refractometer was calibrated against an Abbe refractometer (Carl Zeiss No. 202553) which had been checked with solutions of known concentration at several temperatures. These results are listed in Appendix II.

The operating temperature of the refractometer was chosen to be 25°C as a compromise between the heating/cooling capacity of the refractometer thermostat and the temperatures of the liquid sample streams (distillate $\simeq$ 20°C, bottoms $\simeq$ 35°C). At 25°C, the thermostat maintained an approximate 50% on/off duty cycle.

Interactions between the zero and span adjustments on the OKOMETER refractometer necessitated the following calibration procedure. Two reference streams were connected to the refractometer reference channel, one reference being pure methanol and the other a mixture of approximately

.40 mole fraction methanol of known composition and refractive index. Having selected the refractometer reference channel, the procedure was to switch between the two references, adjusting the zero point on pure methanol and the span on the mixture, until the voltage output settled at those levels determined from the calibration in Appendix II. This tedious procedure was necessary to ensure that the refractometer readings were correct. Reproducibility checks on the Okometer refractometer showed it to be capable of measuring the compositions of the column products within ± .004 mole fraction. Its performance on measuring feed composition was much less satisfactory due to the very steep nature of the calibration chart over the feed composition range. The error in the feed composition was estimated to be ± 0.020 mole fraction.

Close temperature control was found to be important as the refractive index is a strong function of temperature as shown in Appendix II. The Okometer thermostat was found to control within ± 0.2°C, and using the refractive index data in Appendix II, this was predicted to cause a variation in the estimated composition of approximately ± 0.004 m.f. This error combined with the errors in the calibration chart formed the final estimate of ± 0.004 m.f. on the estimated composition for the ranges 0.0 to 0.20 m.f. and 0.80 to 1.00 m.f. The likely error on the feed composition estimate was found to be considerably higher - of the order of ± 0.02 m.f., hence all feed composition measurements were cross checked against density measurements with a density bottle, off-line.

Operation of the Okometer refractometer required careful attention to setting up and calibration. It was necessary to allow at least four hours for the air thermostat to bring the refractometer case and contents to its working temperature (30°C) and about one hour for the water bath thermostat to reach its stable operating temperature (25°C). The presence of the relatively cold sample cell in the heated refractometer caused condensation to form on the cell; a small amount of silica gel placed within the refractometer case removed this problem. Once set up in accordance with the procedures outlined in this section, the refractometer could be relied on to give good estimates of tops and bottoms composition for periods of up to 12 hours. After this, checks against the standard solutions were required to prevent small drifts causing measurement errors.

3.10.4 Pumps and Rotameters

Some modifications were required before the variable speed pumps would work reliably. Once the individual pumps were correctly setup according to their conditions of operation, they behaved well. A small problem with the reflux pump was the gradual drift in flowrate with time for a constant rotational speed. This drift was small but continuous and was thought to be due to wear and movement in the vanes and cams system of the pump resulting in slippage. The head on the delivery side of the pump was increased by fitting a 200 mm length of 3 mm diameter stainless steel tube inside the discharge line. This flow restriction increased the back pressure on the pump, required a higher rotational speed for a given flowrate, and steadied the drift problem. The pump flowrate was also found to be suction head dependent due to some flow inadequacies in the pump design. Precise regulation in the level controllers held the suction pressures constant and minimised this problem. For the reflux pump, reliable constant flows could be maintained for periods up to four hours. The stability of the distillate and bottoms pumps was less critical as these were generally operating in the level control loops under feedback control.

The rotameters in the discharge lines of the three pumps were calibrated using water at 20°C because the calibration charts recommended by the manufacturer (Rotameter (1960)) were inaccurate due to inconsistencies in the shape and mass of the floats. Corrections for changes in liquid composition and temperature were estimated using the correlations given by the manufacturer (Rotameter (1960)). The calibrations and the corrections are given in Appendix II.

3.10.5 Steam Flow Loop

The steam flow loop was installed as described in section 3.5.5. The capacity tanks were fitted to the output lines of all the pneumatic devices to prevent the pumping action which effectively upset the control action and threatened to destroy the control valve. The tanks supplied sufficient capacity to damp out this action. The d/p cell was set up so its range matched the available steam flow range with the steam reducing valve set at 275 kPa. The PI controller was tuned to give a reasonably fast response to setpoint changes while maintaining a steady flow without large valve movements for load changes. The best settings were found to be: proportional band = 150%

reset time = 11 min.

The response time of the steam flow control loop was almost instantaneous (≃10s).

3.10.6 Column Holdups

Holdups within the column were measured where possible and calculated otherwise.. The reboiler holdup at normal operating conditions was found to be 40 litres. The maximum reflux accumulator holdup was found to be 25 litres but for most operating conditions, a level of 300 mm corresponding to a holdup of 12 litres was used. Tray holdups were calculated based on an average liquid depth on the tray equal to the height of the weirs, and a maximum backup of 100 mm in the downcomers. The results are summarised below in Table 3-2.

TABLE 3-2

COLUMN HOLDUPS

Measured Holdups	:	Reboiler	40 litres
		Reflux Accumulator	25 litres
Calculated Holdups	:	Tray	1.1 litres
		Feed Tank	270 litres

Total Column Holdup : 61 litres under normal operating conditions.

CHAPTER FOUR

COMPUTER SYSTEMS

CONTENTS

CHAPTER FOUR

COMPUTER SYSTEMS

4.1 INTRODUCTION

The computer system developed for this work was designed to be as flexible as possible and to allow a number of different configurations. A microcomputer with peripherals and process interface was used as a software development system, and a control system. A PDP-11 minicomputer in the Department of Chemical Engineering was connected to the microcomputer to provide program development facilities for the microcomputer, mass storage, utility routines, and a hierachical control scheme. Simple analog proportional only controllers were constructed to provide limited control functions and hardware signal filtering.

Software was written to allow the microcomputer to control its peripherals and the process instruments, and to allow the minicomputer to control the microcomputer. The computational capability of the microcomputer was extended by the addition of a hardware arithmetic unit.

Programs were developed in the minicomputer and then downloaded to the microcomputer for debugging and implementation. A cross-assembler was written to produce code for the M6800 microprocessor. All software operating in the microcomputer was developed using this system.

4.2 ANALOG CONTROLLERS

Four analog proportional only controllers with adjustable setpoints, gains, filter constants and output biases were constructed. The controllers were used to maintain the material balances within the column, when operating without computer control. A circuit diagram of the controllers is shown in Appendix I.

Two of the controllers were constructed with damping on their outputs to prevent rapidly changing outputs causing the speed controllable pump motors to lose synchronisation. The other two controllers were set up with zero setpoint, unity gain and zero bias to serve as filters for the two liquid level signals in the reflux accumulator and the reboiler.

The filter built into each controller was a lag unit operating as a low pass filter. The first order lag unit was constructed after a circuit by Bristow (1972). A series of sinusoidal inputs to the filter established that the filter did approximate a first order lag. The filters on the liquid levels signals were adjusted to a time constant of 5.5 seconds, corresponding to a corner frequency of 0.03Hz . This filtering proved very satisfactory in smoothing the noise in the liquid level signals caused by liquid agitation.

4.3 MICROCOMPUTER

The microcomputer was constructed around an American Microsystems Inc. EVK300 prototyping single board computer. This board used a Motorola M6800 8 bit microprocessor. The major features of the board are summarised in Table 4-1. The prototyping system memory was expanded by the addition of two 4K byte read/write memory (RAM) boards, and two 4K erasable programmable read-only memory (EPROM) boards (using M6834 EPROMS). This gave a total memory complement of 10K bytes of EPROM and 9K bytes of RAM.

Interfacing to the distillation column was organised through a data acquisition unit attached to the CPU via a PIA, and through four 8 bit digital-to-analog converters via two PIAs. The process interface hardware and software will be further discussed in the following sections.

TABLE 4-1

AMI EVK300 PROTOTYPING SINGLE BOARD COMPUTER

- M6800 CPU operating at 1 MHz
- 3 peripheral interface adapters (PIA), each with two 8 bit I/O ports and four control/handshaking lines
- 1 asynchronous communication interface adapter (ACIA) serial interface, RS232 and 20mA current loop, operating at 110 - 9600 baud
- 1K read/write memory (RAM)
- 2K read-only-memory (ROM) containing a small operating system (PROTO) and a collection of subroutines (RS^3)
- 2K erasable programmable read only memory (EPROM) organised as 4xM6834 chips
- EPROM programmer (handling M6834 EPROMS)
- Fully buffered address, data and control busses to allow for system expansion
- An internal timer providing pulses at 100μs and 1ms intervals (used by the EPROM programmer)
- Three types of direct memory addressing (DMA) - halt processor, cycle steal, and multiplex mode.

The 20mA current loop serial port on the EVK300 board was used to control and communicate with the microcomputer. An interface from the PDP-11 minicomputer was connected to this port to allow communication between the two machines. Alternatively, a visual display unit (VDU) was attached to the port for direct interaction with the microcomputer.

Five output control lines from the three PIAs on the EVK300 board were connected via transistor drivers to front panel light emitting diodes (LED). These LEDs were used as indicators and for monitoring programs.

The computational capability of the microcomputer was extended by the addition of an American Micro Devices Am9511 arithmetic processing unit as described in section 4.6.

The microcomputer was assembled in a 480 mm rack-mounting case with the printed circuit boards arranged horizontally. A backplane construction was used with the printed circuit boards mounted in sockets, wire wrapped together to form the backplane. This allowed maximum flexibility in adding to the system; the EVK300, RAM memory, EPROM memory and APU boards all had different pin connections.

The assembled microcomputer dissipated approximately 50 W into its case, and a cooling system was necessary. A 100 mm diameter fan was mounted in one side of the case and arranged to draw air through a cloth filter on the opposite side of the case, across the printed circuit boards. The integrated circuits (especially the RAM chips) operated at a reasonable temperature (35°C).

Power for the microcomputer was supplied by a single power supply mounted externally to the case. The circuit diagram of the power supply is shown in Appendix I. The 5V regulator had current limiting protection up to a maximum of 10A and an overvoltage trip out. The regulator was based on a conventional series regulator design. The ± 12V supplies were rated at 1.5A maximum and also had current limiting protection. A -50V supply was included for EPROM programming on the EVK300 board.

4.4 MICROCOMPUTER/PROCESS INTERFACE

The microcomputer interface to the distillation column and instruments was used to monitor process variables and set controls. All interfacing to the process was through Motorola M6820 parallel interface adapters (PIA). These devices provided two parallel 8 bit ports and 4 handshaking lines, and could be configured under software control. The following sections describe the hardware used in the microcomputer/ process interface and the connections and operation of that hardware. The interface structure is shown in figure 4-1.

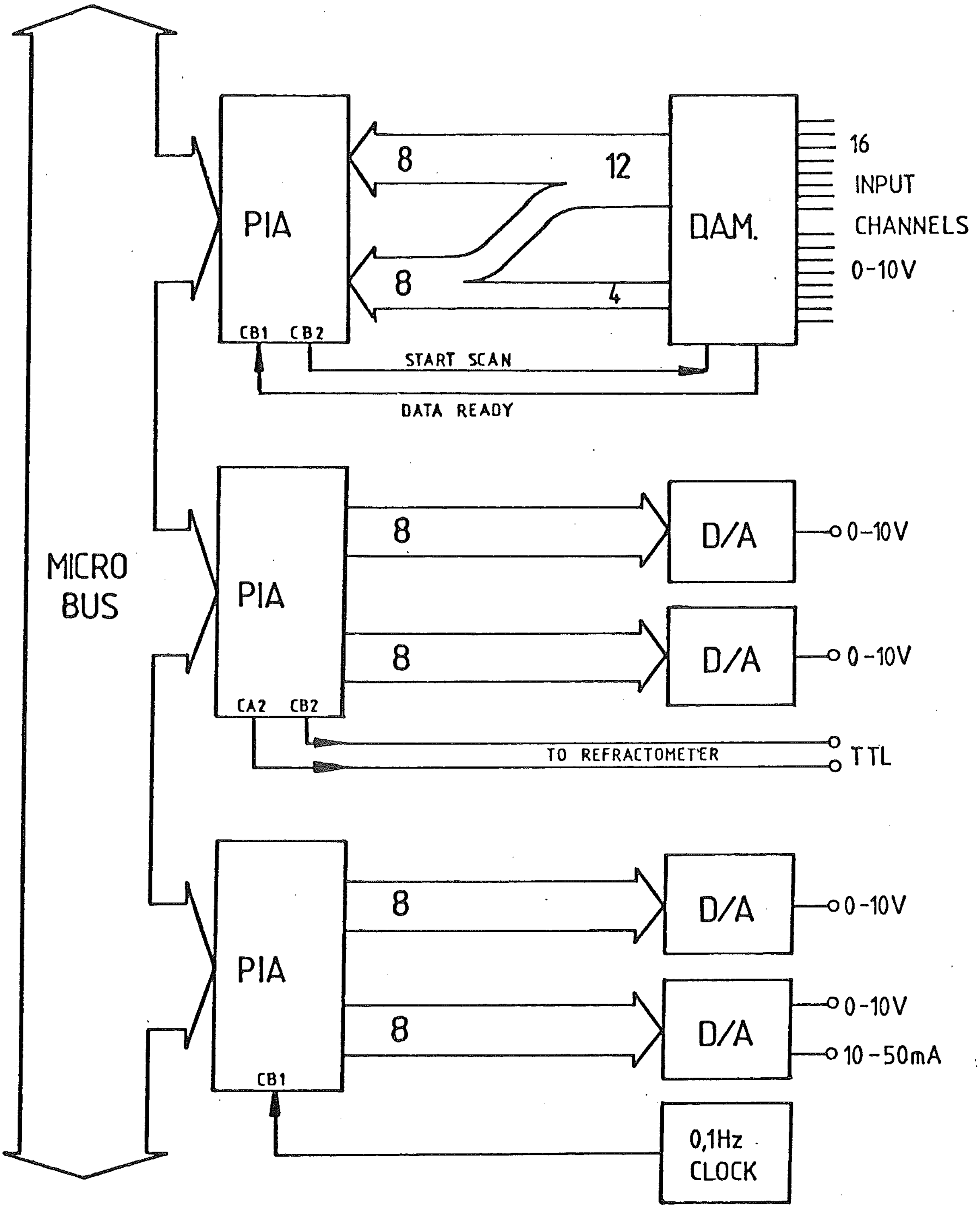


FIGURE 4-1 MICROCOMPUTER/PROCESS INTERFACE

4.4.1 Data Acquisition System

A Burr Brown SDM853 data acquisition module (DAM) was used to monitor the process variables. This unit was a hybrid package containing a sixteen channel multiplexer with control logic, a scaling amplifier, a sample and hold unit, and a twelve bit analog-to-digital converter with control logic.

The DAM for this application was configured to operate in one of the four switch selectable modes described in Table 4-2 by the addition of some discrete logic to the basic module. A circuit diagram of the unit appears in Appendix I.

TABLE 4-2

DATA ACQUISITION SYSTEM

Modes of Operation:

(1) Computer Run - To sample all sixteen channels sequentially once, and stop.

(2) Computer Hold - To sample the current channel once and stop.

(3) Manual Run - To advance to the next channel and sample it on depression of the push-to-convert button.

(4) Manual Hold - To sample the current channel on depression of the push-to-convert button.

Features:

- Input ranges ± 2.5V, ± 5V, ± 10V, 0-10V, 0-5V
- Variable scan frequency from 0 to 10 kHz
- Sixteen single input channels
- Channel roll over: channel 0 follows channel 15
- LED front panel to display the twelve data bits (red), and four channel identification bits (green)
- SCAN and DATA READY TTL logic lines for handshaking operation with a computer.

For control by the microcomputer, the data acquisition system was connected as in figure 4-1. A low to high transition on the SCAN line initiated either a sixteen channel sweep or a single channel sample depending on the operating mode. When the digital-to-analog converter had completed the data conversion on a particular channel, a low to high transition occurred on the DATA READY line to signal the microcomputer to take the available data.

The timing logic of the unit allowed variable sweep rates for interfacing to microprocessors of various speeds. In this case, the M6800 microprocessor was not the limiting factor in the data acquisition process. For convenience, and to allow visual monitoring of performance, the sweep rate was set at 30 Hz giving one channel every 33.3 ms. This slow sweep rate also allowed the microcomputer to operate on the data between samples.

To conform to the interface standard of 0-10V used for all the column instrumentation, the data acquisition system was operated on the 0-10V range, in the COMPUTER RUN mode.

4.4.2 Control Outputs

Control outputs to the process were in two forms:

(1) Analog control signals (0-10V, 10-50 mA) for controlling the speed controllable pumps and for the setpoint to the steam flow loop.

(2) Logic control signals (TTL) for controlling the selection of a sample stream to the on-line refractometer.

The connections to the microcomputer are shown in figure 4-1.

The analog outputs were generated using Motorola MC3408 8 bit digital-to-analog converters. Each converter was connected to one 8 bit parallel port of a PIA. Some signal conditioning was necessary on the digital-to-analog converter outputs since these were current output devices. An operational amplifier was used to allow both scaling and offset adjustment on the converter output in the range 0-15V. In addition a voltage to current converter was also constructed to drive the current-to-air converter in the steam flow loop and allowed remote control of the loop setpoint. A circuit diagram is shown in Appendix I.

The logic control for the refractometer sampling system used two of the output control lines available on a PIA. The solenoid controller described in section 3.8.4 performed a four from two decoding and activated the appropriate solenoid valve. The two control lines used for this purpose were also connected to front panel LED's 1 and 2 of the microcomputer, to provide a visual indication of the current sampling channel.

4.4.3 Timing

Software timers and timed interrupts were generated from a locally constructed 10^{-4} to 10^{5} Hz external clock and one PIA input line.

4.5 MICROCOMPUTER DEVELOPMENT SYSTEM

The PDP-11 minicomputer in the Department of Chemical Engineering was linked to the microcomputer through a serial interface to form a development system. A cross-assembler was written to produce code for the microcomputer using the minicomputer facilities. An interface program was written for the minicomputer to transfer data and code between the computers, and to control the microcomputer for debugging. The development system is shown schematically in figure 4-2.

4.5.1 Development System Hardware

The Department of Chemical Engineering maintained a PDP-11 minicomputer for use in real time process control work, undergraduate design and research work. The features of this machine are summarised in Table 4-3. The PDP-11 is housed in an airconditioned room approximately 150 m from the distillation column and the microcomputer.

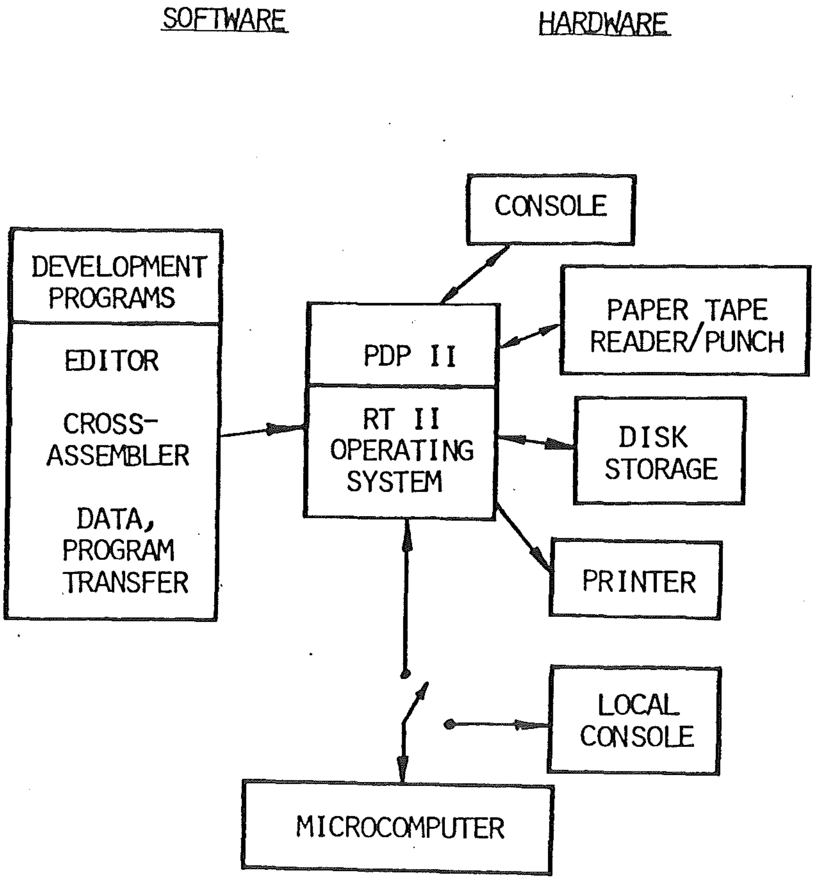


FIGURE 4-2 DEVELOPMENT SYSTEM

TABLE 4-3

MINICOMPUTER FEATURES

PDP 11/15 CPU

20K Core memory, 8K Semiconductor memory

RK05 disk storage 1.2 M words

KE11 extended arithmetic element

Serial Printer (600 baud)

Teletype terminal (110 baud)

Console V.D.U. (2400 baud)

Paper Tape Reader (300 ch/s)

4 x DR11C general purpose interfaces

- 16 bit word in, 16 bit word out
- 2 control and 2 interrrupt lines

KW11-L line clock

Operating System: RT-11 V03B

To make use of the available features of the minicomputer, a serial link to allow communication between the minicomputer and microcomputer was constructed. To simplify this link, and to allow the use of existing software in the microcomputer operating system, the serial I/O port of the microcomputer was used. This interface used a 20mA current loop and was immune to electrical noise, important because of the considerable electromagnetic radiation in the vicinity of the cable which linked the microcomputer to the distant minicomputer.

A hardware interface was constructed using a universal asynchronous receiver/transmitter (UART) to link one of the minicomputer parallel interfaces (DR11C) to the microcomputer serial port via a serial 20mA current loop. A circuit diagram of the interface and a description of its operation is given in Appendix I.

The two output control lines on the DR11C interface were connected to the reset ($\overline{\text{RESET}}$) and non-maskable interrupt ($\overline{\text{NMI}}$) lines of the microcomputer. Software control of these lines in the PDP-11 allowed the minicomputer to control the operation of the microcomputer.

4.5.2 Interface Software

To control the interface described in 4.5.1, a FORTRAN program, MGW, was written to make the PDP-11 appear to the microcomputer as a VDU. This allowed use of the software available in the microcomputer operating system PROTO. All data from the microcomputer was accepted under interrupt by the PDP-11 and directed to the console terminal. The PDP-11 console terminal operated at 2400 baud and this placed an upper limit on the speed of the inter-computer link.

Commands were accepted from the PDP-11 console and decoded; those for the microcomputer were passed down the serial link, and the others were actioned by MGW. The commands available were those listed in the PROTO manual (AMI (1976)) and the following

RE - to generate a $\overline{\text{RESET}}$ command in the microcomputer

NM - to generate a $\overline{\text{NMI}}$ interrupt in the microcomputer

ST - to stop MGW, disable interrupts and clean up the system.

The load command format remained the same as for PROTO, but the PDP-11 responded by asking for a code file name. On receipt of a valid file name, MGW checked that the file existed on the PDP-11 disk. If the file was found, MGW opened the file and then sent the original load command to the microcomputer, which in turn responded with a DC1 character ($11) to indicate 'ready to receive data'. MGW then read the file from disk, block by block, and sent the data to the microcomputer byte by byte down the link. When an end-of-file record was received by the microcomputer, it sent a DC3 character ($13) to the PDP-11 to stop the data transfer. The format of the transmitted file followed the standard Motorola hex formatted record system (Motorola (1975)).

The program MGW was written in RT-11 FORTRAN and used the Digital Equipment RT-11 System Subroutine Library. It was necessary to operate the serial link at a lower baud rate than the PDP-11 console, because system overheads in interrupt processing sometimes caused characters to be lost. This was found to occur only in longer transmissions of more than one hundred bytes from the microcomputer to the PDP-11. MGW was used to load programs into the microcomputer, start them executing and to monitor program performance by way of breakpoints (incorporated in PROTO) and by diagnostics written into the microcomputer program.

A listing of MGW can be found in Appendix III.

4.5.3 Software Development

To produce code for a microcomputer at the machine level is slow and tedious. The next step up from this level is to use an assembler, and results in a very large increase in the rate and ease of production of software. Ideally a further step up to a higher level language such as FORTRAN or PASCAL would produce even greater ease and speed of programming. A control system often requires to operate at the bit or byte level for peripheral interfacing and hence some assembler level routines are required. The original microcomputer configuration using only the EVK board was restricted in the available memory, and consequently the decision was made to program in assembly language. Cross-compilers for high level languages were not readily available in 1975, and hence there was no alternative to assembly language programming. With hindsight, it appears that a higher level language using assembler subroutines where necessary would have simplified the programming of the microcomputer.

A cross-assembler, XASMBL, was written in RT-11 FORTRAN to run on a PDP-11 and produce code for a M6800 microprocessor. XASMBL was written in modular form as a series of subroutines. The input file to the cross-assembler was from the PDP-11 disk, and XASMBL produced two output files - one a load file formatted to the Motorola hex tape format (Motorola (1975)), and the second a list file containing the addresses of the assembled instructions, the generated program code and errors as well as the instruction mneumonics, labels and comments of the source program. The instruction manual for XASMBL is in Appendix V. The cross-assembler has been marketed as a commercial product, and has been used in industry, local government and research situations.

Source programs for input to XASMBL were written using the RT-11 system utility program EDIT. The power of this text editing program obviated the need for some of the more sophisticated assembler features such as local symbols, macros and relocatable code. The mass storage capability of the PDP-11 disk meant that source programs were always readily available for modification and patching, assembly by XASMBL and downloading into the microcomputer by MGW.

4.5.4 Software Debugging

The PDP-11 minicomputer was used to control the microcomputer during debugging. The microcomputer software was monitored using breakpoints and register snapshots; using the software interrupt (SWI) routines contained in the microcomputer operating system PROTO (AMI (1976)). These features allowed checking of program performance, display of memory contents, and restarting of the program. With the original source program stored on disk, it was a simple operation to make small or large changes to the program, then re-assemble and download the modified program for further testing. With the serial communication link operating at 2400 baud, 1K bytes of memory could be transferred in 10.8 seconds.

The control software required for this work was written in modular subroutines, with each subroutine being debugged and checked. The routines were then collected with a main program and tested together. The development up to this stage was carried out in RAM using the development system. After the software was fully checked and debugged, it was transferred into EPROM, and ready for use. The development system simplified these operations and reduced software development time.

4.6 ARITHMETIC PROCESSING UNIT

Floating point arithmetic was necessary in the microcomputer to allow more sophisticated control routines to be implemented. These functions could be provided by software or by dedicated hardware processors. Software routines to perform the basic arithmetic operations in both fixed and floating point formats were available through users groups such as the Motorola Users Group. However, the more complex routines such as trigonometric functions, logarithms and exponentials were not readily obtainable.

The major disadvantages of using software routines were the speed restrictions and memory requirements. A typical 32 bit floating point package such as FPAL from Intel required approximately 2K bytes of memory and executed at the following speeds:

Operation	Time/µs
Addition	700
Subtraction	700
Multiplication	1500
Division	3600

The alternative to using software routines, was to use a microprogrammed hardware processor which is a dedicated microprocessor with a specialised instruction set. A number of such units were available to perform a number of operations, e.g. National MM57109 and the American Micro Devices Am9511. The latter unit was the fastest and most powerful eight bit based arithmetic processor available.

The microcomputer described in previous sections was expanded by the addition of an Am9511 arithmetic processing unit (A.P.U.). This device made available 16 bit integer, 32 bit integer, and 32 bit floating point arithmetic operations as well as 32 bit floating point trignometric and arithmetic functions such as cosine, inverse cosine, natural logarithms, exponentials and power raising. A printed circuit board was designed and built to interface an Am9511 APU to the M6800 microcomputer system. The board also had provision for a 2708 1Kx8 EPROM to be used as a store for a control program to operate the APU.

4.6.1 Interfacing the APU to the M6800 CPU

The Am9511 APU is designed to interface to the Intel 8080 family of microprocessors. A circuit was developed and a printed circuit board constructed to interface the APU to the Motorola M6800 family. A circuit diagram of the interface, and a description of its operation can be found in Appendix I.

4.6.2 Operating the Am9511 APU

The APU was essentially a stack oriented, reverse polish device. All data transfers to and from the chip took place over an 8 bit bidirectional data bus, but internally the device operated a sixteen byte stack organised either as 8 levels of 16 bits or 4 levels of 32 bits, depending on the type of data being used. Further information on the chip and its operation is available in the data sheet and algorithm sheet published by American Micro Devices (AMD (1978)).

To make best use of the stack of the APU and to simplify programming, an interpreter APUDRV was written to handle threaded code and to control the APU. This program resided in the 2708 EPROM on the APU board.

Threaded code is a string of commands and data stored in memory. The commands of APUDRV are listed in Table 4-4. Entry to the interpreter was achieved through a subroutine call JSR APUDRV. All bytes in memory from this location onwards were interpreted and actioned by APUDRV until a command byte UNTHRD was reached. At this point the threaded code was terminated and execution continued from the instruction following the UNTHRD command.

TABLE 4-4

APUDRV COMMANDS

(a) APU OPERATIONS

Arithmetic Operations

16 bit integer	SADD, SSUB, SMUL, SDIV, SMUU
32 bit integer	DADD, DSUB, DMUL, DDIV, DMUU
32 bit floating point	FADD, FSUB, FMUL, FDIV, PWR (= x^Y)

Functions

32 bit floating point	SQRT
	SIN COS TAN
	ASIN ACOS ATAN
	LOG LN EXP

Stack Operations	16 bit integer	32 bit integer	32 bit floating point
Change sign	CHSS	CHSD	CHSF
Swap the top of the stack	XCHS	XCHD	XCHF
Roll the stack up	RUPS	RUPD	RUPF
Push the stack down	PTOS	PTOD	PTOF
Float fixed point	FLTS	FLTD	
Fix floating point	FIXS	FIXD	
Push π onto the stack			PUPI

(b) APUDRV OPERATIONS

APU Stack I/O

Load immediate	IPSHS	IPSHD	IPSHF
Load from memory	PUSHS	PUSHD	PUSHF
Read from APU to memory	POPS	POPD	POPF

Conditional Branches

BRA	BEQ	BLT	BLE	BCS
	BNE	BGE	BGT	BCC

Error Branches

BER - branch on any error

BEN - branch on negative argument

BEZ - branch on zero divisor

BEO - branch on overflow

BEU - branch on underflow

BEA - branch on argument too large

Miscellaneous

NOP - No operation (status word set)

UNTHRD - Indicates the end of threaded code

ENTER - Pushes bytes onto the APU stack from the microcomputer console

PRINT - Displays the contents of the APU stack and the address of the PRINT command on the microcomputer console

LOOP COUNT, START - Defines the end of a repeat loop beginning at location START. COUNT is the address of a downcounter which is decremented each time the loop is executed. When the downcounter reaches zero, the next command following LOOP is executed.

As well as the commands destined for the APU, the interpreter also responded to a number of other commands and special conditions. These actions included reading and writing to and from the APU, handling error conditions, and providing loop and branch operations. APUDRV contained error handling procedures; a general error handler was used to take all errors unless the user specified a different action by one of the branch-on-error commands. Part of the top 1K of RAM was used to store pointers and data relating to error handlers as shown in Table 4-5.

TABLE 4-5

APU RAM REGISTERS

Memory Location	Use
$FFE2, $FFE3	Memory address at which last APU error occurred.
$FFE4	APU status byte at the last APU error
$FFE7, $FFE8	Pointer to the general error handler for APU detected errors
$FFE9, $FFEA	Pointer to the next threaded code instruction to be executed.

The user would set up the address of the error handler in locations $FFE7, $FFE8 before entering the interpreter to ensure that error conditions were correctly handled. By examining locations $FFE2, $FFE3, $FFE4 , the location of any APU error, and the type of error could be determined.

The I/O instructions came in two forms. Immediate mode pushes onto the APU stack took the required data bytes directly from the threaded code immediately following the command byte. Direct mode pushes and pops shifted data to the APU from specified memory addresses, and from the APU to specified memory addresses.

Branches were possible within the threaded code and followed the same format as the M6800 branches except that they were absolute branches and could jump to any label, anywhere in memory within a threaded code section. The branches operated on the condition code bits in the APU status word which were set or cleared according to the value on top of the APU stack.

Two useful commands for program development were the ENTER and PRINT commands. The ENTER command accepted bytes from the microcomputer console and successively pushed these bytes on to the APU stack. PRINT caused a listing of all sixteen bytes of the APU stack on the microcomputer console; the memory address at which the PRINT command occurred was displayed on the console so that the output of more than one PRINT command could be resolved.

The loop command provided for repeated operation of a section of threaded code. Its position defined the end of the loop and its syntax is LOOP COUNT, START where COUNT is the address of an 8 bit downcounter which is decremented on each loop. When the downcounter reaches zero, the next command following LOOP is executed. START is the address of the first instruction of the loop.

Future additions planned for APUDRV include indexed pushes and pops allowing operation on arrays etc., I/O formating routines to convert from standard ASCII formats to the internal binary forms, and the ability to call threaded code subroutines from within threaded code. These expansions are planned and will approximately fill the remaining space in the 1K byte EPROM on the APU board.

4.6.3 Am9511 APU Performance

By using the interpreter APUDRV, certain overheads were incurred in operating the APU. The overheads are listed in Table 4-6.

TABLE 4-6

APUDRV OVERHEADS

Operation	Time Overhead/µs
Load 2 bytes into APU	270
Load 4 bytes into APU	360
Read 2 bytes from APU	230
Read 4 bytes from APU	360
APU command	90

Note that the overhead for an APU operation is in addition to the actual execution time for that particular operation.

Consider a worst case arithmetic operation using the APU where two floating point numbers were loaded into the APU, multiplied together, and the answer replaced in memory. The operation times are listed in Table 4-7.

TABLE 4-7

EXECUTION TIMES FOR AN APU TEST PROGRAM (CPU 1MHz)
(APU 2MHz)

Operation	Time/µs	% of Total Time
Load APU (8 bytes)	720	60
Multiply	170	14
Read from APU (4 bytes)	320	26
	1210	100

For this example, I/O occupied 86% of the total execution time for a 1 MHz CPU. This time could be reduced by using a faster CPU, e.g. a 2 MHz M6800, and marginally reduced by using a 4 MHz APU. Further time savings are possible by using in-line code instead of a threaded code interpreter. However, the use of in-line code would increase the code requirement 3 to 20 times, and lead to longer program development and debugging times. For distillation column control, the speed restrictions resulting from using threaded code were not critical.

As a comparison, a test program using the feedforward column controller described in Chapter eight was coded and executed in FORTRAN and BASIC on the PDP-11 minicomputer described in Table 4-3, and in assembler for execution by the APU. The relative times are given in Table 4-8.

TABLE 4-8

COMPARISON OF EXECUTION TIMES

Language	Machine	Time/ms
FORTRAN	PDP-11	23
BASIC	PDP-11	65
ASSEMBLER	Am9511/M6800	41

A further comparison was made between the speed and memory requirements of the APU and software routines written for an 8 bit microcomputer. Because a floating point arithmetic package was not available for the M6800, an approximately equivalent package produced by Intel for the 8080/8085 family of microprocessors was used in comparison with the APU. The Intel Floating Point Arithmetic Package (FPAL) quoted average times for the four basic arithmetic operations. These were compared with average times for the APU using in-line code and using threaded code with the interpreter APUDRV in Table 4-9.

From Table 4-9 the time advantages for using the APU are marginal when using threaded code for addition and subtraction, but significant for

TABLE 4-9

COMPARISON OF INTEL FPAL WITH APU (APU 2MHz) (CPU 1MHz)

Operation	Average Time/μS FPAL	Average Time/μs APU Threaded Code	Average Time/μs In-line Code
+	700	785	145
-	700	790	150
*	1500	759	120
/	3600	765	125

multiplication and division operations. It should also be noted that the APU times include the time to move the data bytes onto the APU stack and off again. If, in the course of a series operations, data can be held on the APU stack, then the execution time for the APU will be more favourable with respect to the floating point package.

The FPAL package for performing the basic arithmetic operations required 1900 bytes of memory plus 20 bytes of scratchpad memory for an accumulator. The APU unit on the other hand occupied only two bytes of the possible memory space, plus 1K EPROM for the interpreter, APUDRV. To include all the functions available in the APU, it was estimated that 5K bytes of memory would be required based on the routines available through the Intel Users Group.

In conclusion, the Am9511 was a very powerful device for extending the computational capability of an 8 bit microprocessor based system. The addition of an interpreter APUDRV to handle threaded code reduced the throughput capability of the APU as a trade off against programming ease. In this application the Am9511 was found to be extremely useful in extending the possible range of microcomputer control programs to feedforward and adaptive controller strategies.

4.6.4 Generating Threaded Code for the APU Interpreter

A pre-assembler, PREASS, was written to translate the APU mneumonics into the equivalent opcode in a form acceptable to the cross-assembler. PREASS also expanded a set of macros to handle the additional operations described in section 4.6.2. A more detailed description of PREASS, and an example of its use can be found in Appendix V.

4.7 NOMENCLATURE

ACIA	-	asynchronous communications interface adapter
APU	-	arithmetic processing unit
CPU	-	central processing unit
DAM	-	data acquisition module
DMA	-	direct memory access
EPROM	-	erasable, programmable read-only memory
LED	-	light emitting diode
PIA	-	peripheral interface adapter
RAM	-	read/write memory
ROM	-	read only memory
TTL	-	transistor-transistor logic
VDU	-	visual display unit
$	-	prefix indicates a hexadecimal number (base 16).

CHAPTER FIVE

MICROCOMPUTER SOFTWARE

CONTENTS Page

CHAPTER FIVE

MICROCOMPUTER SOFTWARE

5.1 INTRODUCTION

The direction of this work changed during the period of the project due to major changes in technology, especially in the microprocessor field. The original design was for a mini-computer controlled distillation column; this led to a minicomputer/microcomputer configuration, and finally to a stand alone microcomputer system. The changing level of emphasis from the minicomputer to the microcomputer led to a change in direction for the software development. Initial plans were for the microcomputer to be no more than an intelligent peripheral controlled by the minicomputer, whereas in the final design the microcomputer became self-sufficient.

The first step was the replacement of the read-only memory (ROM) based operating system, PROTO, on the EVK300 board with a subset called OPSYS. This avoided some of the incompatability between PROTO and the microcomputer peripherals.

The second step was the creation of a suite of programs, CC68, for the microcomputer to operate the peripherals, and to provide individual loop controllers. These programs were written as subroutines with a main program to sequence them and a common database. An overall flow diagram of CC68 is given in figure 5-1. The individual routines of CC68 are discussed further in the following sections.

5.2 OPSYS - A MICRO OPERATING SYSTEM

PROTO, the ROM based operating system had a number of useful commands that allowed control of the microcomputer from power on (AMI(1976)). However, there were several disadvantages to using PROTO to control the microcomputer:

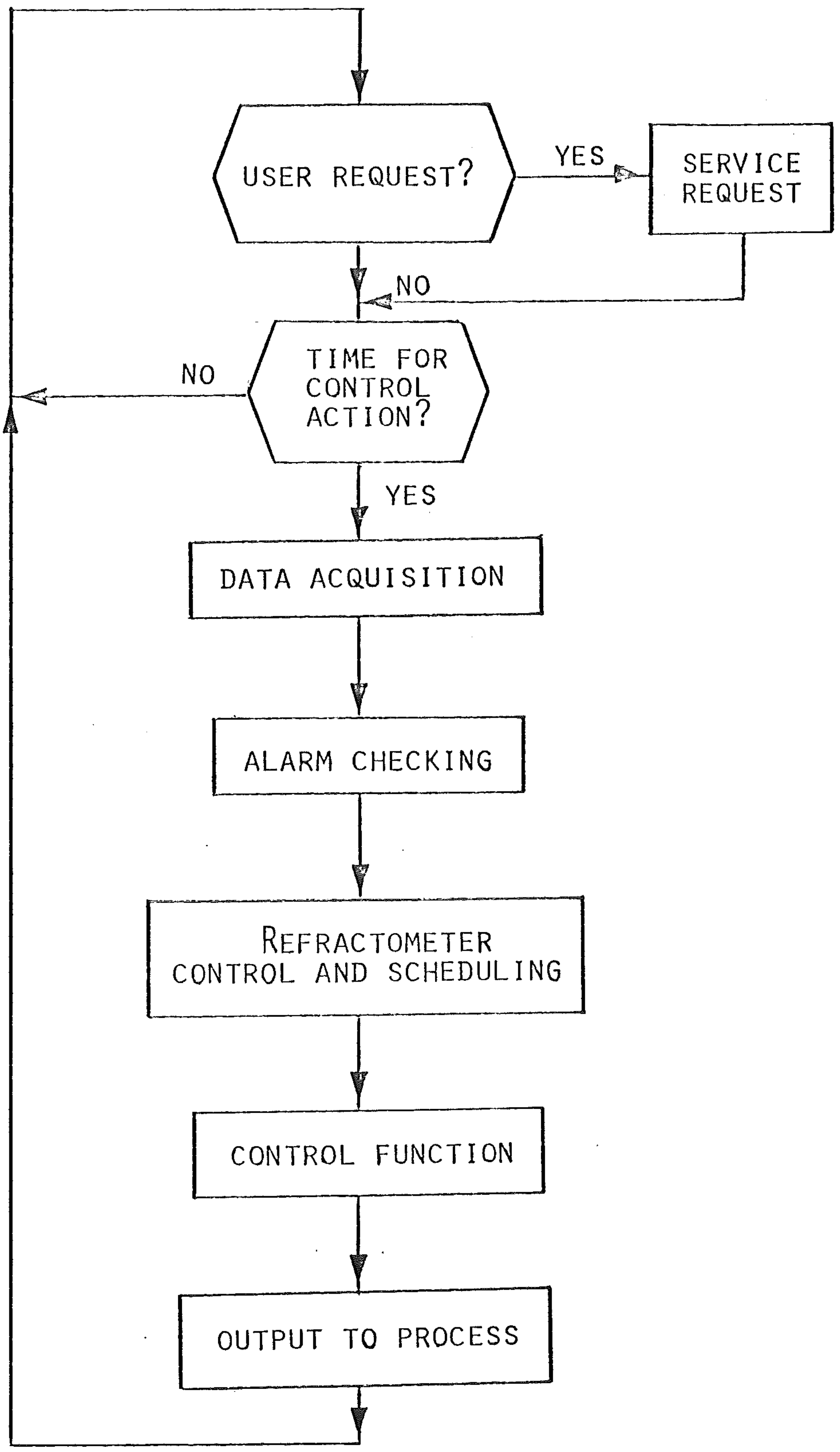


FIGURE 5-1 BASIC CONTROL PROGRAM

(i) On $\overline{\text{RESET}}$ or on a detected error, PROTO initialised two of the peripheral interface adapters (PIA) on the EVK300 for EPROM programming. This created problems for any peripheral that was using those PIAs.

(ii) PROTO started a program by pushing all the registers onto the stack (including the new starting address) and then executing a return-from-interrupt (RTI) instruction. (This sequence allowed PROTO to provide breakpoints). This method of starting proved unreliable and often a $\overline{\text{RESET}}$ was required.

OPSYS was written as a subset of and an adjunct to PROTO with the features summarised in Table 5-1. Other differences between OPSYS and PROTO were

(i) On $\overline{\text{RESET}}$, OPSYS configured the PIA's to suit the micro-computer peripherals.

(ii) OPSYS could be entered via an $\overline{\text{NMI}}$ interrupt without reinitialising the PIA's or altering any variables.

(iii) OPSYS started a program by executing a JMP instruction.

TABLE 5-1

OPSYS FEATURES

C - to configure PIAs - PIA #1, #3 for output via D/A
- PIA #2 for input from data acquisition module

G < addr > - to start execution at < addr >

J - to jump to PROTO entry point

L - to load. This command was as for PROTO but no parameters were accepted

Z - to zero all RAM locations from $0 up.

OPSYS and PROTO both share the 512 bytes of RAM at the top of memory for data and stack space.

OPSYS was written in assembler using the cross assembler described in Chapter 4, burnt into a M6834 EPROM and installed on the EVK 300 board with a start address of $E600 ($ prefix indicates a hexadecimal number). The EVK300 restart switch register was changed to route resets to OPSYS.

5.3 DATA BASE

The programming of the CC68 routines was arranged so that all program code resided in EPROM , and the data base resided in RAM . To make best use of the direct memory addressing mode of the M6800 microprocessor (Motorola (1975)) the data base was organised to start at $0 and extend up to $1FF. All the data associated with the CC68 routines was located in the area $0 to $C0. Expansion of the data base was possible after the last entry. Those sections of the data base as necessary were included with the individual routines described below for assembly via XASMBL . A listing of the CC68 data base is given in Appendix IV.

5.4 TIMING FUNCTIONS AND THE INTERRUPT ROUTINE

Timing registers were implemented using an $\overline{\text{IRQ}}$ interrupt routine driven by an external clock (section 4.4.3). The operator interface also operated under the same $\overline{\text{IRQ}}$ interrupt line, so the peripheral devices were polled to determine which required attention. This situation was further complicated by the fact that if an $\overline{\text{NMI}}$ interrupt occurred while an SWI interrupt was in progress, then neither the SWI or the $\overline{\text{NMI}}$ interrupt was executed but an $\overline{\text{IRQ}}$ interrupt was taken instead (Motorola (1975)). This possibility was also polled. A flow chart of the $\overline{\text{IRQ}}$ interrupt routine, TIMER, is given in figure 5-2 and a listing of the routine is given in Appendix IV.

The timer registers were five single byte RAM locations, each of which was incremented by one on each interrupt of the external clock.

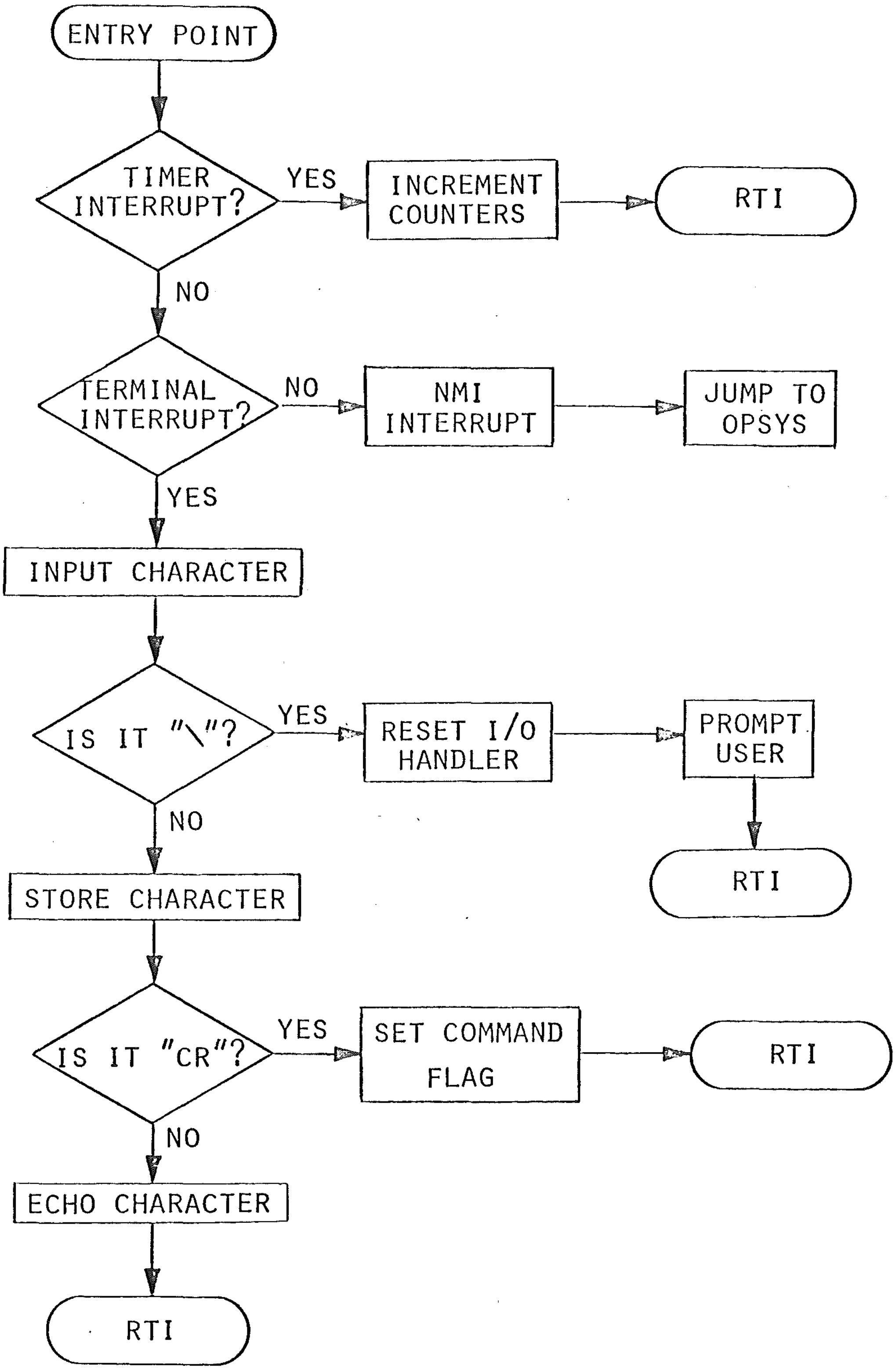


FIGURE 5-2 INTERRUPT HANDLER

The user program could clear or modify these registers in any way. Any $\overline{\text{NMI}}$ interrupts detected in this routine were passed to OPSYS . This allowed entry into the microcomputer operating system without going through the $\overline{\text{RESET}}$ procedure. The operator interface terminal (VDU) control was also performed in the $\overline{\text{IRQ}}$ routine. This involved accepting characters from the terminal and echoing valid ones. A backslash ($5C) was defined to be an escape character, and caused the current line of input to be rejected and a user prompt for a new line of input made. A carriage return ($0D) completed the line of input and set a flag to indicate to the main program that there was a completed line of input available.

5.5 OPERATOR INTERFACE

The operator interface was a minimal set of commands for altering and displaying parts of the database. It consisted of two parts; the operator I/O which was under $\overline{\text{IRQ}}$ interrupt (see section 5.4) and the operational section which actioned the operator commands. On receiving a valid line of input, the main program of CC68 branched to the routine USERC and decoded the input line. The available commands are listed in Table 5-2. The command consisted of a single alphabetic character (with or without modifiers) followed by a carriage return. The input line was compared with a table of available commands, and a branch taken to the appropriate routine. If the command was undefined, the routine returned to the main program. A listing of the routine USERC is given in Appendix IV.

TABLE 5-2

OPERATOR COMMANDS

C - To print the controller parameters as

< double byte > < double byte > < single byte >

K_1 K_2 PV Index

for each of the four controllers in order, one per line. After the four controllers, a single byte was printed showing the current controller division factor. (See Section 5.9 for details).

D - To print out the data from the data acquisition system as two rows of eight double bytes, in sequence from channel 0 to 15 . Each double byte contained a four bit channel identifier and 12 bits of data, e.g.

$AB23 ⇒ Channel 10 ($A) had a value of 2851 ($B23).

P - To print out four single bytes representing system parameters as follows:

byte 1 = system status (see Section 5.11)

byte 2 = data acquisition error flag (0 = no error)

byte 3 = current refractometer channel

byte 4 = sampling time factor (see Section 5.11).

R - To print out the data collected from the online refractometer as four double bytes. The most recently acquired data had the upper nibble set to $8. The refractometer channels were in the following sequence:

(1) distillate composition

(2) feed composition

(3) bottoms composition

(4) reference composition.

S - To print out the setpoints for the single loop controllers described in 5.9 as four double bytes in the sequence

(1) loop 1 (controls reflux flow)

(2) loop 2 (controls bottoms flow)

(3) loop 3 (controls distillate flow)

(4) loop 4 (controls steam flow).

V - To print out the current valve positions as four single bytes in the order

(1) reflux flow

(2) bottoms flow

(3) distillate flow

(4) steam flow.

I < addr > , < byte 1 > { , < byte 2 > } ...

to allow the entry of a sequence of bytes (byte 1, byte 2 etc.) into memory starting at location < addr > .

O < addr > , < count >

to output to the terminal < count > single bytes starting at location < addr > . The starting location was also listed as a double byte.

Note: (i) All I/O was done in hexadecimal base integers.

(ii) No conversions were performed - all data was displayed or entered in memory image form, i.e. data acquisition system data was in the form 0 - $FFF (0 - 4095).

(iii) Valve position ≡ pump control.

5.6 DATA ACQUISITION

Data was acquired from the process through the system described in 4.4.1 under the software control of a subroutine READAD. On entry to the routine, a timer register and a channel counter were cleared, and a pointer set up to the data base. The data acquisition system was then triggered by lowering, then raising the SCAN line. When data became available, the DATA READY line was raised high and this was shown in bit 7 of the PIA status register. READAD polled the status register, waiting for the DATA READY signal. When this occurred, READAD accepted the data from the two PIA input registers, and checked that the correct sequence channel had been received. If the channel was not the expected one, an error was logged and the routine continued; otherwise the data was stored as received in the data base. When sixteen channels had been received, the routine returned to the calling program.

If the timer register cleared on entry to the routine incremented before all sixteen channels had been read, READAD aborted with the error flag set to $FF . (Note that the error flag otherwise contains the number of channel read errors.) The errors detected and flagged by READAD were actioned by another routine CHECK which implemented alarm checking.

A flow diagram of READAD is given in figure 5-3, and a listing of the routine is given in Appendix IV. Allocation of the 16 channels is shown in Table 5-3.

TABLE 5-3

DATA ACQUISITION CHANNEL ALLOCATION

Channel Number	Process Variable
0	reflux accumulator level
1	reboiler level
2	temperature tray 1
3	" " 2
4	" " 3
5	" " 4
6	" " 5
7	" " 6
8	" " 7
9	" " 8
10	" reflux accumulator
11	" feed
12	" ambient
13	column pressure drop
14	refractometer composition
15	refractometer temperature

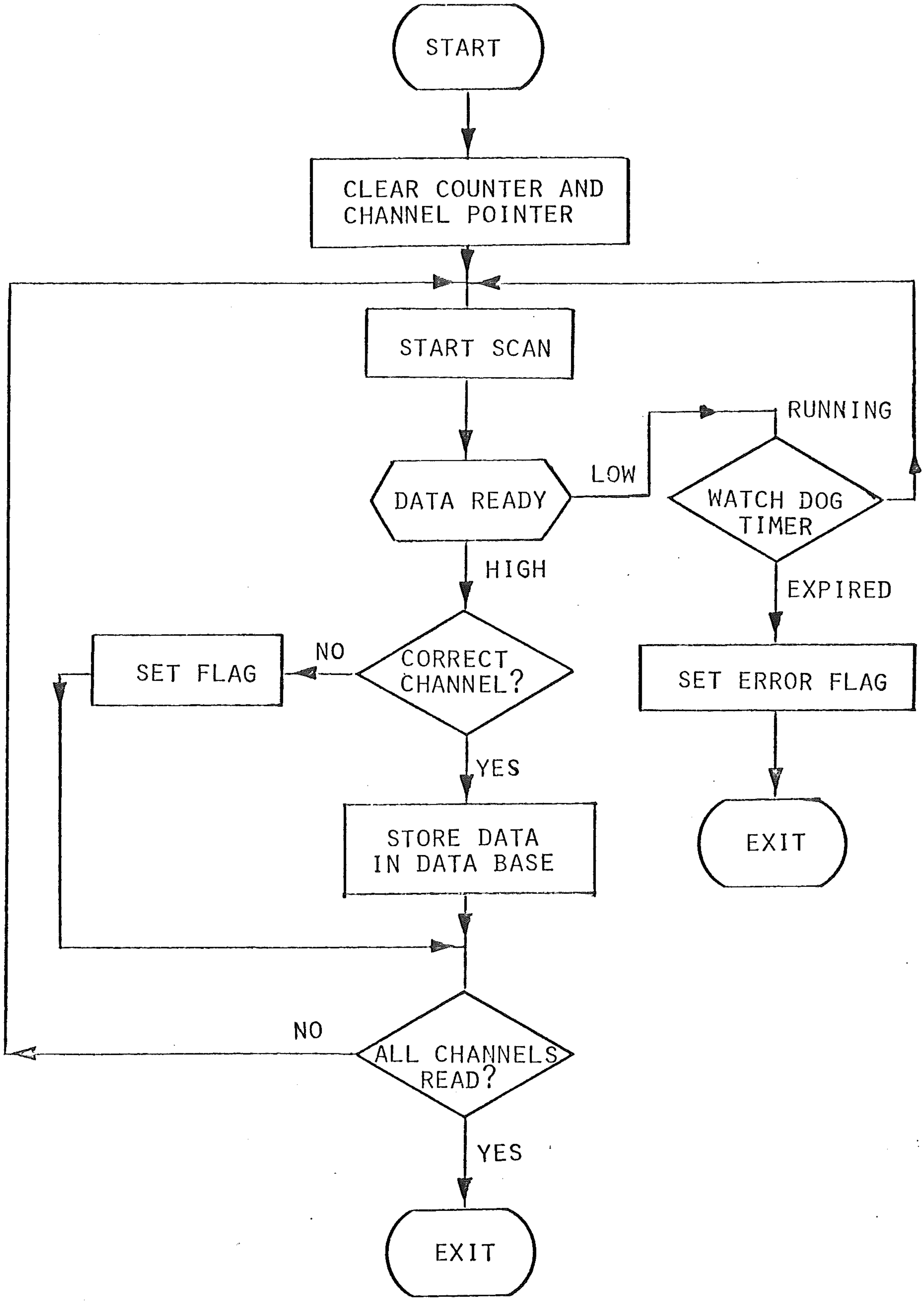


FIGURE 5-3 DATA AQUISITION PROGRAM

5.7 CONTROL OUTPUTS

All analog outputs were voltages in the range 0 - 10V from the digital - to analog converters described in section 4.4.2; the pump control signals were passed directly to the motor speed controllers, and the output to the steam flow control loop was converted to a current (10-50 mA). Interfacing to the digital-to-analog converters is straightforward; the desired output (0-255) was loaded into the PIA output register, and within 250ns the corresponding voltage appeared at the process control elements.

The speed controllable motors driving the control pumps can lose synchronisation and stop rotating if the control voltage changes at a rate greater than $0.2Vs^{-1}$; some hardware protection was included in the speed controllers to ensure that this condition was met. As additional protection, the control outputs were set by a slewing algorithm in the subroutine VALVES. (In this context valves are considered flow controllers and equivalent to pumps). This subroutine slewed each control output at rate selected under software control. The maximum rate of change was stored in memory as the largest possible change in the output byte permissible in a period of 125 ms. For an input of 0-10V corresponding to 255 discrete steps then $0.2Vs^{-1}$ corresponded to a maximum change of 5 steps per second or approximately 1 step per 125ms. A flow diagram of VALVES appears in figure 5-4 and a listing of the subroutine in Appendix IV. Allocation of the control outputs is shown in Table 5-4.

TABLE 5-4

CONTROL OUTPUT ALLOCATION

Channel Number	Connected to
0	reflux pump
1	bottoms pump
2	distillate pump
3	steam flow loop

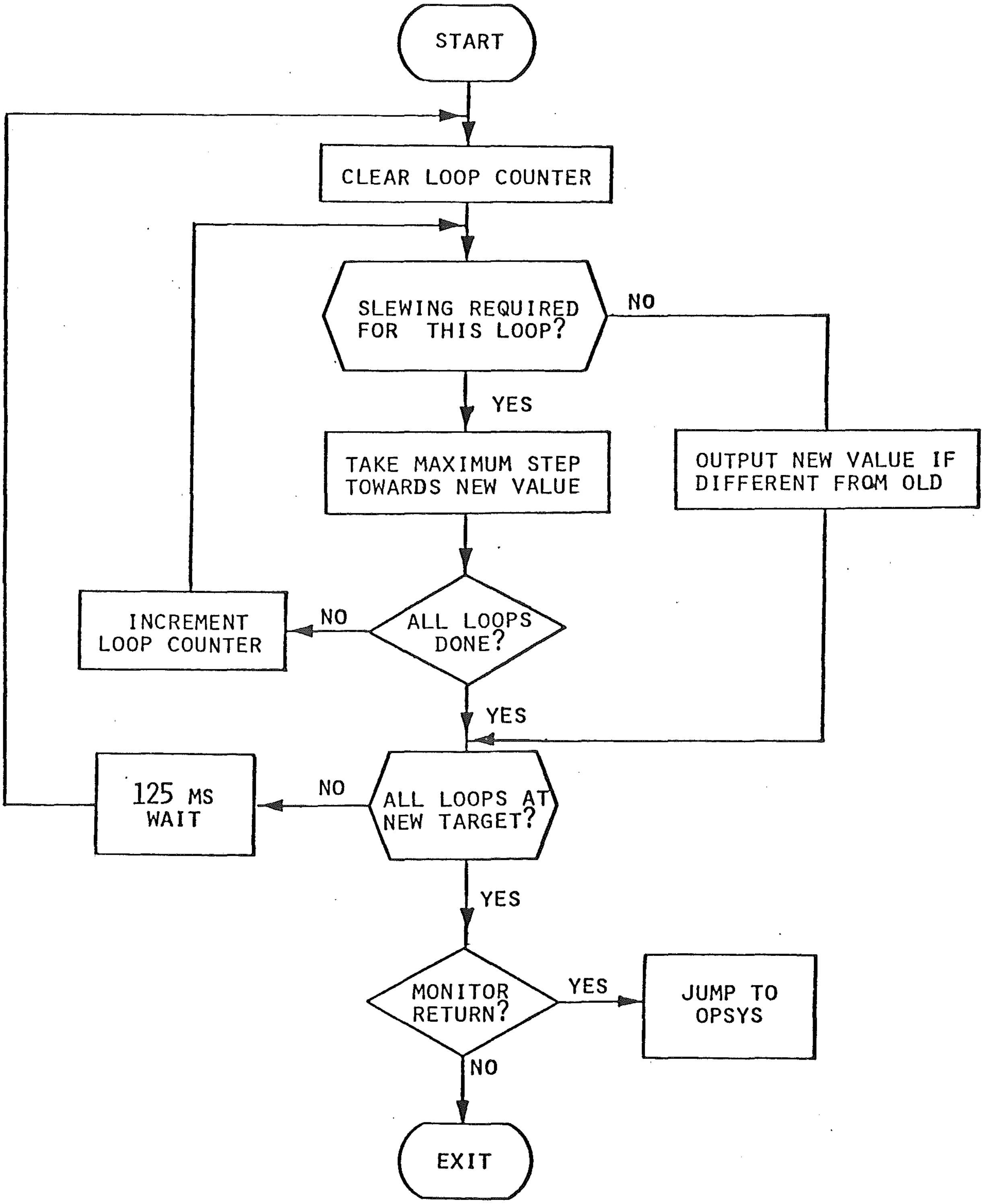


FIGURE 5-4 CONTROL OUTPUT PROGRAM

VALVES could also be used to slew the control outputs under manual control through the operating system, OPSYS. The new required outputs were set at the memory locations $3C to $3F, corresponding to loop 1 to loop 4, and the routine VALVES entered by the command G $E561. The interrupt mask was used to detect whether the control program was running (mask off = program running). This determined whether the routine returned to the calling program with an RTS instruction, or to OPSYS with a JMP (Under OPSYS, the interrupt mask is set).

5.8 REFRACTOMETER CONTROL

Control and scheduling of the OKOMETER online refractometer described in section 3.8.4 was performed by the subroutine, REFCTL. Sequencing of the refractometer channels was implemented through a twelve byte ring buffer; each byte in the buffer contained a code as shown in Table 5-5. A pointer to the ring buffer was stored in the data base, and a timer register was used to ensure that the refractometer settling time met the requirements of the sampling system dynamics. The sampling time was software selectable and was normally set to give 60s between channel changes. When a channel change was required the appropriate code byte from the ring buffer was decoded, and two output control lines of a PIA set by manipulating bits in the control and status register. When a particular channel was deemed to have settled for the selected time interval, the data for the refractometer was taken from the data acquisition system area in the data base, and the channel number stripped off and replaced by a $8 nibble. This indicated that this channel was the most recently sampled one. At the same time, the $8 nibble appearing on the previous channel sampled was cleared to $0.

A flow diagram of REFCTL is shown in figure 5-5 and a listing of the subroutine appears in Appendix IV.

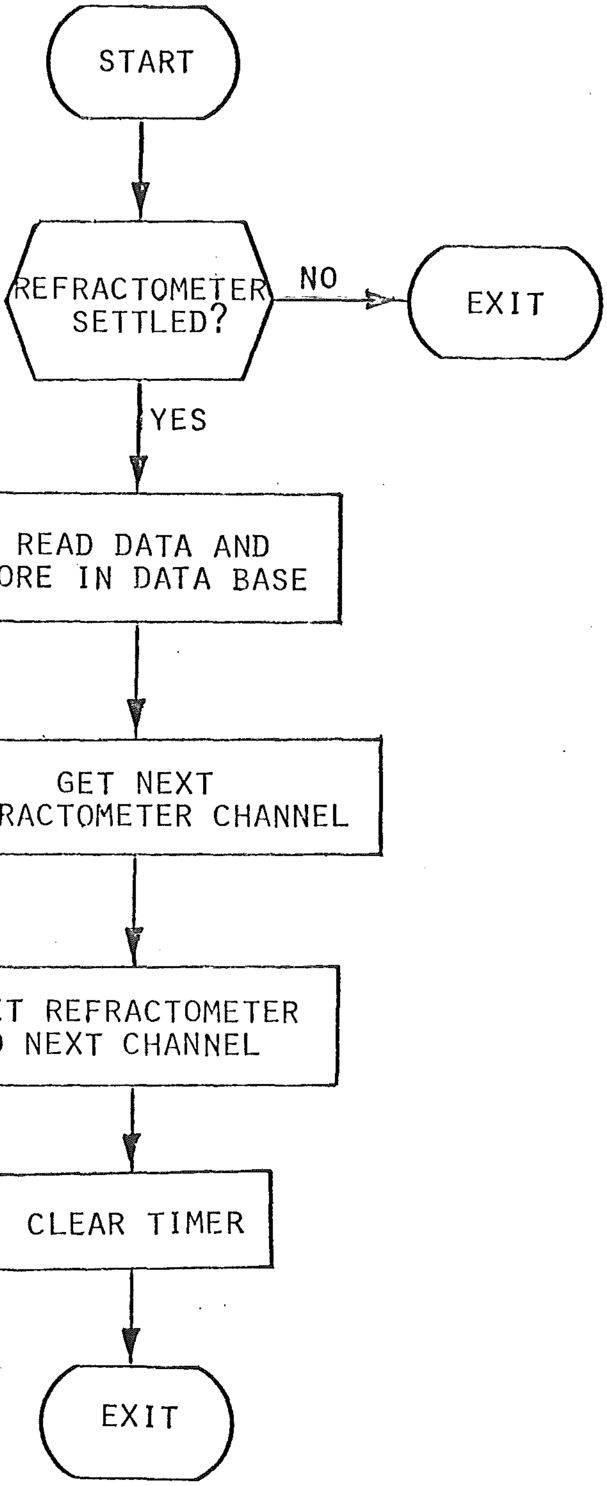

START
REFRACTOMETER SETTLED?
NO
EXIT
YES
READ DATA AND STORE IN DATA BASE
GET NEXT REFRACTOMETER CHANNEL
SET REFRACTOMETER TO NEXT CHANNEL
CLEAR TIMER
EXIT

FIGURE 5-5 ONLINE REFRACTOMETER CONTROL PROGRAM

TABLE 5-5

REFRACTOMETER SAMPLING CODE

Code Byte	Sampled Stream
0	distillate
1	feed
2	bottoms
3	reference

5.9 SINGLE LOOP CONTROLLERS

Subroutine CONTRL provided four digital controllers, each controller associated with one control element as given in Table 5-4. The controllers were sampled data, incremental, proportional-integral types computed using 16 bit integer arithmetic. The arithmetic operations were originally performed using software because the arithmetic processing unit was not available.

The process variable for each controller was selectable by means of single byte offset pointer to the data base. This allowed any element of the data base in the first 256 bytes of memory to be used as the process variable for any of the four loops and added flexibility to the configuration of the control system. The process variables were not scaled, being integers in the range 0- 4095 as converted by the data acquisition system. The setpoints were selected and entered into the controllers on the same basis (0- 4095).

The control function was evaluated as follows:

$$\Delta V = (K_1 E_1 + K_2 E_2)/2^{DIVN}$$

ΔV = change in valve position for the current sample time

where E_1 = $SP - PV_1$ (present error)

E_2 = $SP - PV_2$ (past error)

K_1, K_2 = controller coefficients

DIVN = controller division factor.

The use of valve position was taken to be equivalent to pump speed for the distillation column control scheme. This controller could be related to the industry standard analog controller: for a PI velocity type controller (Smith (1972))

$$\Delta V = K_c(E_1 - E_2 + T\,E_1/T_I)$$

$$= K_c\,E_1\,(1 + T/T_I) - K_c\,E_2$$

K_c = analog controller gain

where T = sampling interval

T_I = analog controller integral time

Comparing these controllers:

$$K_1 = K_c(1 + T/T_I)$$

$$K_2 = -K_c$$

and hence the controller parameters for the CONTRL controllers could be related to the standard controller gain (K_c) and integral time (T_I). The inclusion of the division factor 2^{DIVN} allowed K_1, K_2 values less than one to be specified, and still use integer based arithmetic. The division factor was the last operation performed to retain significance through the calculation of ΔV.

The new valve position (equivalent to pump speed) was calculated as follows:

$$V_{NEW} = V_{OLD} + \Delta V$$

$$V = V_{NEW}/256 + 128$$

where V_{OLD} = V_{NEW} from last calculated control action

V = 8 bit byte output to D/A converter.

The division by 256 was necessary to map the calculated valve positions on to the available valve positions. This can be seen by considering the range of the calculated value of V_{NEW} and the available range for the output byte V.

V_{NEW} = 16 bit integer (range $-$ 32768 to + 32767)

V = 8 bit byte (range $\simeq$ 0 - 255)

$$\therefore \text{Scaling factor} = 2 * 32768/256 = 256$$

The shifting of V by 128 was necessary because of the change from signed binary arithmetic in the controller calculations to the unsigned binary output byte. Dividing V_{NEW} by 256 produced a valve position in the range - 128 to 127 and adding 128 corrected this to 0 to 255 which was the desired interval for the output devices.

The scaling and mapping of ranges in the controllers introduced another gain into the controllers as shown in figure 5-6. The input range is 0-4095 and this was mapped to the output range of 0 - 255 giving a controller scaling factor of $\frac{255}{4095} = 0.0623$. The controller parameters were corrected for this factor. The division factor and the scaling factor mean that the controllers can use fractional gains and integral times. This led to the following method of controller setting; suppose the desired controller settings are

$$K_c = 3.0$$
$$T_I = 1 \text{ min}$$
$$T = 0.1 \text{ min}$$

then using the relationships previously defined

$$K_1 = \frac{3(1 + 0.1/1)}{0.0623} = 52$$

$$K_2 = 3/.0623 = -48$$

These settings were computed using floating point arithmetic and then truncated to 16 bit signed integers.

A flow diagram of CONTRL is shown in figure 5-7 and a listing is given in Appendix IV.

5.10 ALARM CHECKING

Subroutine CHECK performed alarm checking on the data acquisition system and on selected process variables. The former was checked using

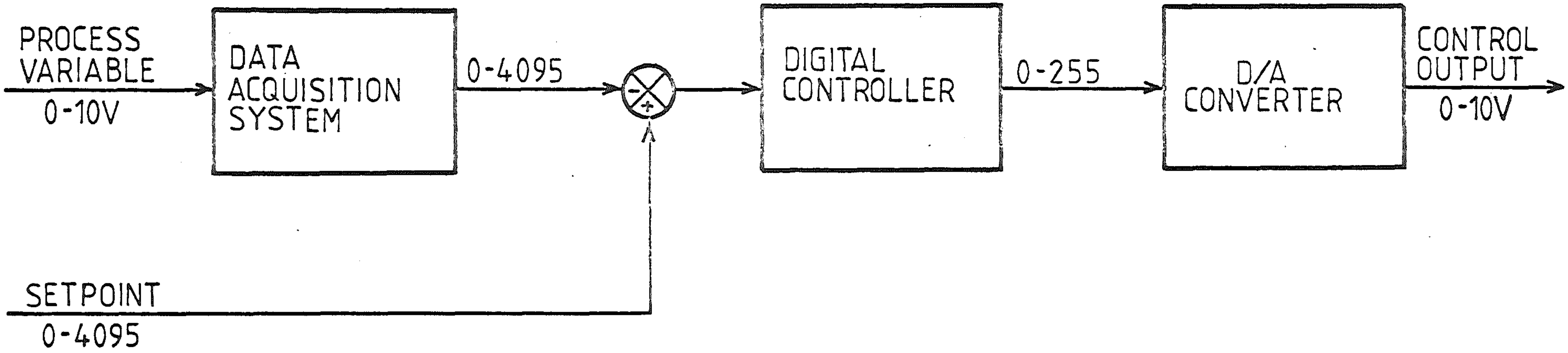


FIGURE 5-6 SINGLE LOOP CONTROLLER

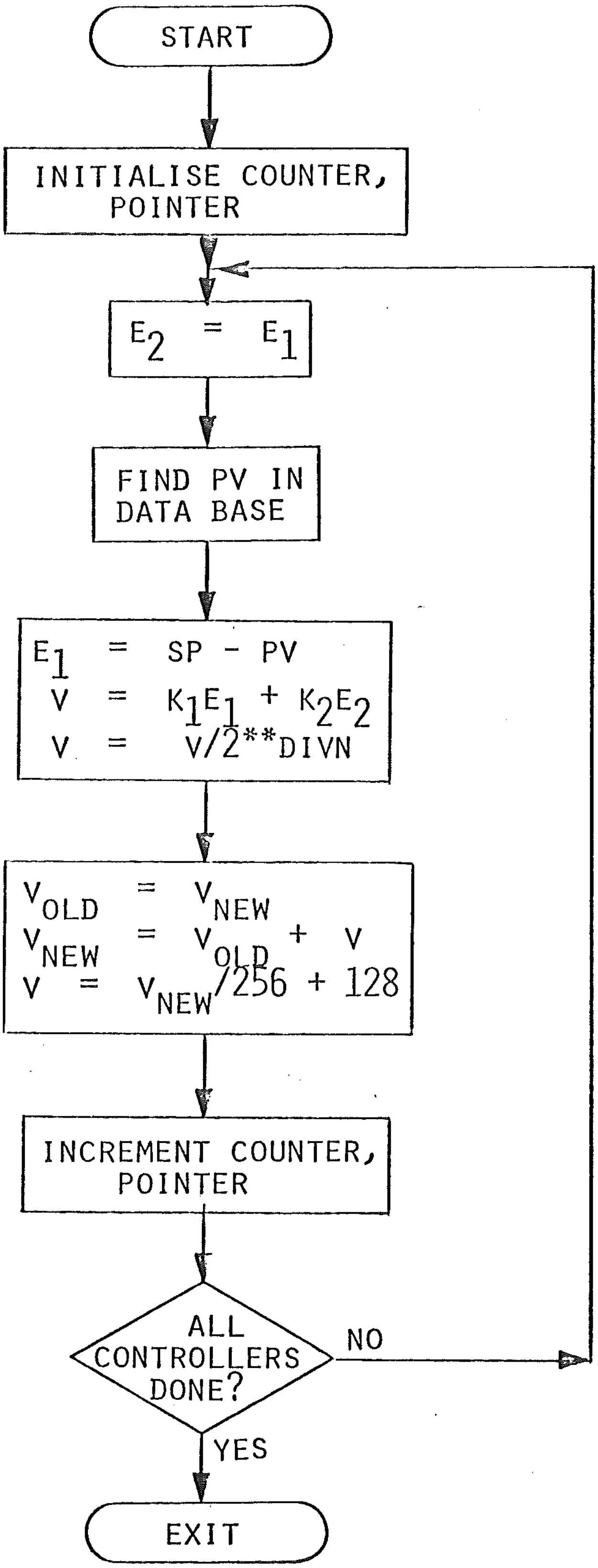


FIGURE 5-7 CONTROLLER PROGRAM

a flag for channel errors or hardware failure while the latter were compared against upper and lower limits set in the data base. On detecting any alarm condition, the terminal bell or buzzer was sounded, a message logged on the terminal as ? ALARM X ? (where X = 0, 1, 2, 3 indicated which alarm was activated), and the terminal bell sounded again. Any action required to correct a specific alarm condition was taken by the operator. Provision was made for the microcomputer to have control of the column power supply to the pumps and steam flow loop by way of the TTL logic signal in the safety interlock system shown in Appendix I. This line could be connected to a PIA control line to allow the microcomputer to shut down the distillation column in an emergency, but for this work, a manual switch was available to the operator to perform this function.

A flow diagram of CHECK showing the alarm codes is given in figure 5-8 and a listing of the subroutine is given in Appendix IV.

5.11 MAIN PROGRAM

The main program served to sequence the subroutines described in the previous sections together into an operational unit. This involved initialising peripheral devices, memory locations, enabling $\overline{\text{IRQ}}$ interrupts, and scheduling of the control routines. Sampling time was determined from one of the timer registers compared against an entry in the data base. While waiting for the correct time to implement control action, the program also checked whether an operator command was pending; if it was, this command was actioned. By interleaving the operator interface and the control system in this way, neither suffered significantly because of the other; operator requests in general were only run in the free time between control action. The variable SYSTAT was used to indicate which section of the control system was active at any time. This feature was included to allow program hangups to be traced quickly.

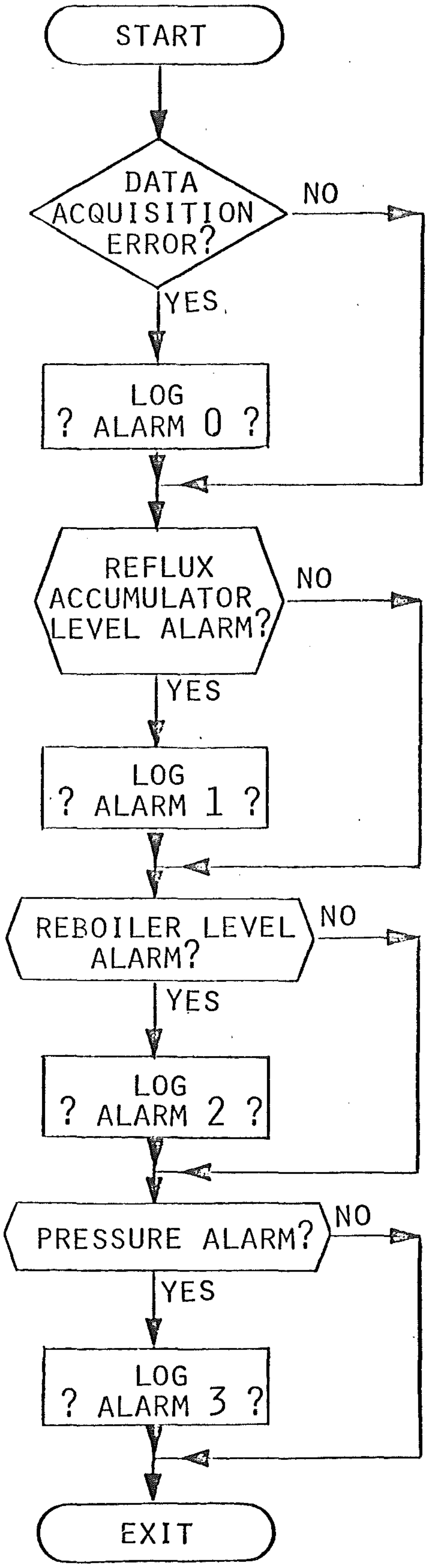


FIGURE 5-8 ALARM CHECKING

The description of the CC68 suite of programs has been restricted to describing the stand alone mode of operation, i.e. the microcomputer plus terminal controlling the distillation column. Provision was made for expansion of the basic system described here by the addition of further routines and/or the addition of PDP-11 supervisory control. The main program executed a subroutine call to memory location $200. For base level operation, this location contained a return instruction, RTS; for expansion either additional subroutines could be placed in memory starting at $200, or this location could contain a jump instruction to another subroutine.

A flow diagram of the main program is given in figure 5-9 and a listing appears in Appendix IV.

It should be noted that the CC68 system was written and developed before the arithmetic processing unit was available, and hence it could have been improved in terms of computational time and storage requirements. The operator interface was primitive and required a working knowledge of the system in order to use it correctly. With the expansion of the available EPROM memory from the original 2K bytes on the EVK300 board to 10K bytes total, it was possible to improve the operator interface. In fact if floating point arithmetic was to be used in the control scheme, it would have been essential to improve the interface and to adopt some standard for all variables such as 0 - 100%. To write such an interface was beyond the aims of this work, but it should be done in the future, and would be a recommended step. Alternatively, if the microcomputer ran in conjunction with the PDP-11 minicomputer, the latter could be programmed to interface to the operator interface in the microcomputer and provide a more sophisticated system.

5.12 CC68 INSTALLATION AND USE

The program, CC68, was installed in three M6834, 512 byte erasable programmable read-only memory chips on the EVK300 board. The disposition

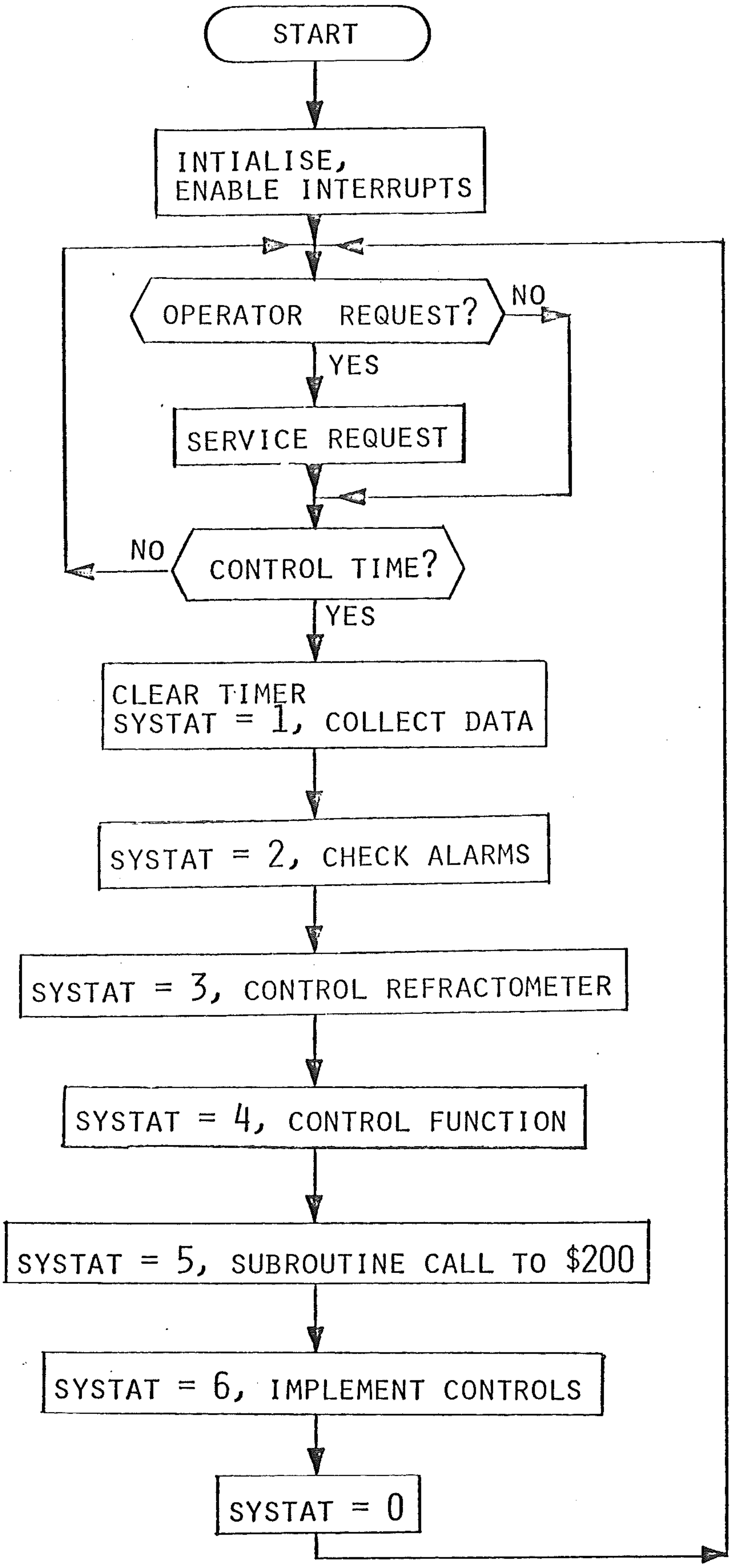


FIGURE 5-9 CC68 MAIN PROGRAM

of the routines was as given in Table 5-6.

TABLE 5-6

MEMORY MAP OF CC68

Address Range	Memory Type	Routines
\$0 - \$CO	RAM	Data Base
\$E000 - \$E1FF	EPROM	MAIN, TIMER, USERC
\$E200 - \$E3FF	EPROM	READAD, CHECK, REFCTL
\$E400 - \$E5FF	EPROM	CONTRL, MULTI6, VALVES
\$E600 - \$E7FF	EPROM	OPSYS

Operation of the system required

(i) on power on, to zero all memory locations;

(ii) to set up the data base parameters as required, e.g. controllers, setpoints, alarm limits, refractometer sequence, etc.

(iii) to start the system with G E000 command.

Thereafter the data base could be accessed or manipulated using the commands available through the operator interface (section 5.5). This system was used during commissioning of the distillation column and for tests using single loop controllers to maintain column product specifications. The system logged up 350 hours of operation with no hardware or software faults.

5.13 COST COMPARISON OF MICROCOMPUTER CONTROL SYSTEM WITH AN EQUIVALENT ANALOG SYSTEM

The control system described in chapters four and five was costed against an equivalent system using conventional analog PID controllers. The same instrumentation and control elements were used in both analyses. All costs are in \$NZ (1979).

(i) Microcomputer System

(a) Software Costs - only the essential sections of the software described in this chapter were included (alarm checking, refractometer control, and the mini operating system were ignored).

Code $\equiv$ 593 lines of assembler.

Allowing 15 lines of debugged documented assembler code per day (Schindler (1979)), and costing programmer time at $96 per day, the estimated cost of the code was $3792.

(b) Hardware Costs - based on retail N.Z. prices including sales tax.

EVK300 microcomputer (assembled and tested)	$1400
Terminal (VDU)	$1700
Data Acquisition System (installed)	$1200
D/A converter Outputs	$200
	$4500

TOTAL COST = $8292

(ii) Analog System

(a)	Electronic PID controllers, 4 @ $2000	$8000
(b)	Interfacing to instrument and control elements	$200

TOTAL COST = $8200

The analog system was found to be less than $100 cheaper than the microcomputer based system. The latter was costed on a one off basis using retail component prices. If the development costs could be spread over several systems, and the hardware purchased in bulk direct from the manufacturer, the microcomputer system would undercut the analog system in price, e.g. on a run of 10 units using the same hardware prices the microcomputer system cost would drop to $4879.

The microcomputer based control system has been shown to be cost comparable with a conventional analog control system. The advantages of the microcomputer system were its flexibility in allowing modification of the control strategy, the addition of new features such as alarm checking, and the centralisation of control with an operator interface.

5.14 NOMENCLATURE

Hardware:		PIA	peripheral interface adapter
		RAM	read/write memory
		ROM	read only memory
Software:		PROTO	operating system on EVK300 board
		OPSYS	improved operating system
		CC68	column control software including
	(i)	MAIN	main program
	(ii)	TIMER	timer registers, terminal I/O
	(iii)	USERC	operator interface
	(iv)	READAD	data acquisition
	(v)	CHECK	alarm checking
	(vi)	REFCTL	refractometer control
	(vii)	CONTRL	controller calculation
	(viii)	MULTI6	signed 16 bit integer multiplication
	(ix)	VALVES	control output

$ indicates a number to base 16.

CHAPTER SIX

STEADY STATE BINARY DISTILLATION COLUMN MODEL

CONTENTS

CHAPTER SIX

STEADY STATE BINARY DISTILLATION COLUMN MODEL

6.1 INTRODUCTION

Distillation columns have been modelled many times for many reasons. The modelling objectives have generally been in two categories:

(i) The design problem where the number of stages required and the column operating conditions are to be determined from specifications on the feed and product streams. A number of methods based on the analyses of McCabe and Thiele (1925), Sorel (Robinson and Gilliland (1950) and Lewis and Matheson (1932)), involving plate to plate calculations have been used. There have been several attempts to find analytical solutions to the design problem (Martin (1963), Singh (1966), Sarkarny et al (1970)), but these methods require a number of simplifying assumptions such as equimolal overflow. There have also been attempts to correlate plate requirements with other column parameters such as reflux ratio (Gilliland (1940), Mason (1959), Brown et al (1939), Erbar et al (1961)).

(ii) The rating problem where the column configuration is fixed along with the feed conditions and the control variables (reflux ratio, boilup rate). It is then required to compute the product flows and compositions. This problem has been approached by formulating the mass and energy balances around the column trays, and then solving the resulting simultaneous equations. Most of the work in this area has been directed toward multicomponent distillation problems, but the same approaches can be used for binary systems (Thiele and Geddes (1933), Amundson and Pontinen (1958), Holland (1963), Wang and Henke (1966)).

The requirement of the steady state model for this work was the prediction of the column performance for a given set of operating conditions leading to the design and simulation of a feedforward control system.

6.2 MODEL DEVELOPMENT

The approach chosen was that of Wang and Henke (1966) adapted to a binary system. The model was based on the distillation column described in Chapter three, and selected to cover all the possible operating conditions of that column. The general layout of the column is shown in figure 6-1. The model assumptions were:

(i) adiabatic operation (no heat losses);

(ii) total condenser producing subcooled reflux;

iii) thermosiphon reboiler (an equivalent tray);

(iv) liquid feed (at or below its bubble point);

(v) specified column variables

feed rate	F	feed temperature	T_F
feed composition	x_F	reflux temperature	T_0
reflux ratio	R	heat input (reboiler)	Q_S
number of stages	N	Murphree Vapour efficiency	E_{mv}

The mass balances for the lighter component over the individual trays of the column were written as follows:

Condenser -

$$V_1 \, y_1 = L_0 \, x_0 + D \, x_0$$

$$y_1 = y^*(x_1) = K_1 \, x_1$$

$y^*(x)$ = vapour/liquid equilibrium function

$$x_0 = x_D$$

$$\therefore \quad -(L_0 + D)x_0 + K_1 \, V_1 \, x_1 = 0 \qquad (6.1)$$

for the rectifying section, tray n -

$$V_n y_n + L_n \, x_n = L_{n-1} \, x_{n-1} + V_{n+1} \, y_{n+1}$$

$$y_n = K_n \, x_n \, , \quad y_{n+1} = K_{n+1} \, x_{n+1}$$

K_i = equilibrium ratio $y^*(x_i)/x_i$

$$L_{n-1} \, x_{n-1} - (K_n \, V_n + L_n)x_n + K_{n+1} \, V_{n+1} \, x_{n+1} = 0 \qquad (6.2)$$

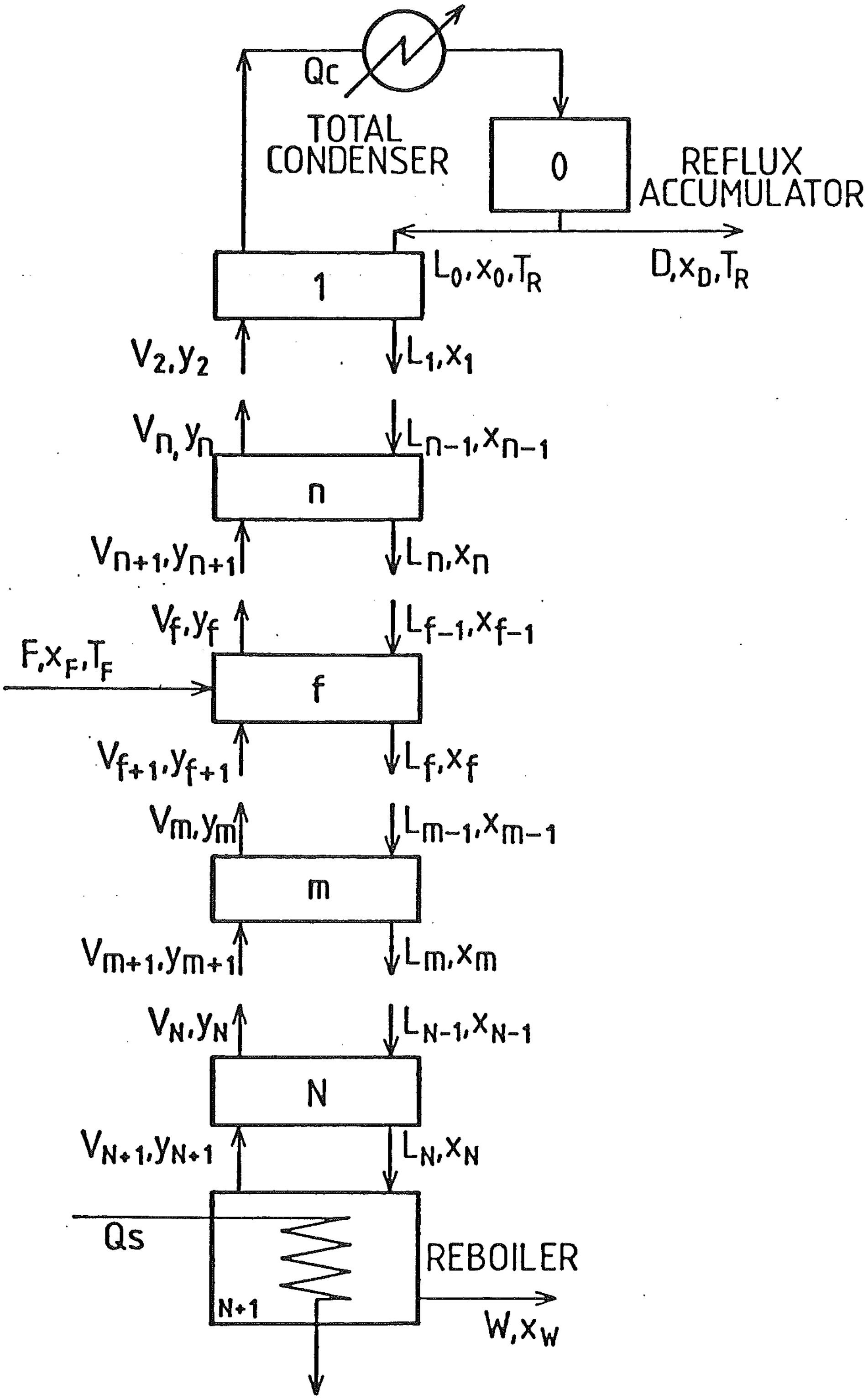


FIGURE 6-1 COLUMN MODEL VARIABLES

for the feed tray f -

$$V_f\, y_f + L_f\, x_f = L_{f-1}\, x_{f-1} + V_{f+1}\, y_{f+1} + F\, x_F$$

$$y_{f+1} = K_{f+1}\, x_{f+1}\,, \quad y_f = K_f\, x_f$$

$$L_{f-1}\, x_{f-1} - (K_f\, V_f + L_f) x_f + K_{f+1}\, V_{f+1}\, x_{f+1} = -\, F\, x_F \tag{6.3}$$

for the stopping section, tray m -

$$V_m\, y_m + L_m\, x_m = L_{m-1}\, x_{m-1} + V_{m+1}\, y_{m+1}$$

$$y_{m+1} = K_{m+1}\, x_{m+1}\,, \quad y_m = K_m\, x_m$$

$$L_{m-1}\, x_{m-1} - (K_m\, V_m + L_m) x_m + K_{m+1}\, V_{m+1}\, x_{m+1} = 0 \tag{6.4}$$

for the reboiler -

$$L_N\, x_N = V_{N+1}\, y_{N+1} + W\, x_w$$

$$y_{n+1} = K_{N+1}\, x_{N+1} = K_{N+1}\, x_w$$

$$L_N\, x_N - (K_{N+1}\, V_{N+1} + W)\, x_w = 0 \tag{6.5}$$

These equations could be written in tridiagonal form

$$Ax = b \tag{6.6}$$

where

$$A = \begin{bmatrix} B_1 & C_1 & & & & \\ A_2 & B_2 & C_2 & & & \\ - - - & - - - & - - - & & & \\ & A_n & B_n & C_n & & \\ & - - - & - - - & - - - & & \\ & & A_f & B_f & C_f & \\ & & - - - & - - - & - - - & \\ & & & A_m & B_m & C_m \\ & & & - - - & - - - & - - - \\ & & & & A_{N+1} & B_{N+1} \end{bmatrix} \tag{6.7}$$

and $A_i = L_{i-1}$

$B_i = -\,(K_i\, V_i + L_i)$ $\qquad B_1 = -(L_0 + D)$

$C_i = K_{i+1}\, V_{i+1}$ $\qquad B_{N+1} = -\,(K_N\, V_N + W)$

$$x = [x_o\ x_1 \mid x_n \mid x_f \mid x_m \mid x_{N+1}]^T \tag{6.8}$$

$$b = [0\ \ 0 \mid 0 \mid -Fx_F \mid 0 \mid 0\ \ ]^T \tag{6.9}$$

Given all L, V, K , then equation (6.6) could be solved simultaneously for the liquid compositions x . However, the K values were functions of the liquid compositions, and consequently an iterative solution was required. Further, if the model was to remain sufficiently general, then the liquid and vapour flows had to be found by making energy balances around the stages as follows:

$$D\ h_D + W\ h_W + Q_c = F h_F + Q_S \tag{6.10}$$

h_i = liquid enthalpy of stream i

H_i = vapour enthalpy of stream i

Q_C = Condenser heat load

for the condenser -

$$Q_C = (R + 1)\ D\ (H_2 - h_D) \tag{6.11}$$

from an overall mass balance

$$W = F - D \tag{6.12}$$

and combining (6.10), (6.11), (6.12)

$$D = \frac{F(h_F - h_W) + Q_S}{h_D - h_W + (R+1)(H_2 - h_D)} \tag{6.13}$$

and

$$L_1 = RD \tag{6.14}$$

$$V_2 = (R+1)D \tag{6.15}$$

for the rectifying section, tray n -

$$L_n = V_{n+1} - D \tag{6.16}$$

$$L_n\ h_n + V_n\ H_n = V_{n+1}\ H_{n+1} + L_{n-1}\ h_{n-1} \tag{6.17}$$

$$L_n = \frac{V_n\ H_n - D\ H_{n+1} - L_{n-1}\ h_{n-1}}{H_{n+1} - h_n} \tag{6.18}$$

for the feed plate f -

$$L_f = V_{f+1} + W \qquad (6.19)$$

$$L_{f-1}\, h_{f-1} + V_{f+1}\, H_{f+1} = L_f\, h_f + V_f\, H_f - F h_F \qquad (6.20)$$

$$L_f = \frac{V_f\, H_f - F h_F + W H_{f-1} - L_{f-1}\, h_{f-1}}{H_{f+1} - h_f} \qquad (6.21)$$

for the stripping section, plate m -

$$L_m = V_{m+1} + W \qquad (6.22)$$

$$L_{m-1}\, h_{m-1} + V_{m+1}\, H_{m+1} = L_m\, h_m + V_m\, H_m \qquad (6.23)$$

$$L_m = \frac{V_m\, H_m + W\, H_{m+1} - L_{m-1}\, h_{m-1}}{H_{m+1} - h_m} \qquad (6.24)$$

If the plate compositions were available, the equations given could be used to estimate the column flows. The procedure was to find the product flows, D and W , and then step down from the top of the column calculating pairs of liquid and vapour flows - L_i and V_{i+1} .

The model consisted of two sets of simultaneous equations - the compositions and the flows neither of which was explicit. The iterative solution technique used was as follows:

(i) to estimate all x, L, V

(ii) to form the matrix equation (6.6)

(iii) to solve equation (6.6) for new compositions

(iv) to solve the energy balance equations for new L, V

(v) to repeat steps (iii) and (iv) until convergence occurred.

6.3 SOLUTION OF THE DISTILLATION COLUMN MODEL

The equations derived in section 6.2 were programmed along with physical property data for the binary system into a digital computer and solved. The program, SSGW, was written in FORTRAN to run on the PDP-11 minicomputer described in section 4.5.1. It was designed to operate interactively through the minicomputer console.

6.3.1 Solving the Model

The equations describing the model have been derived in section 6.2. The solution to the component mass balances (equation (6.6)) was obtained using the recursive Gaussian elimination method of Thomas (Carnahan et al (1969)). The liquid and vapour flowrates were solved sequentially down the column, using the method outlined in section 6.2 once the tray compositions had been determined. The procedure was repeated until some convergence criteria was met. A number of criteria were tested:

$$\text{all} \quad | \; x_i - x_{i-1} \; | < \text{limit} \tag{6.25}$$

$$\text{all} \quad | \; T_i - T_{i-1} \; | < \text{limit} \tag{6.26}$$

$$\frac{\sum_{j=0}^{n+1} (T_{ij} - T_{i-1j})^2}{(N+2) \; (\frac{T_D + T_W}{2})} \times \frac{100}{1} < \text{limit} \tag{6.27}$$

where i = iteration number.

The root mean square temperature difference, (6.27), was chosen, but the choice was an arbitrary one.

To start the iterative procedure, it was necessary to estimate initially the column flows and compositions. The compositions were estimated from a linear temperature profile within the column. The top, T_D , and bottom, T_W , temperatures were either estimated by the user or computed by the equations

$$T_D = T_{x=1} + 0.05 \; (T_{x=0} - T_{x=1}) \tag{6.28}$$

$$T_W = T_{x=0} - 0.05 \; (T_{x=0} - T_{x=1}) \tag{6.29}$$

The individual tray temperatures were then estimated and the tray compositions determined from the temperature composition polynomial. This fifth order polynomial in composition was solved using a Wegstein root finding algorithm (Carnahan et al (1969)). The column flows were initially estimated by assuming equimolal overflow in the column.

The performance of each column tray had been assumed to be ideal in the model derivation but in practice this assumption was not valid. Various reasons for varying plate efficiency with composition have been suggested including mass transfer resistance between the phases, plate hydrodynamics and physical properties such as the slope of the equilibrium curve. Mostafa (1979) has proposed a model for tray behaviour to explain these phenomena. For this model, a Murphree vapour phase plate efficiency was used:

$$E_{MV} = \frac{y_{out} - y_{in}}{y^* - y_{in}} \tag{6.30}$$

y_{out} = composition of vapour leaving the tray

y_{in} = composition of vapour entering the tray

y^* = composition of vapour in equilibrium with the liquid leaving the tray

E_{MV} = Murphree vapour efficiency.

The Murphree vapour efficiency could be incorporated into the model by recursively computing the vapour compositions as:

$$y_1 = x_o$$

$$y_i = \frac{y_{i-1} - E_{MV}\, y^*(x_{i-1})}{1 - E_{MV}} \tag{6.31}$$

for $i = 2, 3 \ldots N+1$

as part of the iterative solution of the model. The reboiler may be considered as an ideal stage in (6.31).

However, this method produced instabilities in the calculational procedure especially during the initial stages. Equation (6.31) occasionally predicted vapour compositions greater than those possible (> 1.0 , < 0), and hence the program failed. To avoid this problem, the equilibrium data was modified in the manner normally used in the graphical McCabe-Thiele analysis, and a pseudo-equilibrium line produced. The modification of the equilibrium data required the use of straight operating lines (to have used curved operating lines would have greatly increased the computational effort required in the program) and this implied constant molal overflow. This situation did not occur for the methanol/water system, but the deviations were not too large. The procedure adopted was to modify the equilibrium line using the following equations in each iteration:

$$L_R = \frac{\sum_{i=0}^{f-1} L_i}{f} \qquad \text{average rectifying liquid flow} \qquad (6.32)$$

$$V_R = \frac{\sum_{i=1}^{f} V_i}{f} \qquad \text{average rectifying vapour flow} \qquad (6.33)$$

$$L_S = \frac{\sum_{i=f}^{N} L_i}{N-f+1} \qquad \text{average stripping liquid flow} \qquad (6.34)$$

$$V_S = \frac{\sum_{i=f+1}^{N+1} V_i}{N-f+1} \qquad \text{average stripping vapour flow} \qquad (6.35)$$

then the operating lines became

$$\text{rectifying:} \quad y_{i+1} = \frac{L_R}{V_R} x_i + \frac{D}{V_R} x_D \qquad (6.36)$$

$$\text{stripping:} \quad y_{i+1} = \frac{L_S}{V_S} x_i - \frac{W}{V_S} x_W \qquad (6.37)$$

and the point of intersection was

$$x_{\text{intercept}} = \frac{\dfrac{Dx_D}{V_R} + \dfrac{Wx_W}{V_S}}{\dfrac{L_S}{V_S} - \dfrac{L_R}{V_R}} \qquad (6.38)$$

The equilibrium data was then modified according to the following equations:

$$y' = x + E_{MV} (y^* - x) \qquad x \leq x_W, \ x \geq x_D \qquad (6.39)$$

$$y' = E_{MV}(y^* - \frac{L_S}{V_S} x + \frac{W}{V_S} x_W) + \frac{L_S}{V_S} x - \frac{W}{V_S} x_W \qquad (6.40)$$

$$x_W < x \leq x_{\text{intercept}}$$

$$y' = E_{MV}\,(y^* - \frac{L_R}{V_R}\,x - \frac{D}{V_R}\,x_D) + \frac{L_R}{V_R}\,x + \frac{D}{V_R}\,x_D$$

$$x_{intercept} < x < x_D \qquad (6.41)$$

where y^* = true equilibrium vapour composition

y' = pseudo equilibrium vapour composition for liquid of composition x .

The results of using this approach are discussed in section 6.4.2. A similar approach was used by Gerster et al (1962, 1964) and verified experimentally. It should be noted that this approach treats the reboiler as a non-ideal stage.

6.3.2 SSGW - A Steady State Binary Distillation Column Program

The equations presented in the previous sections were combined with suitable I/O routines. Interaction between the user and the program allowed alteration of any of the program variables in engineering units, e.g.

flows in ℓ/time

compositions in mole fractions

steam flow in kg/time

temperatures in °C

The user was able to enter or change program parameters, store parameters in disk files, recall parameters from disk files, control the output of the program, and initiate the solution of the column variables. The solution procedure used was

(i) to estimate compositions and flows in the column

(ii) to set up the tridiagonal matrix equation (6.6)

(iii) to solve the tridiagonal matrix equation

(iv) to recalculate the flows

(v) to check that minimum stages, and minimum reflux limitations were not exceeded

(vi) to recalculate the pseudo-equilibrium line based on the new compositions and flows

(vii) to test for convergence and if not achieved repeat steps (ii) through (vii)

(viii) to compute heat loadings, external volumetric flows, end temperatures, internal reflux ratio

(ix) to list results.

A flow diagram of the program is shown in figure 6-2, and a listing is given in Appendix VI.

6.3.3 Binary System Data

In order to solve the model, it was necessary to express the physical and thermodynamic properties of the liquid and vapour streams in a form suitable for digital computer use. Data was required for:

(i) the vapour/liquid equilibrium relationship

(ii) the boiling point relationship

(iii) the liquid and vapour enthalpies

(iv) the liquid heat capacities

(v) the liquid densities

all as a function of composition. Polynomials were used to fit the data in all cases except the vapour/liquid equilibrium data for which a lookup table was used. The system properties were handled by a separate program SYSHDL which created a disk file of the relevant parameters. SSGW then loaded this disk file when it was started. A listing of SYSHDL is given in Appendix VI.

6.4 VERIFICATION OF THE STEADY STATE MODEL

The distillation column described in Chapter 3 was compared with the model presented. The property data required for the binary system methanol/water was correlated as follows:

(i) Vapour/liquid equilibrium data - a lookup table of 41 vapour equilibrium compositions for equally spaced liquid compositions in the range 0-1 m.f. was used; linear interpolation was used within the table.

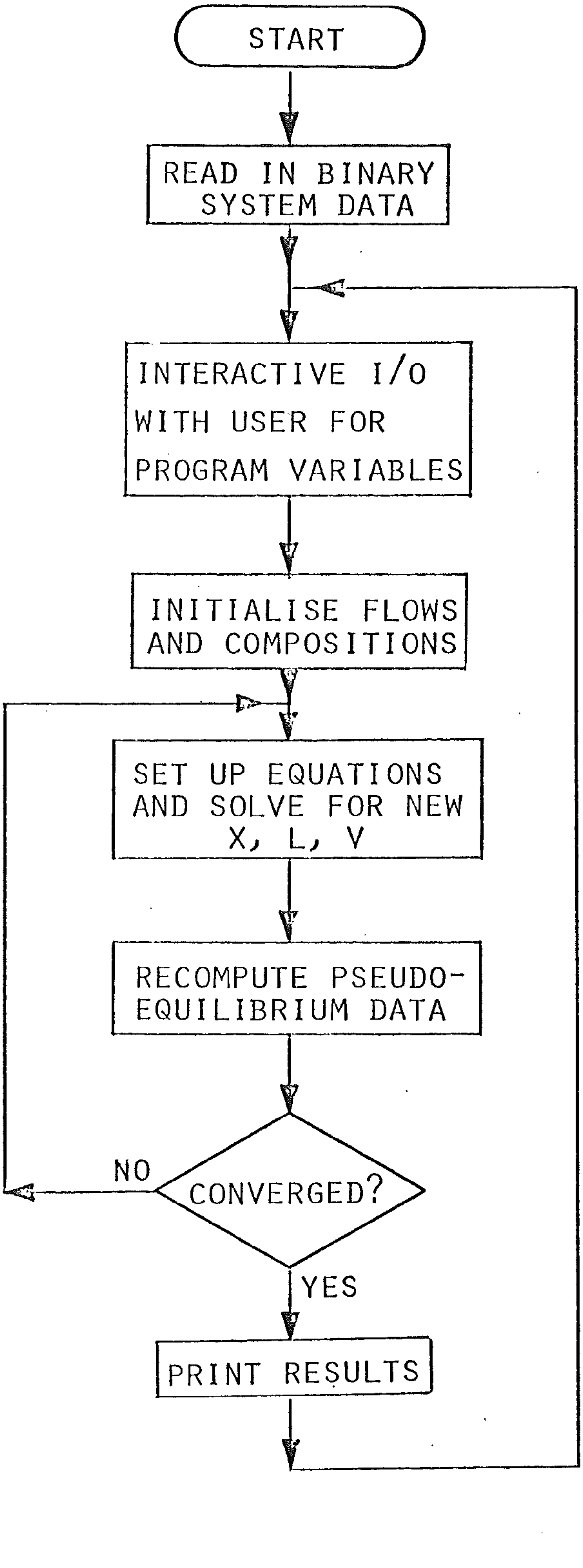

START
READ IN BINARY SYSTEM DATA
INTERACTIVE I/O WITH USER FOR PROGRAM VARIABLES
INITIALISE FLOWS AND COMPOSITIONS
SET UP EQUATIONS AND SOLVE FOR NEW X, L, V
RECOMPUTE PSEUDO-EQUILIBRIUM DATA
CONVERGED?
NO
YES
PRINT RESULTS

FIGURE 6-2 SSGW FLOW DIAGRAM

(ii) Boiling point data - a fifth order polynomial was used to relate the liquid boiling point to the liquid composition.

(iii) Enthalpy data - fifth order polynomials were used to relate the liquid and vapour enthalpies to composition.

(iv) Heat capacity data - a mole fraction weighted average of the pure component heat capacities at the required temperature was used.

(v) Liquid density data - for saturated liquids, a linear relationship between density and composition was used; for subcooled liquids, a mole fraction weighted average of the pure component densities at the required temperature was used.

The correlations used are listed in Appendix VII.

The computer model was used in comparison with experimental data. To account for measurement and process errors in the input model parameters, a sensitivity analysis was carried out.

6.4.1 Sensitivity Analysis

Calculation of the partial derivatives of the model equations for use in a sensitivity analysis produced very complex expressions. Instead, the computer model was subjected to small perturbations about a single operating point, and the partial derivatives estimated from the steady state results. The operating point is given in Table 6-1; the values were typical experimental results from the column, and the uncertainty in each measurement is given.

TABLE 6-1 - OPERATING CONDITIONS FOR SENSITIVITY ANALYSIS

Variable	Value	Experimental Uncertainty	Units
Feed Rate	1.75	± .005	ℓmin^{-1}
Feed Composition	0.460	± .005	m.f.
Feed Temperature	26.0	± 1	°C
Reflux Ratio	0.73	± 0.03	-
Reflux Temperature	20.0	± 1	°C
Steam Flowrate	1.03	± .02	kg min^{-1}
Murphree Vapour Efficiency	0.80	± .02	-

For a general sensitivity analysis (assuming random errors)

$$U = \text{function of } (X, Y, Z)$$

$$u^2 = \left(\frac{\partial U}{\partial X} \right)^2 x^2 + \left(\frac{\partial U}{\partial Y} \right)^2 y^2 + \left(\frac{\partial U}{\partial Z} \right)^2 z^2 \qquad (6.42)$$

where u, x, y, z are the absolute errors in U, X, Y, Z . The partial derivatives can be estimated by a central difference approximation

$$\frac{\partial U}{\partial X} \simeq \frac{U_{X+\delta X} - U_{X-\delta X}}{2\delta X}$$

where δX = the perturbation in X .

Each of the variables in Table 6-1 was individually perturbed by ±5% from its steady state value. The partial derivatives were calculated using the above formula, and, assuming the principle of superposition, a determinate linear model was found. For random errors in the model inputs, equation (6.42) was used to produce a variance model. The linear model related the effects of small variations in the model input parameters to the major model outputs. The variance model related the likely error in the output parameters to the experimental measurement errors in the model input parameters. Both models are shown in figure 6-3. Substituting the measurement errors in Table 6-2 into the variance model produced confidence estimates for the product compositions and flows predicted from the computer model. These estimates are shown in Table 6-2.

TABLE 6-2

ERROR ESTIMATES ON SSGW PREDICTED VARIABLES

Variable	Likely Absolute Error	Likely % Error
x_D	± .009 m.f.	± 1%
x_W	± .032 m.f.	±63%
D	± .81 mol min^{-1}	± 3%
W	±2.41 mol min^{-1}	± 7%
L_0	± .89 mol min^{-1}	± 4%

FIGURE 6-3

SENSITIVITY MODELS FOR SSGW

(i) Linear Model

$$\begin{bmatrix} \Delta x_D \\ \Delta x_W \\ \Delta D \\ \Delta W \\ \Delta L_0 \end{bmatrix} = \begin{bmatrix} .047 & .106 & -.001 & -.0006 & .270 & .122 & -.041 \\ .817 & .376 & -.065 & -.010 & -.270 & .220 & -.820 \\ -1.020 & 4.60 & 2.21 & .500 & .200 & -12.6 & 32.2 \\ 64.1 & -28.1 & -3.38 & -.500 & -.200 & 12.6 & -32.2 \\ -.583 & 3.22 & 1.56 & .400 & .200 & 13.7 & 23.5 \end{bmatrix} \begin{bmatrix} \Delta F' \\ \Delta x_F' \\ \Delta T_F' \\ \Delta T_0' \\ \Delta E_{MV}' \\ \Delta R' \\ \Delta Q_S' \end{bmatrix}$$

$$\Delta F' = \frac{\Delta F}{F} \quad \text{etc.}$$

(ii) Variance Model

$$\begin{bmatrix} \Delta x_D^2 \\ \Delta x_W^2 \\ \Delta D^2 \\ \Delta W^2 \\ \Delta L_0^2 \end{bmatrix} = \begin{bmatrix} .001 & .053 & 1.6E-9 & 9.E-10 & .114 & .028 & .002 \\ .218 & .668 & 6.3E-6 & 2.5E-7 & .114 & .091 & .634 \\ .340 & 100. & .007 & 6.3E-4 & .063 & 300. & 976. \\ 1338. & 3721. & .017 & 6.3E-4 & .063 & 294. & 976. \\ .111 & 49. & .004 & 4.0E-4 & .063 & 350. & 520. \end{bmatrix} \begin{bmatrix} \Delta F^2 \\ \Delta x_F^2 \\ \Delta T_F^2 \\ \Delta T_0^2 \\ \Delta E_{MV}^2 \\ \Delta R^2 \\ \Delta Q_S^2 \end{bmatrix}$$

From this analysis, it can be seen that the bottom of the column is very sensitive to errors in specifying the model inputs. The factors contributing most to the uncertainty in the bottoms composition and flowrate were the feedrate and the steam rate; these variables were measured as carefully as possible to ensure that uncertainties did not swamp the results. The estimated errors in Table 6-2 were used in the verification of the actual column experiments against the model prediction.

6.4.2 Model Comparison with Experimental Results

A number of runs on the column were made under different operating conditions. Before the experimental results could be compared against the model predictions, it was necessary to estimate the heat losses from the column, and to determine a correlation for the Murphree vapour efficiency as a function of composition.

Surface temperature measurements were made on the outside of the column using a copper constantan thermocouple. Using estimated heat transfer coefficients for natural convection and estimated surface emissivities (Welty et al (1969)), the measured temperatures were used to estimate the heat losses from the reboiler and column trays:

Reboiler - area = 1.8 m^2, ε = 0.35 , $h \simeq 4$ $W_m^{-2}K^{-1}$

temperature difference = 95°C - 20°C

$Q \simeq$ 930W

Trays - area = 3.0 m^2, ε = 0.85 , $h \simeq 3$ $W_m^{-2}K^{-1}$

temperature difference = 80°C - 20°C

$Q \simeq$ 1700W

$Q_{total} \simeq$ 2630W

and using steam with a latent heat of vaporisation of 2250 kJ kg^{-1}, the heat loss corresponds to 0.07 kg min^{-1} of steam. Hence the measured steam flow was corrected by subtracting 0.07 kg min^{-1}; this analysis assumed all the heat losses were lumped into the reboiler which was not strictly true. An individual loss for each plate could have been incorporated in the energy balance equations. The heat loss represented a small fraction (≃ 7%) of the total energy input and this treatment of heat losses seemed reasonable.

The problem of deciding a priori on a Murphree vapour efficiency relationship with composition has been investigated by Hay and Johnson (1960) and Bakowski (1969) for methanol/water separations in sieve tray columns. Their results depended on the method of analysis used. Hay and Johnson (1960) investigated the mass transfer on a tray for varying liquid/vapour ratios, and measured Murphree vapour efficiencies as a function of composition. These values are at variance with those predicted by Bakowski (1969) because the latter assumed that the point and Murphree vapour efficiencies were equal. The results of these workers are compared in figure 6-4.

The method of Bakowski was used to predict the Murphree vapour phase efficiency since the method had been shown to reproduce experimental results, and could be used a priori requiring only the properties of the binary system and the tray geometry. There are more rigorous methods of determining Murphree vapour efficiencies based on individual mass transfer resistances (Treybal (1968)).

Bakowski's method is

$$E_{MV} = \frac{1}{1 + \frac{3.7\ KM}{h\ \rho T}} \qquad (6.43)$$

where K = y^*/x , the vapour/liquid equilibrium ratio

M = molecular weight

ρ = liquid density (kg ℓ^{-1})

T = absolute temperature (K)

h = active liquid depth on the tray = weir height (cm)

3.7 = experimentally determined constant.

The Murphree vapour efficiency against composition curve for methanol/

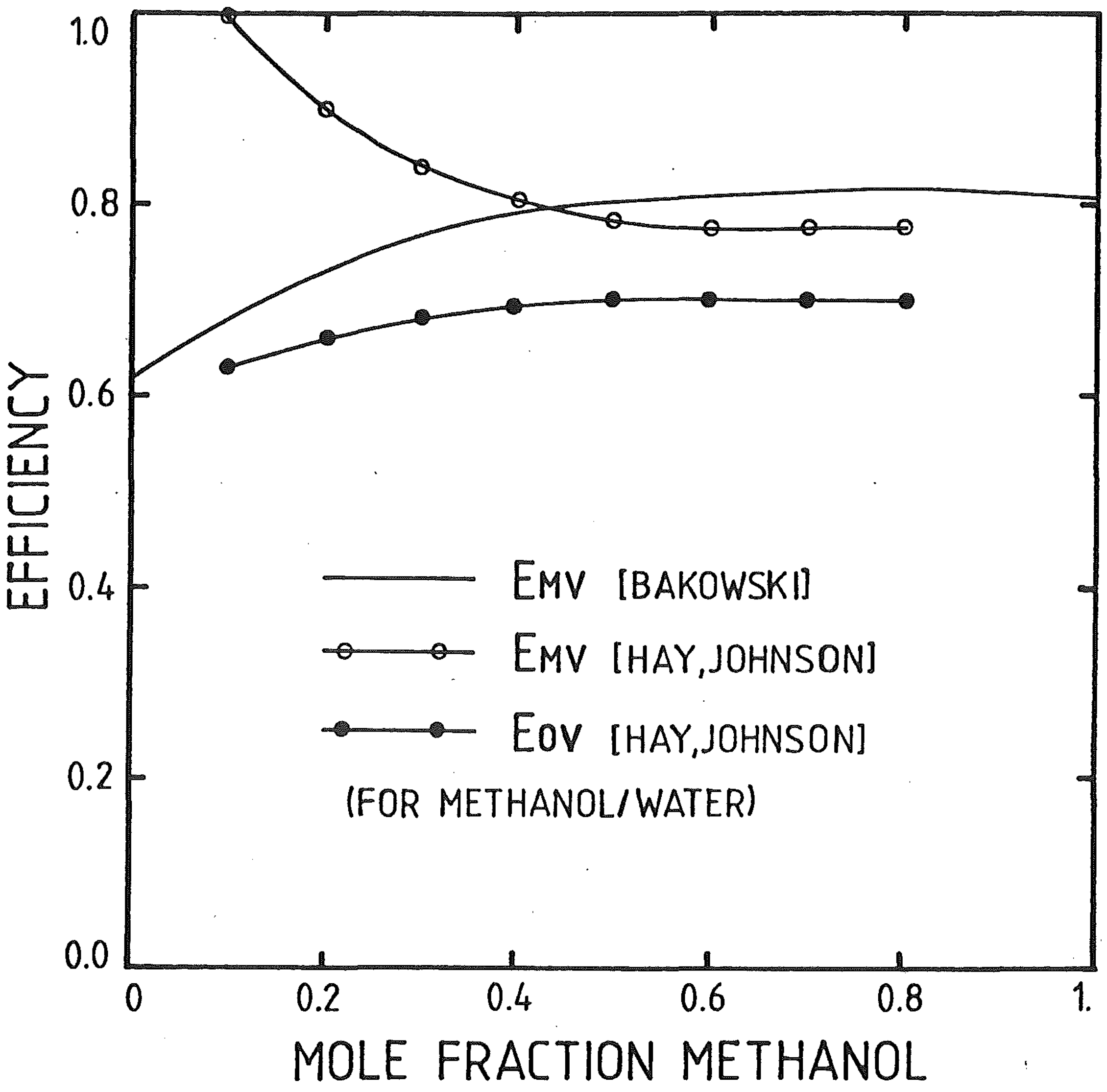


FIGURE 6-4 TRAY EFFICIENCIES

water is shown in figure 6-4. The curve was fitted with a cubic polynomial for inclusion in the model.

$$E_{MV} = .614 + .786x - 1.03x^2 + .436x^3 \qquad (6.44)$$

for h = 1.9 cm.

Using the data presented, a number of steady state experimental runs were compared against the model using Table 6-2 to account for differences between the two. The results are summarised in Appendix VIII. Nineteen of the twenty-two runs were found to agree within the likely errors. The parameters used for determining the model fit were

(i) distillate composition

(ii) bottoms composition

(iii) distillate flow

(iv) bottoms flow

(v) reflux flow.

The agreement between the model and the experimental data was good in all cases except three where the reflux ratio was greater than 2.2 times the minimum reflux ratio. In these circumstances, the model equation solution procedure became unstable and failed to converge.

6.5 DISCUSSION

Using experimental results, the model was examined to determine the significance of the assumptions made, and its applicability to a particular distillation column.

(i) The model used differed from other models proposed in the assumption of non-equimolal overflow. Table 6-3 shows the variation in the molal latent heat of vaporisation for the methanol/water system. The deviation from ideality is not large but it is significant. The data in Appendix VIII was analysed using an assumed average constant latent heat of vaporisation, and the results are shown in Table 6-4. Small changes in the column flows compound in their effect on the slope of the operating line (if the liquid flow increases, then the vapour flow must decrease).

TABLE 6-3

MOLAL LATENT HEATS OF VAPORISATION FOR METHANOL/WATER

Mole Fraction Methanol	ΔH_V/kJ mol^{-1}
0.0	40.69
0.1	40.86
0.2	40.58
0.3	40.05
0.4	39.38
0.5	38.69
0.6	38.02
0.7	37.38
0.8	36.77
0.9	36.10
1.0	35.30

TABLE 6-4

EFFECT OF ASSUMING EQUIMOLAL OVERFLOW ON COLUMN FLOW RATES

Flow Variable	Absolute Average % Change (over all runs)
Rectifying Liquid	9.1
Rectifying Vapour	3.8
Stripping Liquid	4.0
Stripping Vapour	6.5

To demonstrate the effect of assuming equimolal overflow consider Run 21. The experimental measurements were used to estimate the column flows assuming equimolal overflow. The results are summarised in Table 6-5 and compared with the equivalent flows predicted by the model for the same conditions. The steam flow predicted from the equimolal overflow analysis was higher than the experimentally measured flow. This indicated that an equimolal overflow analysis for predicting the required steam flow in a feedforward controller would be inadequate.

TABLE 6-5

COMPARISON OF EQUIMOLAL AND NON-EQUIMOLAL OVERFLOW ANALYSES

(Run 21)

Variable	Experimentally Measured Flow	Equimolal Calculation	Model Predictions (Non-Equimolal)
D	31.2 mol min^{-1}	31.2 mol min^{-1}	31.0 mol min^{-1}
W	32.6 " "	32.6 " "	32.4 " "
L_R	21.2 " "	24.0 " "	22.9 " "
V_R	52.4 " "	56.4 " "	53.8 " "
L_S	89.4 " "	95.8 " "	89.4 " "
V_S	56.8 " "	63.0 " "	58.5 " "
Q_S	1.03 kg min^{-1}	1.21 kg min^{-1}	1.03 kg min^{-1}

(ii) Murphree Vapour Efficiency. The method of including a Murphree vapour efficiency into the model was not strictly correct in that it required the use of average flows through the column sections. This particular computational method was chosen for model stability. From the results generated for comparison with the experimental data, the Murphree vapour efficiencies were calculated according to equation (6.30) from the model predictions. The results from a number of runs are shown in figure 6-5 in comparison with the Murphree vapour efficiency correlation used in the model. When the feed was not on the optimum tray as predicted by the McCabe-Thiele analysis the efficiency was found to vary from 50% to 90% (for liquid compositions 45 to 50 mol%) compared with the expected value of 80%. The overall effect on the model predictions was small since only one tray was affected, and if the reflux ratio tended toward the minimum reflux ratio there would be a small degree of separation on that tray.

The effect of assuming 100% tray efficiency was investigated and typical results are shown in Table 6-6.

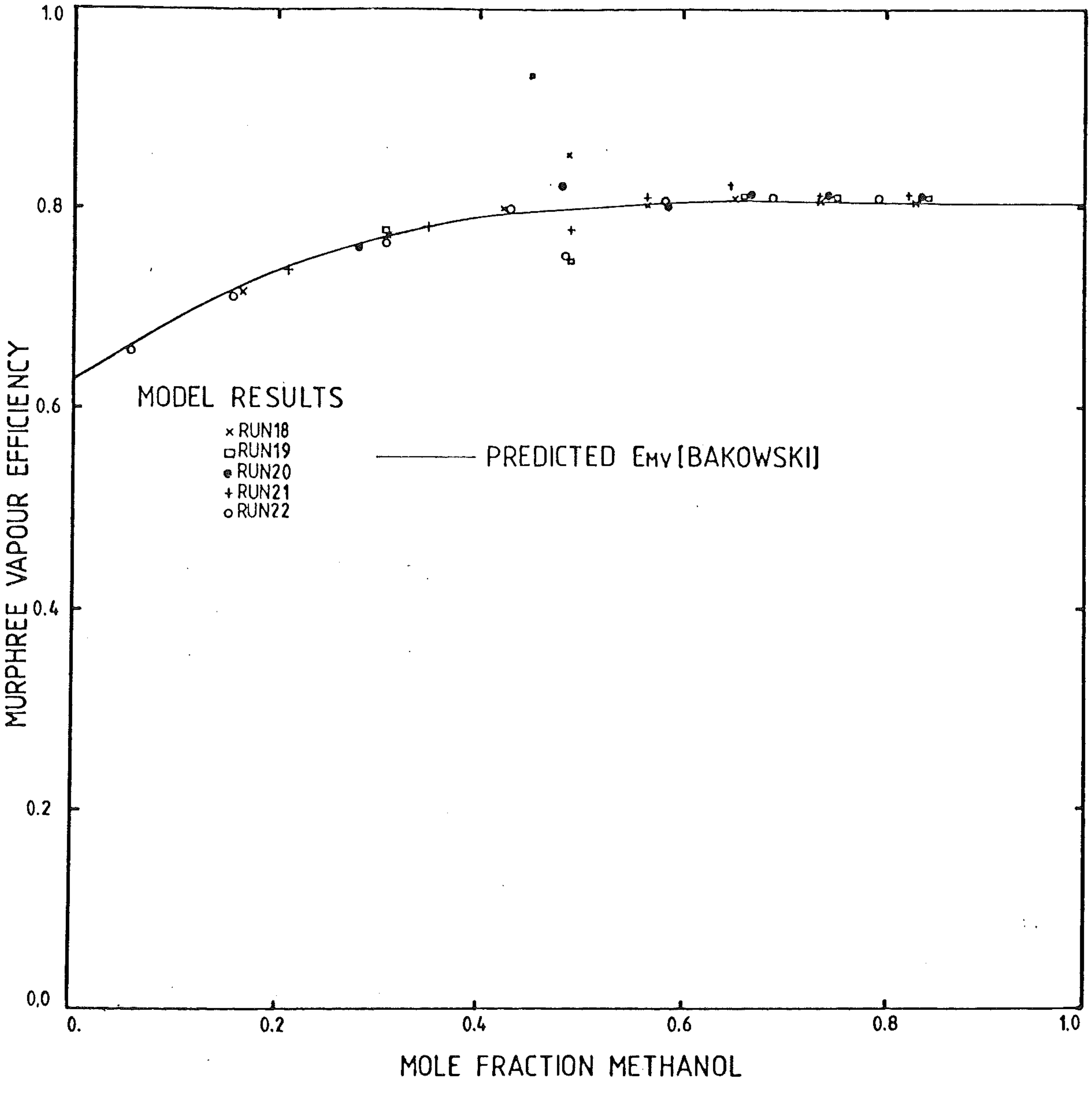


FIGURE 6-5 COMPARISON OF TRAY EFFICIENCIES

TABLE 6-6

THE EFFECT OF TRAY EFFICIENCY ON THE MODEL PREDICTIONS

RUN	VARIABLE$^+$	EXPERIMENTAL MEASUREMENTS	MODEL PREDICTIONS $E_{MV} = 1$	E^*_{MV}
14	x_D	92.0	94.87	92.59
	x_W	2.5	1.71	3.92
	D	31.7	31.04	31.02
	W	31.5	32.81	32.83
	L_O	28.8	28.24	28.23
18	x_D	91.5	93.49	92.14
	x_W	15.0	15.76	16.68
	D	24.4	25.04	25.00
	W	39.3	39.32	39.35
	L_O	18.0	18.53	18.50
21	x_D	91.5	93.58	91.75
	x_W	5.0	4.39	6.18
	D	31.2	30.97	30.95
	W	32.6	32.36	32.38
	L_O	23.2	22.92	22.91

^+x_D, x_W are mol%, D, W, L_0 are mol min^{-1}

* E_{MV} predicted using equation (6.44)

A comparison of the model predictions, using two different Murphree vapour efficiencies, with the experimental results showed discrepancies (Table 6-6), and highlighted the difficulty of specifying a priori the efficiencies. The method of Bakowski (1969) provided a first estimate of the Murphree vapour efficiencies, but as shown by Hay and Johnson (1960), the tray geometry, compositions, and operating conditions could significantly affect the tray efficiency. It was not possible to estimate the tray efficiencies from experimental measurements because sampling ports for the vapour streams were not available, and temperature measurements were unreliable (it was difficult to measure the temperature corresponding to the liquid and vapour flows leaving the tray).

An alternative approach would be to incorporate the Murphree vapour efficiency into the model equations as shown by Sargent and Murtagh (1969). This approach would require modifications to the solution technique used on the model equations. The good agreement between the experimental data and the model predictions suggested that the benefits of this approach would be small, and that the effects of the other factors discussed in this section would be more significant.

(iii) Model Stability and Convergence. The model solution converged except when the reflux ratio/minimum reflux ratio was greater than 2.2 or the product compositions were outside the range 0 - 1. In these cases, the solution oscillated between two states neither of which was the true solution. The experimental column was not generally operated under these conditions and consequently the convergence problem was ignored. The rate of convergence of the solution was asympototic and depended on the column operating conditions. The major factor was the location of the feed tray in the column. If the steady state composition on the feed tray, x_f, was significantly different from the feed composition, x_F, then a large number of iterations were required. If $x_F < x_f$, convergence was slow (300 iterations), but if $x_F > x_f$, convergence was obtained in 30-50 iterations (with the root mean square temperature convergence criteria = 0.001). Execution time on the PDP-11 minicomputer described in section 4.5 was approximately 30 seconds for 100 iterations.

(iv) Sensitivity Analysis. The assumptions made in this analysis were

(a) that the model was linear for small perturbations;

(b) that the errors on the experimental values were random;

(c) that the partial derivatives could be approximated by a central difference. The form of the model equations made any other approach intractable. The errors in the process measurements are detailed in Table 6-7. The measurements were subject to both systematic and random errors; the former were minimised by careful calibration and technique, and were considered to be insignificant compared to the latter.

TABLE 6-7

MEASUREMENT ERRORS

Variable	Method of Measurement	Likely Measurement Error
Feed Flow	Pump Calibration	± .05 ℓmin^{-1}
Feed Composition	Density Bottle	± .005 m.f.
Feed Temperature	Integrated Circuit Sensor	± 1°C
Reflux Temperature	Integrated Circuit Sensor	± 1°C
Reflux Ratio	Flow Measurement	± 0.03
Steam Flow	Bucket & Stopwatch	± .02 kg min^{-1}
Product Composition	Refractometer	± .005 m.f.
Product Flows	Rotameter	± .02 ℓmin^{-1}
Reflux Flow	Rotameter	± .02 ℓmin^{-1}

The models in figure 6.3 show some interesting features. The most noticeable was the effect of small errors in the input parameters on the bottoms product and flow predictions; the errors on these variables were about three times those on the corresponding distillate predictions. The shape of the vapour/liquid equilibrium curve explained this effect;

at low concentrations ($x = 0.05$, relative volatility, $\alpha = 6.9$) the curve was very steep while at high concentrations it was shallow ($x = .90$, $\alpha = 2.5$). Small changes in the stripping line produced much larger changes in the tray compositions than did similar changes in the rectifying line. Consequently, there was a relatively large uncertainty in the prediction of the bottoms flow and composition from the computer model. In general the model predictions for bottoms composition were higher than those measured experimentally.

(v) Variation of Property Data. The system property data for methanol/water as given in Appendix VII was at variance with some of the data in the literature. In some areas, there was a lack of data.

(a) Density - the correlations used were checked against Gallant (1968). The two data sets as shown in Appendix VII showed reasonable agreement with a maximum deviation of 1.4% on the subcooled mixture and - 2.4% on the saturated mixture.

(b) Enthalpy - the variation between the sources is shown in Appendix VII. The data quoted was at variance even for the pure component enthalpies. The chosen set had the best agreement with other measurements of the pure component enthalpies. The deviations between the two sets were 3.5% for the vapour enthalpy and 14% for the liquid enthalpy. In terms of the molal latent heat of vaporisation, the maximum deviation was 2%.

(c) Heat capacity - there was a lack of heat capacity data for methanol/water mixtures close to the saturation temperature, hence a simple molal mixing law based on the pure component values was used. A comparison of the predicted and literature values is given in Appendix VII. A maximum deviation of 5% between the two sets was observed in the middle of the range. The effects of such inaccuracies in the heat capacities was calculated to produce errors in the internal reflux and feed tray liquid flows of approximately 0.5%.

(d) Equilibrium data - as can be seen from the figure in

Appendix VII, there were a number of published sets of equilibrium data for methanol/water at atmospheric pressure. The works of Cornel and Montonna (1933), Hughes and Maloney (1952), Doroshevsky and Polansky (1910), Uchida and Kato (1934) and Othmer and Benenati (1945) all agree, but were opposed by the results of Green and Vener (1955) and Bredig and Bayer (1927). Variations in experimental technique would appear to explain the differences of up to 2% between these two groups. The effect of other trace components (aldehydes, ketones, ammonia, acetic acid and sulphur compounds) in the methanol feedstock used in this work would also alter the equilibrium characteristics. The choice of data then became arbitrary, and hence the data of Cornel and Montonna (1933) as quoted by Perry (1963) was used to be consistent with other published work (Svrcek (1967)).

There was a degree of uncertainty in the system property data as used in the computer model. The uncertainty was caused both by conflicting data in the literature and by the simplifications in the correlations used (simple mixing models).

(vi) Model Verification. The results in Appendix VIII showed good agreement between the experimental data and the model predictions in 19 out of 22 runs. The agreement on the distillate composition and flow was within the likely error of ± 1 mol%. The bottom composition and flow were also in agreement within the likely error (± 3.2 mol%).

As a further check, the model was used on the experimental data of Svrcek (1967) and a comparison for two runs is given in Appendix VIII. The agreement was good in the light of an assumed Murphree vapour efficiency of 0.9 (calculated from Bakowski's (1969) correlation) and no estimate of error on the experimental data. Svrcek included heat losses in each tray of his model but SSGW assumed a lumped heat loss in the reboiler stage. This approach was acceptable in view of the agreement between the experimentally measured flows and compositions, and the predicted equivalents.

(vii) Heat Losses. The model heat balance equations could incorporate heat transfer on each stage if required, but for simplicity the heat losses were lumped into the reboiler stage. Experimental temperature measurements and heat transfer correlations (Section 6.4.2) predicted a total column heat loss equivalent to 0.07 kg min^{-1} of steam (dry saturated at 212 kPa). The experimental results of Svrcek (1967) for a column of almost identical dimensions showed a heat loss equivalent to 0.06 kg min^{-1} of steam. This, and the agreement of the model and experimental data lent support to the approach of a single heat loss in the reboiler stage.

6.6 CONCLUSION

A steady state binary distillation column model was derived and programmed on a digital computer. The model was shown to agree with experimental data from a pilot plant scale distillation column and with published data (within experimental errors). The model showed that the assumptions of equimolal overflow and constant relative volatility were not valid for the methanol/water binary system. The prediction of tray efficiencies was shown to be important but difficult to achieve with simple correlations. The variance between the published property data for the methanol/water system contributed to discrepancies between the experimental data and the model predictions.

6.7 NOMENCLATURE

A - coefficient array, equation (6-6)

b - vector, equation (6-6)

B - coefficient array, equation (6-6)

C - coefficient array, equation (6-6)

C_p - liquid heat capacity, kJ $\text{mol}^{-1}\text{K}^{-1}$

D - distillate flow, mol min^{-1}

E_{MV}	- Murphree vapour efficiency
F	- feed rate, $\ell\,\mathrm{min}^{-1}$
h	- liquid enthalpy (saturated), kJ mol^{-1}
H	- vapour enthalpy (saturated), kJ mol^{-1}
K	- equilibrium ratio, y^*/x
L	- liquid flow, mol min^{-1}
M	- molecular weight
N	- nunber of column trays
Q	- heat load, kW
Q_C	- condenser heat load, kW
Q_S	- reboiler steam flow, kJ min^{-1}
R	- reflux ratio
T	- temperature, °C
V	- vapour flow, mol min^{-1}
W	- bottoms flow, mol min^{-1}
x	- liquid composition, m.f.
y	- vapour composition, m.f.
y*	- equilibrium vapour composition, m.f.
α	- relative volatility
ρ	- liquid density, kg ℓ^{-1}

Subscripts

D	- distillate stream
f	- feed tray
F	- feed stream
N	- tray N
R	- rectifying section
S	- stripping section
W	- bottoms stream
0	- reflux accumulator

CHAPTER SEVEN

COLUMN DYNAMICS AND FEEDBACK CONTROL

CONTENTS

Page

CHAPTER SEVEN

COLUMN DYNAMICS AND FEEDBACK CONTROL

7.1 INTRODUCTION

Distillation columns have been investigated extensively both experimentally and in computer simulations. Rademaker et al (1975) has presented a comprehensive review of the literature on distillation column dynamics and control. A large proportion of the work has been on simulation studies without extension to experimental verification and control system implementation. Almost every distillation unit or system is unique in some way that precludes a generalised analysis, and each situation should be treated on its merits. The column used in this work was simplified by operation at constant pressure (atmospheric), and the use of a binary system (methanol/water).

Investigation of column dynamics can follow two different paths. A rigorous description for a plate distillation column using a series of ordinary differential equations (for mass and energy balances) can be solved for a variety of column upsets. Controllers can be introduced and their effectiveness evaluated. The alternate procedure is to make tests on the plant and to interpret the results in approximate models which may extrapolate less well than the rigorous model. Lamb et al (1961), Huckaba et al (1963, 1965), and Svrcek (1967), among others, have demonstrated the application and the verification of the rigorous approach; Jafri et al (1965), Marino et al (1968) and Wood and Berry (1973), among others, have fitted models to experimental data with some success. Attempts to predict simple models from steady state operating data have been made by Gilliland and Mohr (1962), Bhat and Williams (1969), Weigand et al (1972) and Buffham (1974). Their work has shown reasonable agreement with simulation models of large columns (more than 25 plates).

Process dynamics were required in this project to estimate the likely column response to upsets, to identify suitable controllers, and to estimate the controller parameters required for best performance. Simple process models were fitted to experimental response data, and controllers designed and tuned using these models. The object was to produce the simplest control scheme that could provide acceptable controlled responses.

Model identification has been extensively investigated in the literature; Gustavsson (1975) has reviewed the application of identification techniques to general chemical and physical processes while Khrishnamoorthy and Edgar (1977) have summarised the identification of distillation column models. Simple models were fitted to experimental data using discrete models fitted in the z domain in this project.

There are a number of reported methods of designing controllers based on z plane models (Kuo (1963), Tou (1969), Mosler et al (1967)). There are advantages when such design methods produce controllers which are discrete equivalents of the standard PID analog controller. In particular, the well recognised methods of tuning PID controllers can be used, and the intuitive feel for the effects of the proportional, integral and derivative modes can be utilised in fine tuning procedures either by the computer or by the operator.

There are four manipulated variables on an atmospheric pressure distillation column assuming constant pressure operation, and a fixed condenser load. The two control loops that maintain the overall column material balance by controlling the reflux accumulator and reboiler levels were considered as being lumped into the column dynamics. The control problem then was to maintain the product compositions using the two remaining control variables.

Temperature control on the top and bottom trays was adopted in preference to composition control on the products because of the long response time of the on-line refractometer analyser, and the lags caused by the liquid holdups in the reflux accumulator and reboiler.

The dynamic behaviour of the distillation column described in the preceding chapters was investigated. Feedback controllers were designed and tuned to give best performance under a wide range of upsets using the microcomputer-based control system described in Chapter four.

7.2 LEVEL CONTROLLERS

The atmospheric pressure column had four control variables and four controlled variables, and hence 24 control combinations were possible. A number of these arrangements were obviously inferior and noting that the response of the bottom composition was much faster when controlled by manipulating the steam flow to the reboiler rather than bottom flow, then two possible control configurations remained (figure 7-1). The only real choice was the selection of the distillate composition control variable.

Shinskey (1967) has suggested that the material and heat balances should be separated (by controlling the distillate composition with the distillate flow in this case), but this is important only when the reflux stream is on flow control. For the column used in this project, the reflux ratio was generally close to 1.0 and the level controllers were reasonably tightly tuned (figure 7-3), hence the choice of distillate or reflux flow for distillate composition control was arbitrary.

Composition control is best achieved by maintaining the separation constant in the column (i.e. constant temperature profile), and it is necessary to ensure that temperature changes on the top tray are corrected as soon as possible. Therefore it is reasonable to control the top tray temperature by manipulating the reflux flow, i.e. figure 7-1b. This configuration was tried and worked well as will be shown in the following sections.

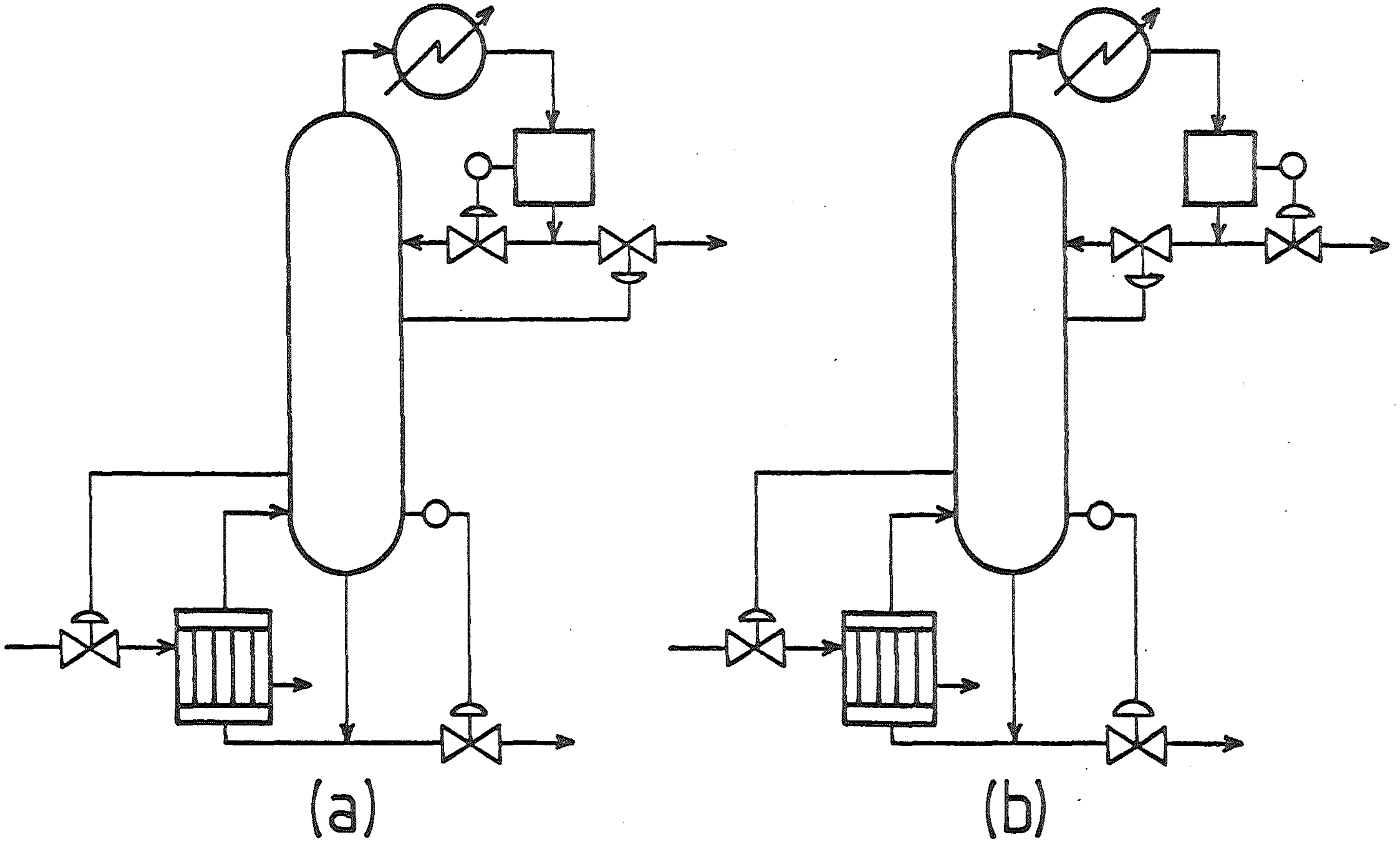


FIGURE 7-1 CONTROL CONFIGURATIONS

7.2.1 Level Controller Design

The reflux accumulator level and reboiler level systems were similar in that each attempted to control the level in a 225 mm diameter column section by a speed controllable pump. The following analysis then applies to both level controllers.

The block diagram for the level control system is shown in figure 7-2. Analysis of the block diagram for upset disturbances only gives (Smith (1972)):

$$C(z) = \frac{NG(z)}{1 + D(z)\ HG(z)\ K_p K_s} \tag{7.1}$$

or

$$D(z) = \frac{NG(z) - C(z)}{K_p K_s\ HG(z)\ C(z)} \tag{7.2}$$

where $NG(z)$ = upset pulse transfer function

$HG(z)$ = process pulse transfer function.

Design of a deadbeat type controller can be achieved by solving (7.2) for a specified $C(z)$ and given $NG(z)$. The tightest possible level control is that in which the level returns to the setpoint in two sampling intervals given a step upset u. For the system in figure 7-2 then

$$HG(z) = \frac{KTz^{-1}}{1 - z^{-1}} \tag{7.3}$$

$$NG(z) = \frac{KuTz^{-1}}{1 - z^{-1}} \tag{7.4}$$

$$C(z) = KuTz^{-1} \tag{7.5}$$

substituting into (7.2)

$$D(z) = \frac{2 - z^{-1}}{KTK_p K_s(1 - z^{-1})} \tag{7.6}$$

or in difference form

$$m_n - m_{n-1} = \frac{1}{KT\,K_p K_s}(2\,e_n - e_{n-1}) \tag{7.7}$$

Equation (7.7) is the velocity form of a discrete PI controller. How-

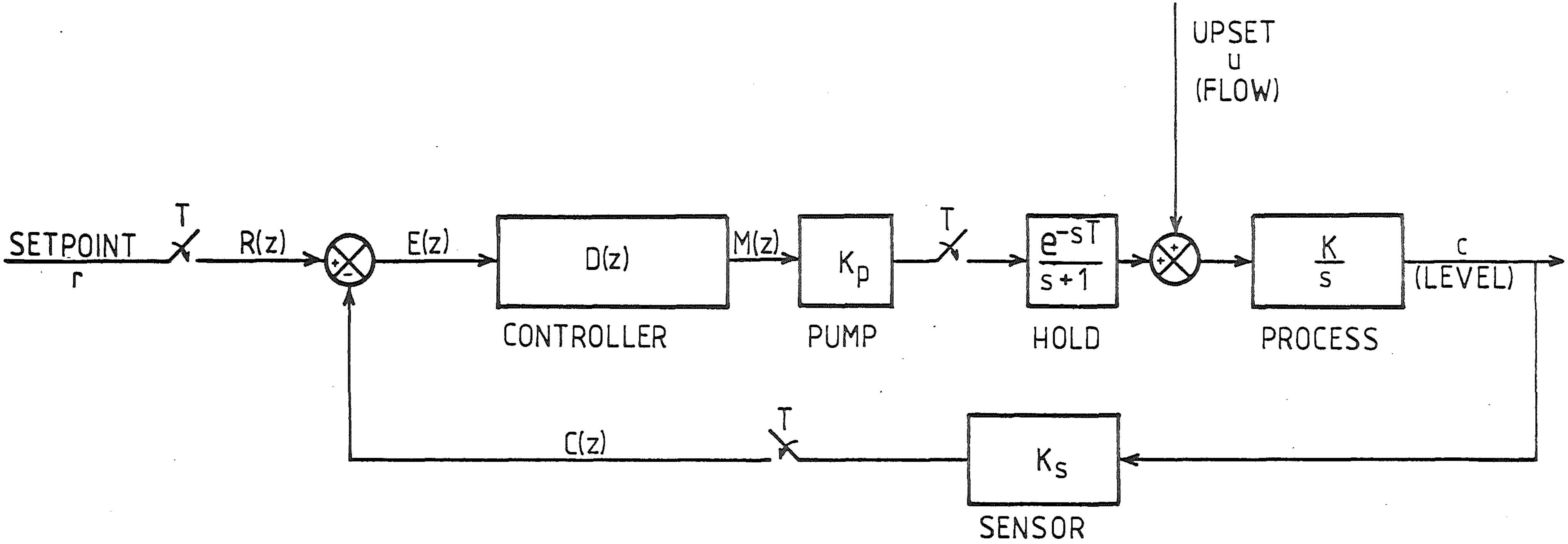


FIGURE 7-2 LEVEL CONTROL LOOP

ever, the control produced by the controller will be unrealisable because the controller requires control action greater than that possible. (The design method takes no account of limits on control action).

e.g. for the distillation column reflux accumulator

$$K = \frac{1}{\text{cross sectional area}} = 2.5 \times 10^{-3} \text{cm}^{-2}$$

$$K_S = - .0115\text{V cm}^{-1} \quad \text{sensor gain}$$

$$K_p = 200 \text{ cm}^3 \text{ min}^{-1} \text{ V}^{-1} \quad \text{control pump gain}$$

$$T = 0.167 \text{ min}$$

from equation (7.7)

$$\Delta m_n = - 2042 \, e_n + 1041 \, e_{n-1}$$

for a change in level of 1 cm

$$e_{n-1} = 0$$

$$e_n = 1 \times K_S$$

$$= - .0115\text{V}$$

$$\therefore \Delta m_n = 23.5\text{V}$$

but the maximum pump control signal is 10V.

There are three possible solutions to this situation:

(i) choose another response, C(z) , which gives a controller with a realisable control action;

(ii) use the controller structure as calculated, but de-tune it (i.e. reduce the gain);

(iii) increase the sample time until the control action is within the desired limits. A sample time of approximately 4 minutes is required to give an initial change in controller output of 1V for a step change in level of 1 cm.

The first case will produce a controller with more terms in it requiring more storage and more computation, while the third case underutilises the capacity of the control microcomputer and could suffer

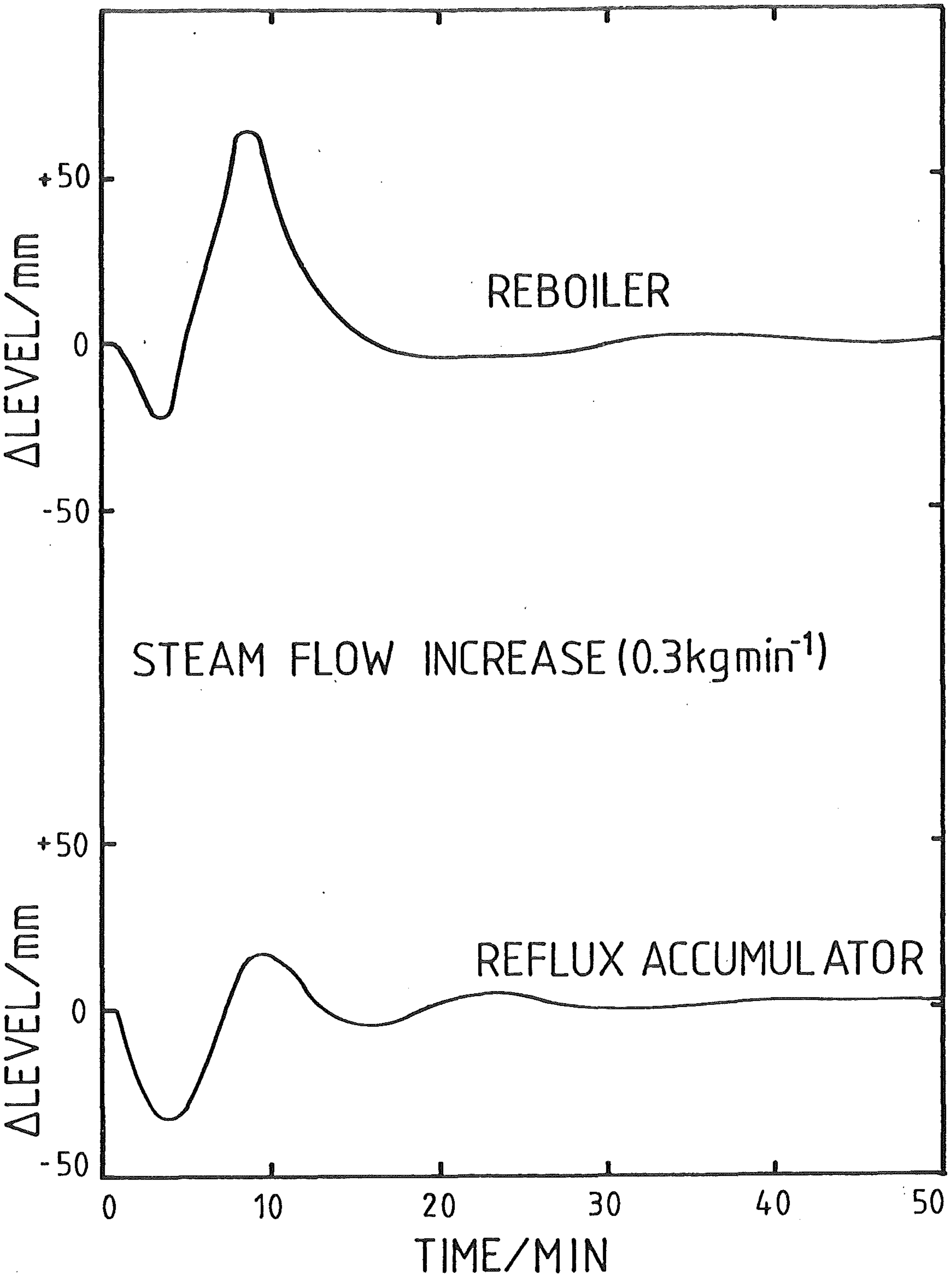


FIGURE 7-3 LEVEL RESPONSES

badly from upsets occurring in between the sampling intervals (if the sample period was 4 minutes). The second solution was chosen since it afforded the simplest controller. The level control loops were not critical to the overall column control, so the choice of controller was not critical.

7.2.2 Implementation of Level Controls

Two of the PI controllers described in section 5.9 were tuned to provide averaging level control in the distillation column. The controller equation used was

$$\Delta m_n = - 392 \; e_n + 368 \; e_{n-1} \qquad (7.8)$$

which is equivalent to a PI controller with

$$K_c = - 368\text{V/V}, \; T_I = 2.6 \text{ min} \;, \quad T = 10 \text{ seconds}$$

The sampling interval was chosen as 10 seconds because the dominant process time constants were of the order of several minutes; hence the digital controller closely approximated a continuous analog PI controller in performance.

Figure 7-3 shows typical closed loop responses of the level controllers to a step increase in steam flow. The responses were essentially second order with oscillations damping out to within ± 5 mm of the setpoint within fifteen minutes. The dynamics of these level control loops were included in the overall column dynamics described in the following sections.

7.3 COLUMN OPEN LOOP RESPONSES

A number of model identification procedures are available in the literature for both on-line and off-line use (Gustavsson (1975)), Box and Jenkins (1970), Saridis (1974), Sundaresan and Krishnaswamy (1978)). The main considerations for an identification procedure are:

(i) the model structure

(ii) the input to excite the plant dynamics

(iii) the analysis method

(iv) the validity and usefulness of the model produced.

The simplest model which adequately describes the process for any particular application should be used. In this application, simple models were required to predict the column response, and for use in designing feedback controllers.

For the purposes of determining the column dynamics, an off-line least squares approach was used to fit determinate discrete models, in the z domain, to step input responses. Such an approach was indicated because the column responses were essentially noise free, and in a digital control system, the inputs to the system would be discrete steps.

The four response variables were the distillate and bottoms product compositions, and the top (T_1) and bottom (T_8) tray temperatures. Each of the temperature probes was located close to the exit weir on that tray, and was expected to indicate the temperature of the liquid leaving the tray (and hence imply the composition of the liquid leaving the tray). The tray temperatures T_1 and T_8 were used as the control variables in the two composition control loops. The composition control loops were much slower than the temperature control loops because of the on-line refractometer sampling rate and the reboiler and reflux accumulator holdups. The upsets to the column were the feed variables: flow, composition, temperature, and the temperature control variables: reflux flow and steam flow.

7.3.1 Feed Upsets

Step upsets of varying magnitude and direction were made to investigate the column dynamics. Feed flow changes were made by altering the stroke of the feed pump, feed composition changes by adding pure methanol or water to the feed tank as required, and feed temperature changes by altering the steam pressure in the feed preheater. The reflux and steam flow changes were made using the manual controls. During the tests the reflux flow and steam flow were on manual control while the reflux accumulator level was controlled by the distillate flow, and the reboiler level by the bottoms flow. Figures 7-4 to 7-11 show typical column responses to step upsets, and the fitted models. The models fitted were discrete z plane polynomials of the form

$$\frac{\text{output }(z)}{\text{input }(z)} = \frac{(a_1 z^{-1} + a_2 z^{-2} + \ldots + z^{-m}) z^{-N}}{(1 + b_1 z^{-1} + b_2 z^{-2} \ldots + z^{-n})} \qquad (7.9)$$

where T = sample time (1 minute)

NT = dead time (min).

The dead time was chosen as a multiple of the sampling time to simplify model fitting. The model parameters were established for various model orders using a search technique based on the Hooke and Jeeves optimisation algorithm (Dixon (1972)). The PDP-11 minicomputer described in Chapter Four was used for this work. The fitted models produced estimates of the process variables at 1 minute intervals, but some of these estimates have been left off the figures for clarity.

The model form was chosen to be the simplest which adequately described the process response especially over the initial part of the response. All the responses to feed upsets were satisfactorily fitted with models of the form

$$\frac{y(z)}{x(z)} = \frac{K(1 - b)z^{-1} . z^{-N}}{1 - bz^{-1}} \qquad (7.10)$$

which is the discrete form of the classical first order system with dead time = NT and time constant = $- T/\ln b$. The fitted models are given in Tables 7-1, 7-2, 7-3.

There was very good agreement between the models identified for increases and decreases (of varying magnitude) in the feed variables. Significant differences might have been expected because of non-linearities in the column and changing flowrates causing changes in the liquid dynamics. The good agreement did show that a tuned controller should be equally effective in controlling upsets in either direction for any one of the feed variables.

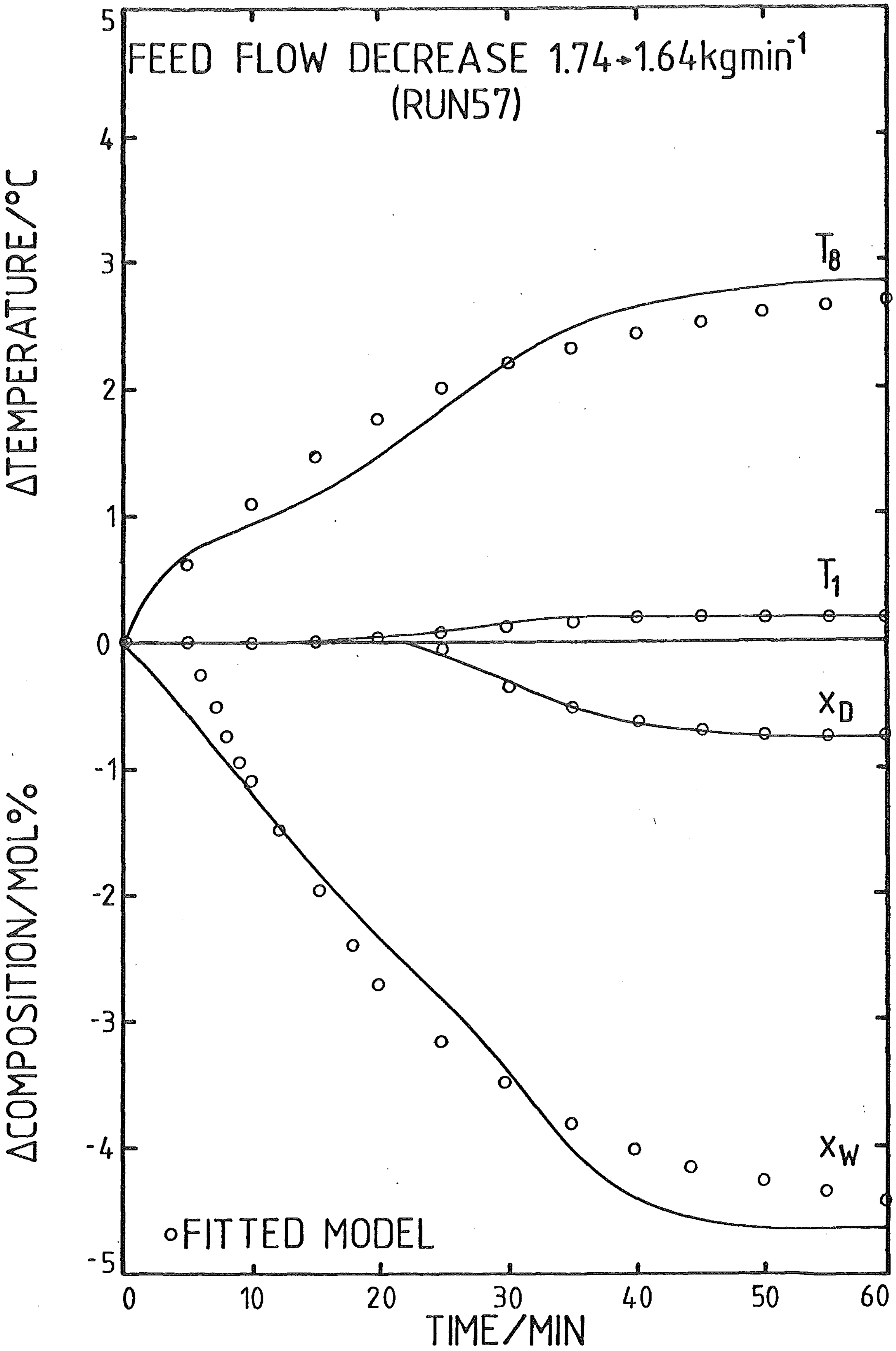


FIGURE 7-4 OPEN LOOP RESPONSE

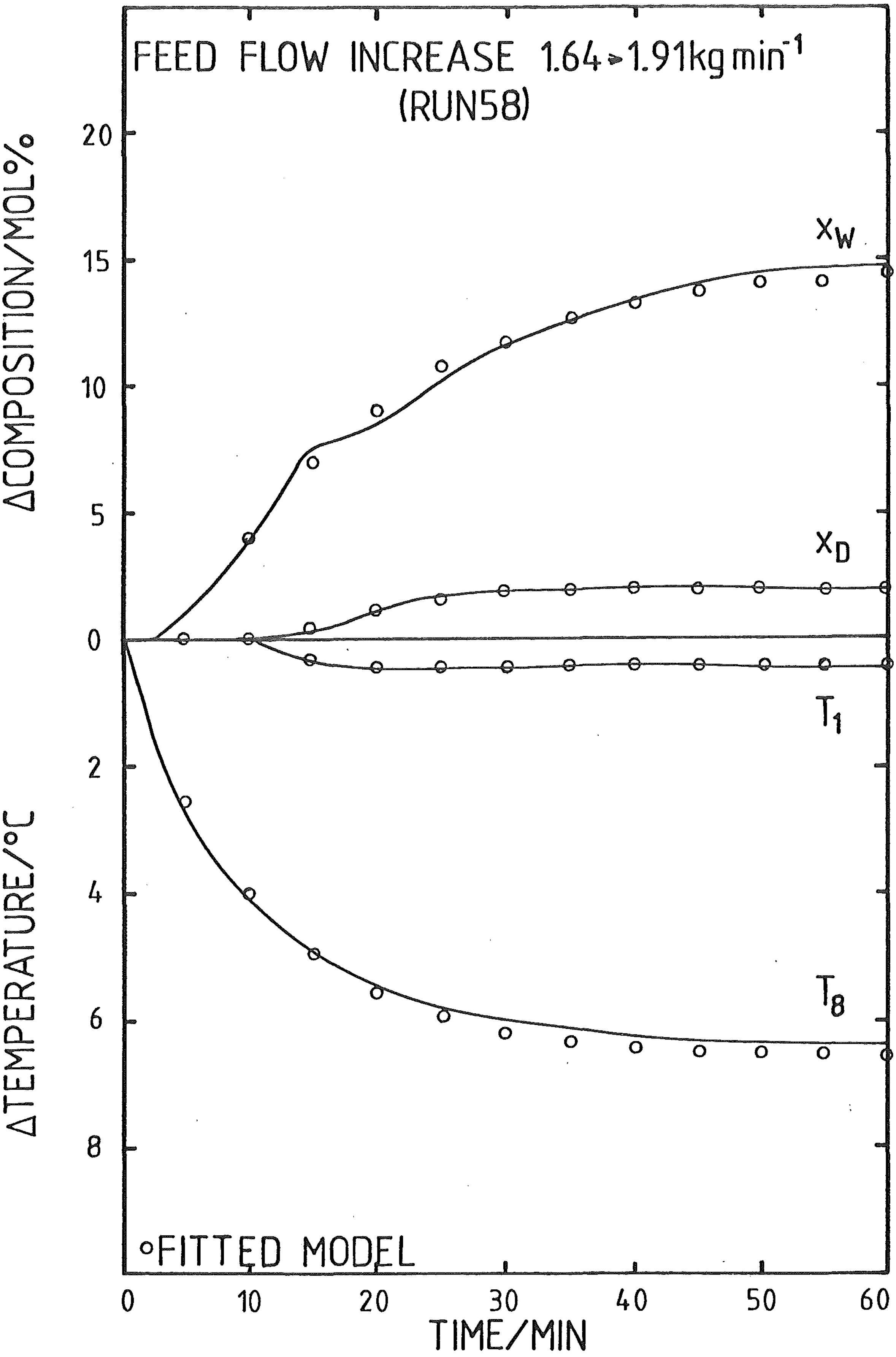


FIGURE 7-5 OPEN LOOP RESPONSE

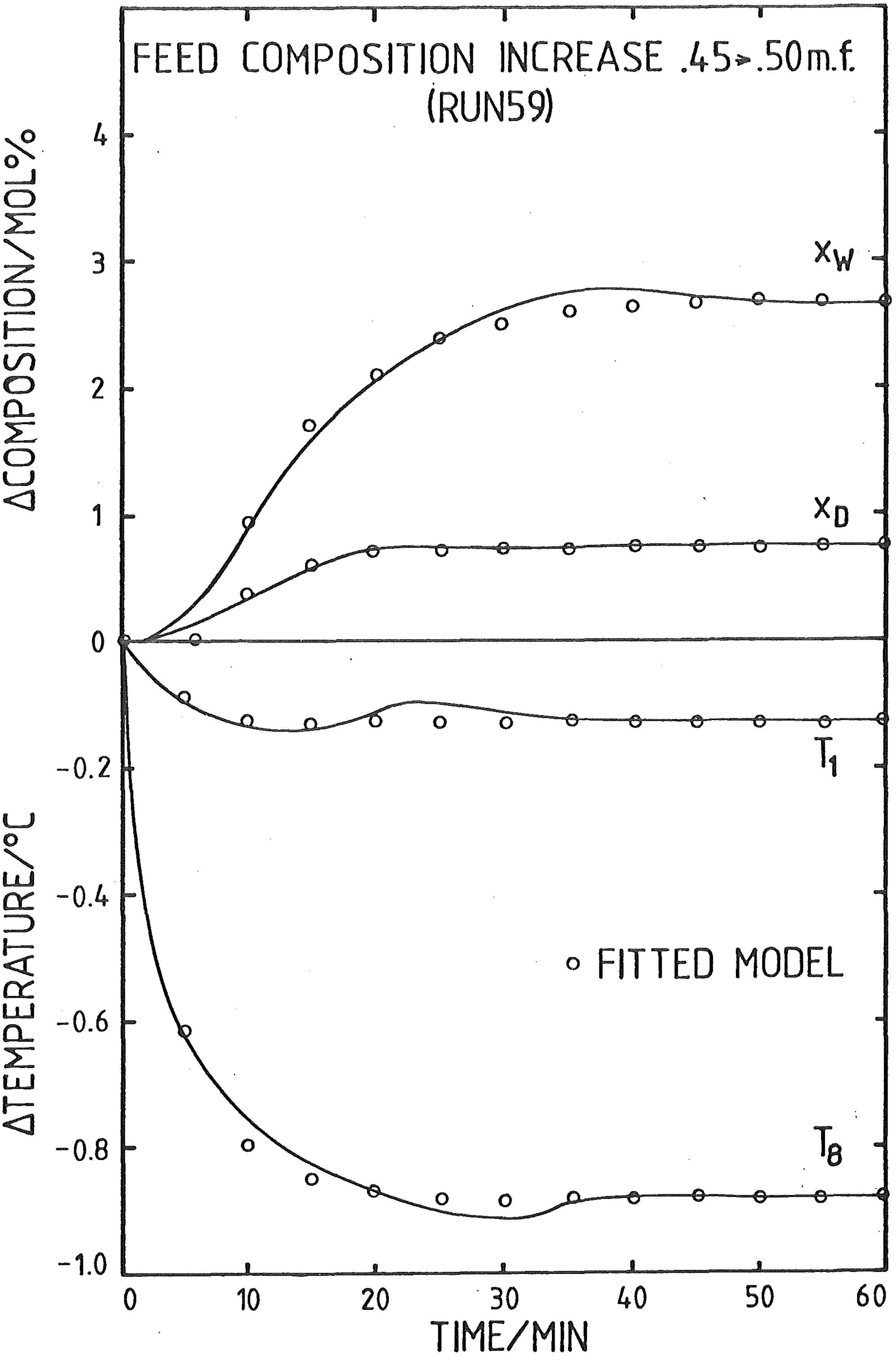


FIGURE 7-6 OPEN LOOP RESPONSE

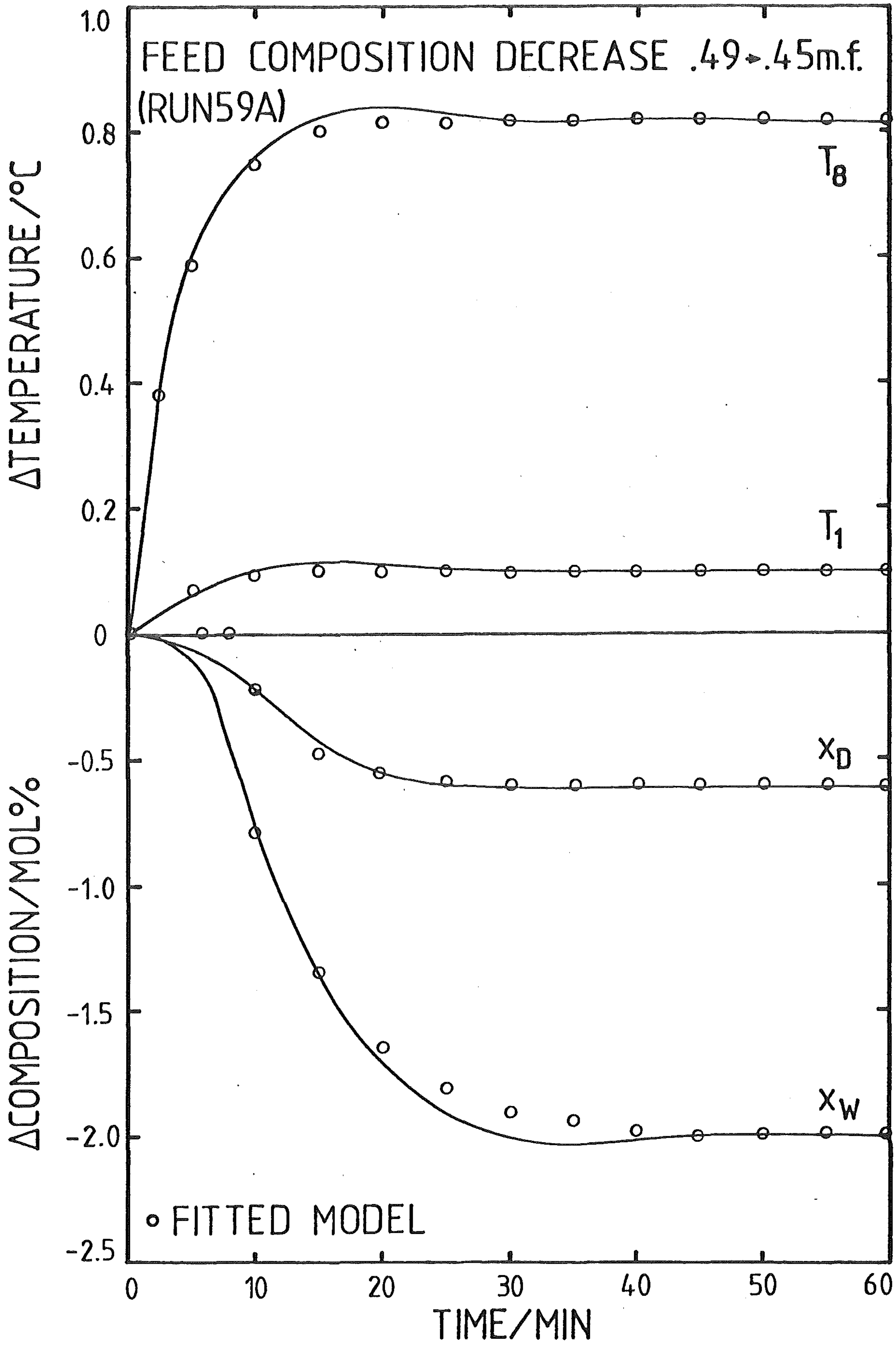


FIGURE 7-7 OPEN LOOP RESPONSE

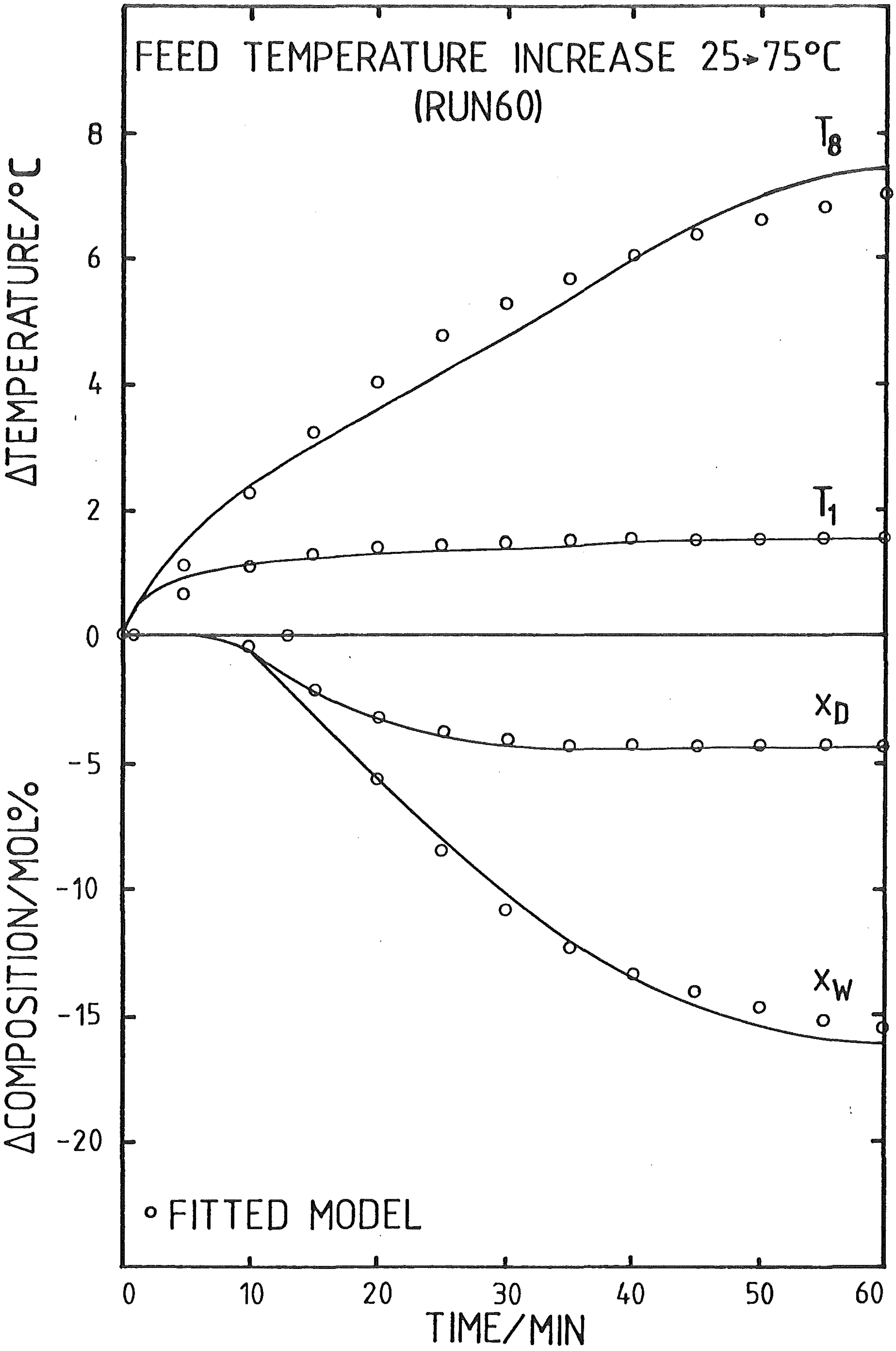


FIGURE 7-8 OPEN LOOP RESPONSE

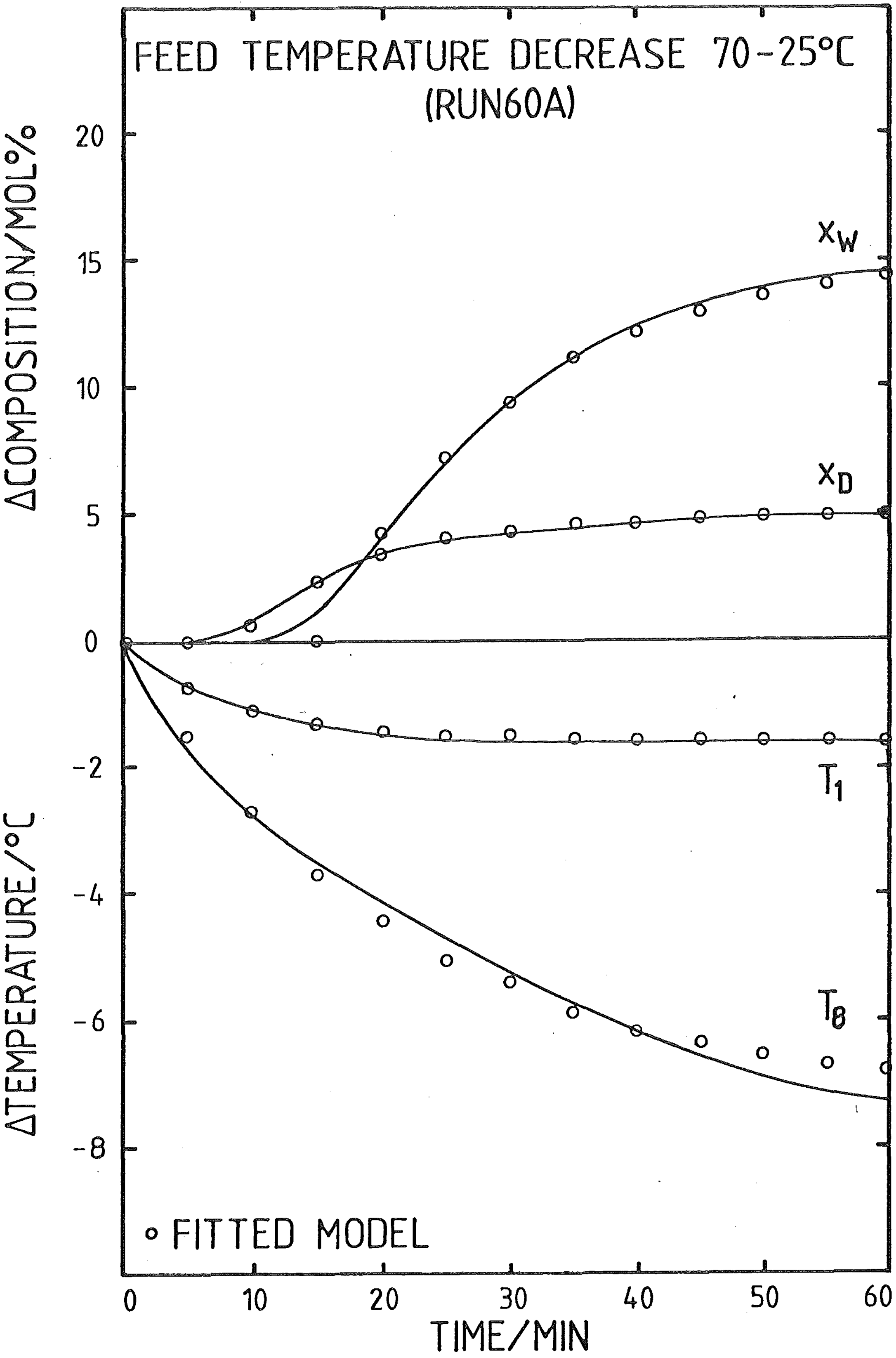


FIGURE 7-9 OPEN LOOP RESPONSE

TABLE 7-1

FEED RATE MODELS

RUN	UPSET	VARIABLE	FITTED DISCRETE TRANSFER FUNCTION T = 1 min	CONTINUOUS MODEL	
				τ/min	θ/min
R58	FEED RATE INCREASE (0.27 kg min^{-1})	x_D	13? $7.4\left(\dfrac{.120\,z^{-3}}{1-.880z^{-1}}\right)$	gam 7.8	12.0
		T_1	$-1.5\left(\dfrac{.231\,z^{-11}}{1-.769\,z^{-1}}\right)$	3.8	10.0
		x_W	$54.4\left(\dfrac{.061\,z^{-6}}{1-.939\,z^{-1}}\right)$	15.9	5.0
		T_8	$-24.4\left(\dfrac{.088\,z^{-1}}{1-.912\,z^{-1}}\right)$	10.9	0.0
R57	FEED RATE DECREASE (0.10 kg min^{-1})	x_D	$7.2\left(\dfrac{.106\,z^{-25}}{1-.894\,z^{-1}}\right)$	8.9	24.0
		T_1	$-1.6\left(\dfrac{.135\,z^{-19}}{1-.865\,z^{-1}}\right)$	6.9	18.0
		x_W	$46.0\left(\dfrac{.057\,z^{-6}}{1-.943\,z^{-1}}\right)$	17.1	5.0
		T_8	$-28.0\left(\dfrac{.048\,z^{-1}}{1-.952\,z^{-1}}\right)$	20.3	0.0

All compositions in mol% , all temperatures in °C

TABLE 7-2

FEED COMPOSITION MODELS

RUN	UPSET	VARIABLE	FITTED DISCRETE TRANSFER FUNCTION T = 1 Min	CONTINUOUS MODEL τ/min	θ/min
R59	FEED COMPOSITION CHANGE (+5 mol%)	x_D	$0.16 \left(\frac{.992\,z^{7}}{1-\,.808\,z^{-1}}\right)$	4.7	6.0
		T_1	$-\,.024 \left(\frac{.227\,z^{-1}}{1-.773\,z^{-1}}\right)$	3.9	0.0
		x_W	$0.54 \left(\frac{.107\,z^{-7}}{1-.893\,z^{-1}}\right)$	8.9	6.0
		T_8	$-\,.18 \left(\frac{.211\,z^{-1}}{1-\,.789\,z^{-1}}\right)$	4.2	0.0
R59A	FEED COMPOSITION CHANGE (-4 mol%)	x_D	$0.15 \left(\frac{.199\,z^{-9}}{1-\,.801\,z^{-1}}\right)$	4.5	8.0
		T_1	$-\,.025 \left(\frac{.237\,z^{-1}}{1-\,.763\,z^{-1}}\right)$	3.7	0.0
		x_W	$0.50 \left(\frac{.118\,z^{-7}}{1-\,.882\,z^{-1}}\right)$	8.0	6.0
		T_8	$-\,.21 \left(\frac{.221\,z^{-1}}{1-.779\,z^{-1}}\right)$	4.0	0.0

All compositions in mol%, all temperatures in °C.

TABLE 7-3

FEED TEMPERATURE MODELS

RUN	UPSET	VARIABLE	FITTED DISCRETE TRANSFER FUNCTION T = 1 min	CONTINUOUS MODEL	
				τ/min	θ/min
R60	FEED TEMPERATURE INCREASE (+45 °C)	x_D	$-.10\left(\frac{.105z^{-10}}{1-.895z^{-1}}\right)$	9.0	9.0
		T_1	$0.033\left(\frac{.125z^{-1}}{1-.875z^{-1}}\right)$	7.5	0.0
		x_W	$-.36\left(\frac{.060z^{-1}}{1-.940z^{-1}}\right)$	16.1	13.0
		T_8	$0.17\left(\frac{.041z^{-2}}{1-.959z^{-1}}\right)$	24.2	1.0
R60A	FEED TEMPERATURE DECREASE (-45 °C)	x_D	$-.11\left(\frac{.099z^{-1}}{1-.901z^{-1}}\right)$	9.6	9.0
		T_1	$0.036\left(\frac{.118z^{-1}}{1-.882z^{-1}}\right)$	8.0	0.0
		x_W	$-.33\left(\frac{.065z^{-12}}{1-.935z^{-1}}\right)$	15.0	11.0
		T_8	$0.16\left(\frac{.047z^{-1}}{1-.953z^{-1}}\right)$	20.8	0.0

All compositions in mol%, all temperatures in °C.

The column responses show some important points:

(i) The bottoms composition, and bottom tray temperature responses were significantly faster than the distillate composition and top tray temperature responses to feed disturbances. This was to be expected since the feed disturbances must, in general, travel down the column in the liquid stream and then up the column in the vapour stream. This effect indicated that the bottom temperature control loop would respond faster than the top temperature control loop to feed upsets.

(ii) The magnitudes of the bottoms composition and bottom tray temperature changes were greater than those of the distillate composition and top tray temperature for a given upset. This could be explained by the shape of the vapour/liquid equilibrium curve which was steep for low methanol concentrations and almost flat for high methanol compositions. Consequently a small change in the stripping operating line had a larger effect on the tray compositions than a corresponding change in the rectifying operating line.

(iii) The fit of the models was good for the initial part of the response which was the region in which a control system might be expected to operate. The latter parts of the response showed some higher order effects, and delayed dynamics (caused by interactions and reflections in the column) but the effects were not significant, e.g. the response of the bottoms variables in figures 7-4 and 7-6.

7.3.2 Control Variable Responses

The reflux flow (L_R) and steam flow (Q_S) were chosen as the control variables for the temperature control loops (see section 7.2). Step upsets in both directions were applied to these variables and the responses analysed. Typical results are shown in figures 7-10 to 7-13. The figures also show the fitted model responses (with some of the fitted model predictions left off for clarity). The models were fitted as described in section 7.3.1 and are listed in Tables 7-4 and 7-5.

The responses of the bottoms composition and bottom tray temperature to changes in steam flow were underdamped second order, and were fitted with second order models. However the initial response before the oscillations commenced can be reasonably approximated by a first order system and the two models are shown in Table 7-5. The responses show several important features.

(i) In most cases, there was no deadtime in the tray temperature responses to changes in reflux and steam flows. This indicated that the column should be relatively easy to control provided interaction was not great.

(ii) The response of distillate composition to changes in reflux and steam flows exhibited deadtime as might have been expected with the reflux accumulator in the loop.

(iii) The response of bottoms composition to changes in steam flow was fast considering the large holdup in the reboiler. This was due to the rapid change in vapour flow for a change in steam flow. Because the boiling point of the liquid in the reboiler was fixed by composition and pressure, any extra heat input was taken up by an instantaneous increase in the boilup rate. This was confirmed by the rapid change in the bottom tray temperature (time constant < 1 minute).

(iv) The response of the top tray temperature and distillate composition to steam flow changes showed two dynamic effects, one fast and one slower (figures 7-12, 7-13). Initially the change in steam flow produced a rapid change in vapour flow, which in turn rapidly altered the separation in the column as shown by the temperature changes in figures 7-12 and 7-13. The reflux accumulator contributed a lag between vapour composition changes and the corresponding reflux composition changes, consequently there was a lag before the changing reflux composition entered the column and caused further changes to the column separation. The result of these two systems was a response with an initial overshoot of the final steady state for the distillate composition and top tray temperature.

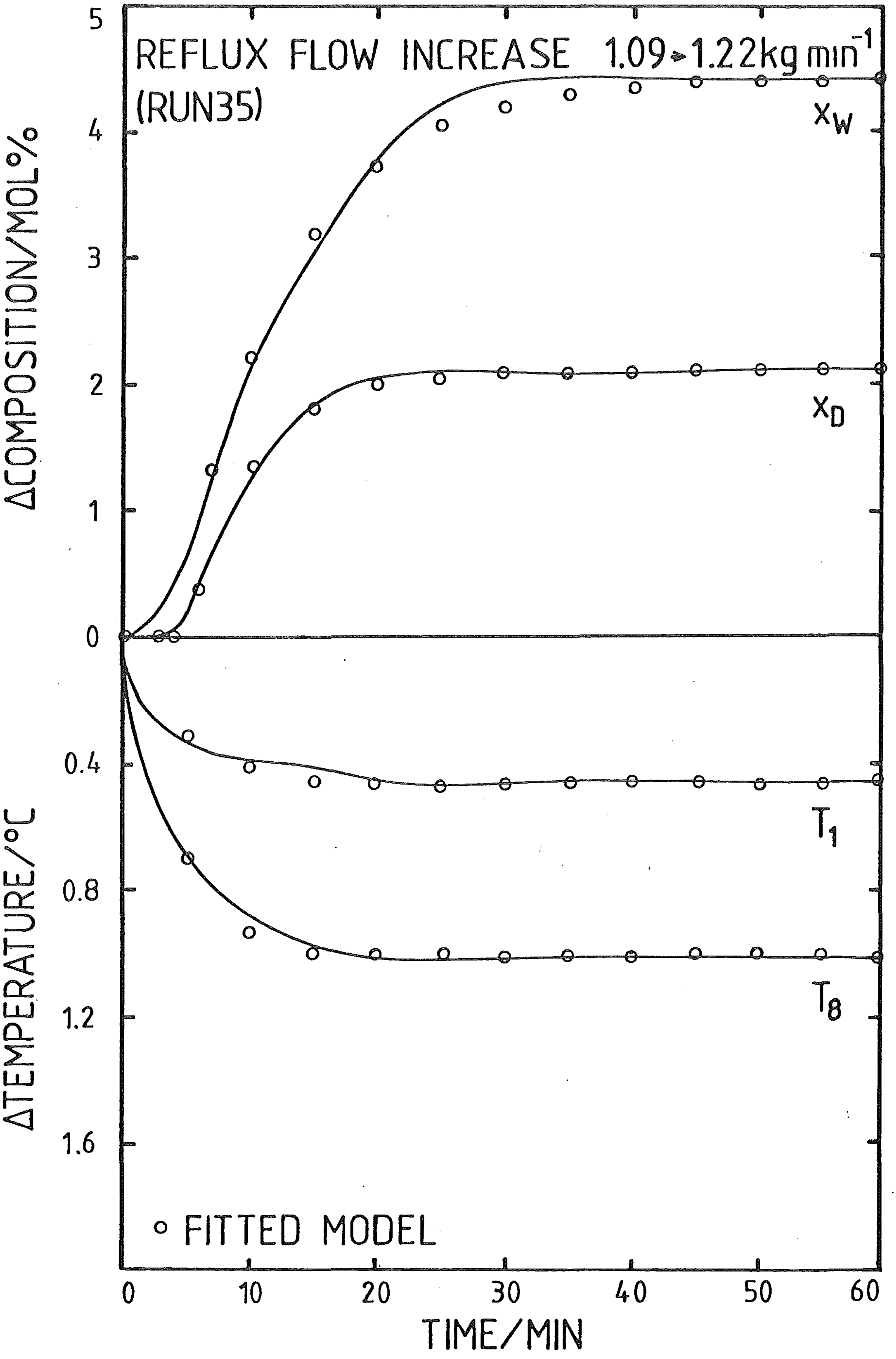


FIGURE 7-10 OPEN LOOP RESPONSE

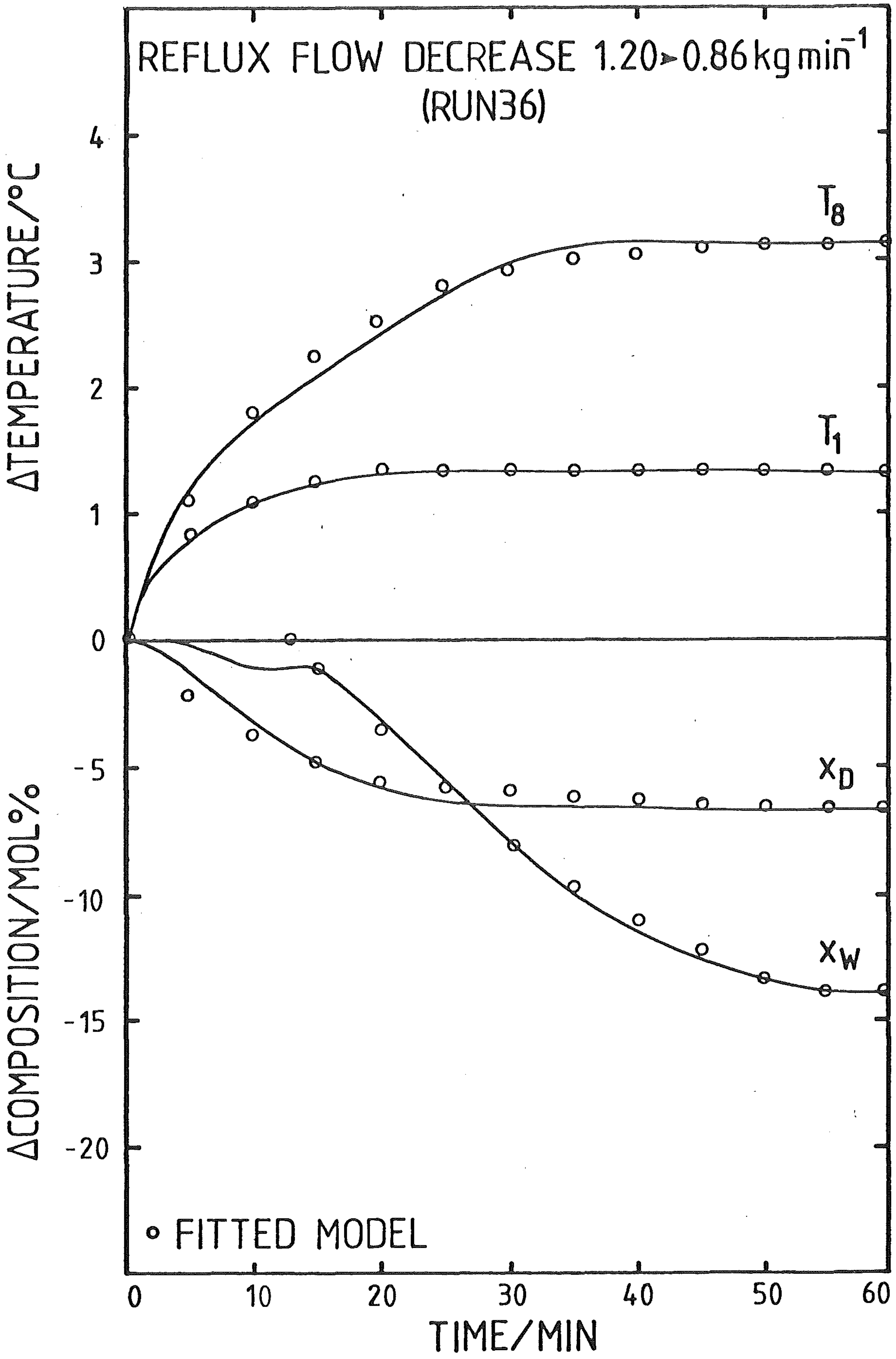


FIGURE 7-11 OPEN LOOP RESPONSE

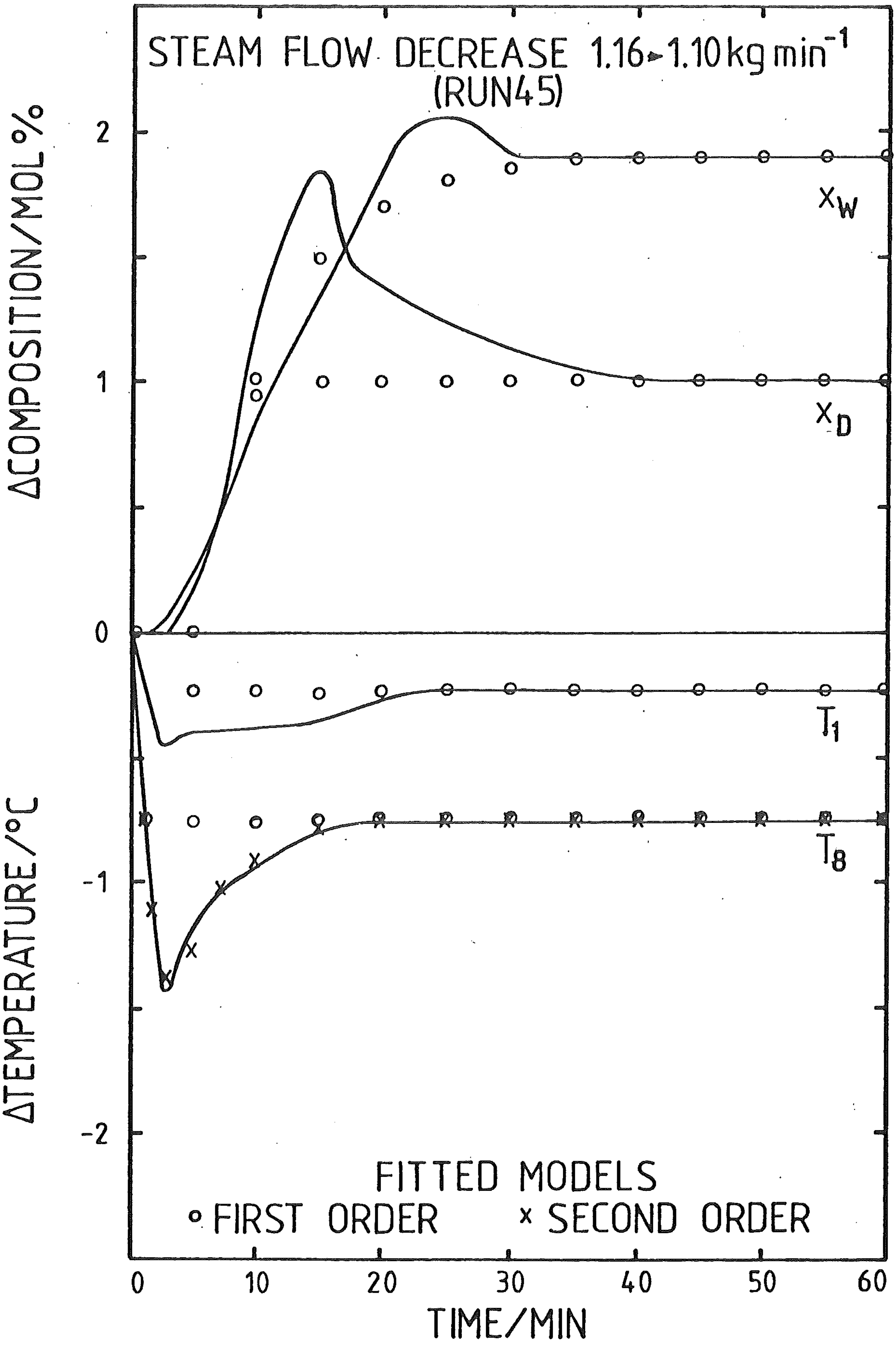


FIGURE 7-12 OPEN LOOP RESPONSE

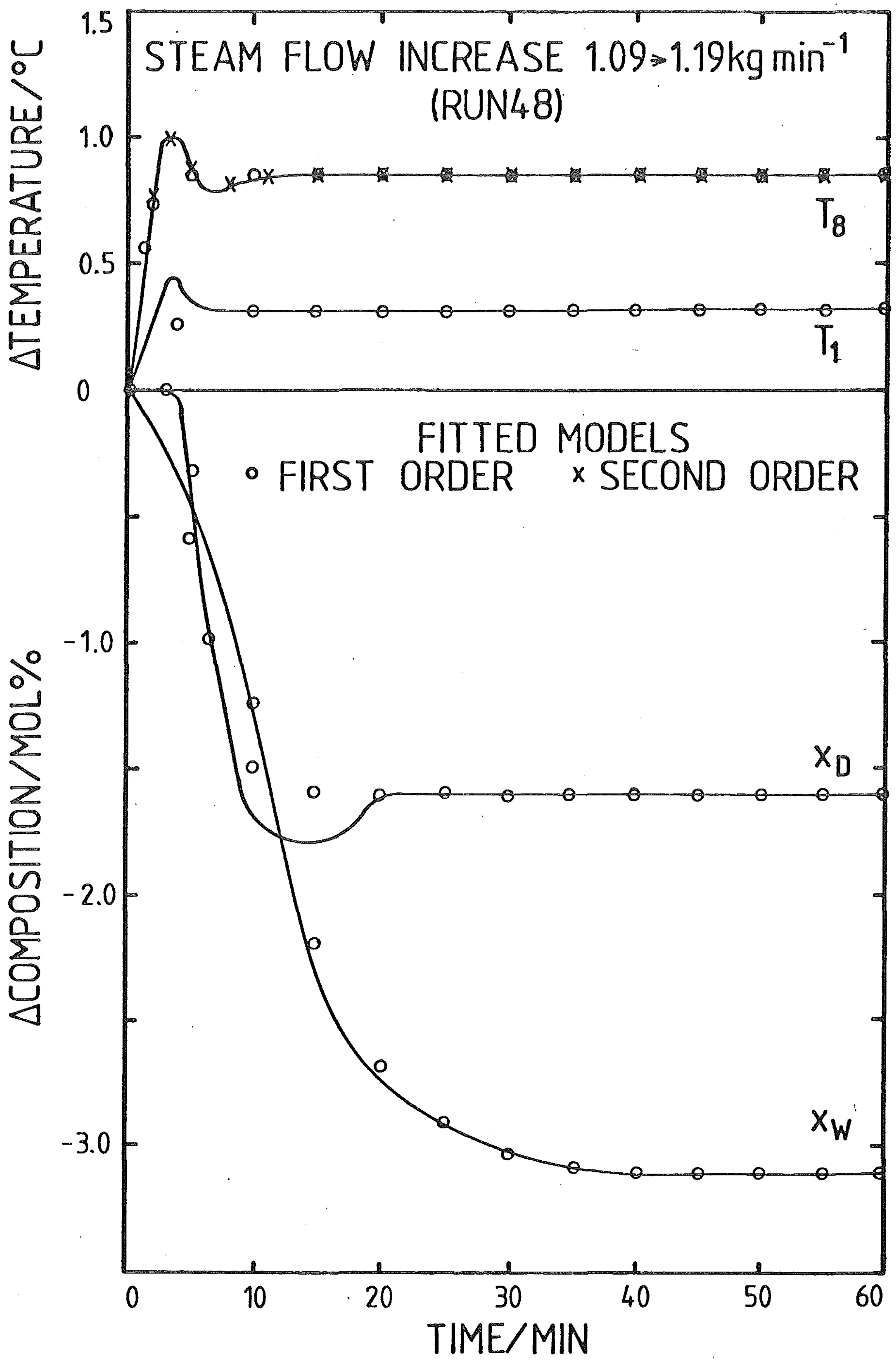


FIGURE 7-13 OPEN LOOP RESPONSE

TABLE 7-4

REFLUX FLOW MODELS

RUN	UPSET	VARIABLE	FITTED DISCRETE TRANSFER FUNCTION T = 1 min	CONTINUOUS MODEL τ/min	θ/min
R35	REFLUX FLOW INCREASE (+.13 kg min^{-1})	x_D	$17.4\left(\frac{.135z^{-1}}{1-.865z^{-1}}\right)$	6.9	0.0
		T_1	$-3.6\left(\frac{.190z^{-1}}{1-.810z^{-1}}\right)$	4.8	0.0
		x_W	$34.0\left(\frac{.112z^{-5}}{1-.888z^{-1}}\right)$	8.0	4.0
		T_8	$-8.2\left(\frac{.195z^{-1}}{1-.805z^{-1}}\right)$	4.6	0.0
R36	REFLUX FLOW DECREASE (-.34kg min^{-1})	x_D	$19.1\left(\frac{.086z^{-1}}{1-.914z^{-1}}\right)$	11.2	0.0
		T_1	$-4.0\left(\frac{.171z^{-1}}{1-.829z^{-1}}\right)$	5.3	0.0
		x_W	$41.2\left(\frac{.040z^{-14}}{1-.960z^{-1}}\right)$	24.5	13.0
		T_8	$-9.3\left(\frac{.081z^{-1}}{1-.919z^{-1}}\right)$	11.8	0.0

All compositions in mol%, all temperatures in °C

TABLE 7-5

STEAM FLOW MODELS

			FITTED DISCRETE	CONTINUOUS	
			TRANSFER FUNCTION	MODEL	
RUN	UPSET	VARIABLE	T = 1 min	τ/min	θ/min
R48	STEAM FLOW INCREASE (+.10 kg min^{-1})	x_D	$-16.2\left(\frac{.373z^{-5}}{1-.627z^{-1}}\right)$	2.1	4.0
		T_1	$4.0\left(\frac{.990z^{-2}}{1-.010z^{-1}}\right)$	0.2	1.0
		x_W	$-32.5\left(\frac{.094z^{-5}}{1-.906z^{-1}}\right)$	10.0	4.0
		T_8	$8.5\left(\frac{.377z^{-1}-.269z^{-2}}{1-.756z^{-1}+.401z^{-2}}\right)$	-	0.0
		T_8	$8.5\left(\frac{.614z^{-1}}{1-.386z^{-1}}\right)$	1.1	0.0
R45	STEAM FLOW DECREASE (-.06 kg min^{-1})	x_D	$-16.1\left(\frac{.499z^{-6}}{1-.501z^{-1}}\right)$	1.5	5.0
		T_1	$3.8\left(\frac{.835z^{-1}}{1-.165z^{-1}}\right)$	0.6	0.0
		x_W	$-31.8\left(\frac{.140z^{-6}}{1-.860z^{-1}}\right)$	6.6	5.0
		T_8	$12.5\left(\frac{1.145z^{-1}-1.044z^{-2}}{1-1.354z^{-1}+.456z^{-2}}\right)$	-	0.0
		T_8	$12.5\left(\frac{.991z^{-2}}{1-.009z^{-1}}\right)$	0.2	1.0

All compositions in mol% , all temperatures in °C

7.3.3 Temperature Control Interactions

The degree of interaction between the control loops in a distillation column can be estimated by the interaction measure (Bristol (1966), Shinskey (1967)) and the allocation of the control loops made so as to minimise the interaction. From the steady state gains of the transfer functions

$$\begin{bmatrix} T_1 \\ T_8 \end{bmatrix} = \begin{bmatrix} -4.0 & -3.8 \\ -9.3 & -12.5 \end{bmatrix} \begin{bmatrix} L_R \\ Q_S \end{bmatrix}$$

$$= P \begin{bmatrix} L_R \\ Q_S \end{bmatrix} \tag{7.11}$$

P = steady state process matrix

and the interaction measure is given by

$$M_{ij} = P_{ij} \, (P^{-1})_{ji} \tag{7.12}$$

$$M = \begin{bmatrix} 3.4 & -2.4 \\ -2.4 & 3.4 \end{bmatrix} \tag{7.13}$$

From the interaction measure the following points can be made:

(i) The controller pairings should be

T_1 controlled by L_R

T_8 controlled by Q_S

and that the opposite pairing will be unstable.

(ii) There will be significant interaction between the loops perhaps leading to instability.

(iii) The other control alternative using distillate and steam flow as the manipulated variables will give similar results to those above since the reflux ratio was close to one, and the reflux accumulator level control was tight.

However, the process matrices from other experimental runs produced the reverse controller pairings to those suggested above, and a very strong interaction between the two loops. The key to determining least interactive loops was the determinant of P . If the gains for the effects of the two inputs on any one output were close to being equal, i.e.

$$P = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad (7.14)$$

$$\det(P) = ad - bc$$

then any errors in measuring the gains a, b, c, d could cause the determinant to change sign and consequently the interaction measure would predict the opposite controller pairings. In the case of the process matrix being singular, Bristol's method predicts very difficult control.

Some of the process matrices found experimentally in this work were found to be almost singular, and hence predicted difficult control and the reverse controller pairing to that suggested by equation (7.13). e.g.

$$\begin{bmatrix} T_1 \\ T_8 \end{bmatrix} \quad \begin{bmatrix} -3.6 & -4.0 \\ -8.2 & -8.5 \end{bmatrix} \begin{bmatrix} L_R \\ Q_S \end{bmatrix}$$

gives

$$M = \begin{bmatrix} -13.9 & 14.9 \\ 14.9 & -13.9 \end{bmatrix}$$

The interaction measure takes no account of the process dynamics, and therefore cannot give a complete answer to the problem of configuring control loops. The controller pairings suggested by equation (7.13) were used because they were in agreement with the arguments presented in section 7.2.

7.3.4 Discussion

Simple models consisting of first order plus deadtime models were fitted in the z plane to the column responses for all upsets except the bottoms composition and bottom tray temperature responses to steam flow upsets. The models provided reasonable agreement even though other dynamics were present in some of the responses. These models were of the form used by Jafri et al (1965) and Meyer et al (1978,1979) on similar distillation columns. The transfer functions showed that the base of the column responded more quickly to upsets than did the top of the column. The transfer functions were used to investigate controller design for the temperature control loops.

7.4 TEMPERATURE CONTROLLERS

From the process dynamics identified by step testing, controller designs and tunings were investigated. The controllers considered were in the categories

(i) conventional analog controller replacements

(ii) specially designed sampled-data controllers.

There are a number of methods of tuning controllers in the first category (Smith (1972)) and the resulting settings can be fine tuned. The sampled-data controller could be quickly and easily designed for a specific upset but lacked flexibility in being able to handle other upsets.

7.4.1 Designing Sampled Data Digital Controllers

The methods used in section 7.2.1 for the level control system were applied to the temperature loops. The basic loop is shown in figure 7-14. The controller could be designed as a regulator to control the upsets, however, the problem was which upset to design for. The upsets all had different dynamics and required a different controller. The alternative was to consider the servo problem and design only for setpoint changes. Either type of controller (regulator or servo) was going to behave differently when subjected to the other

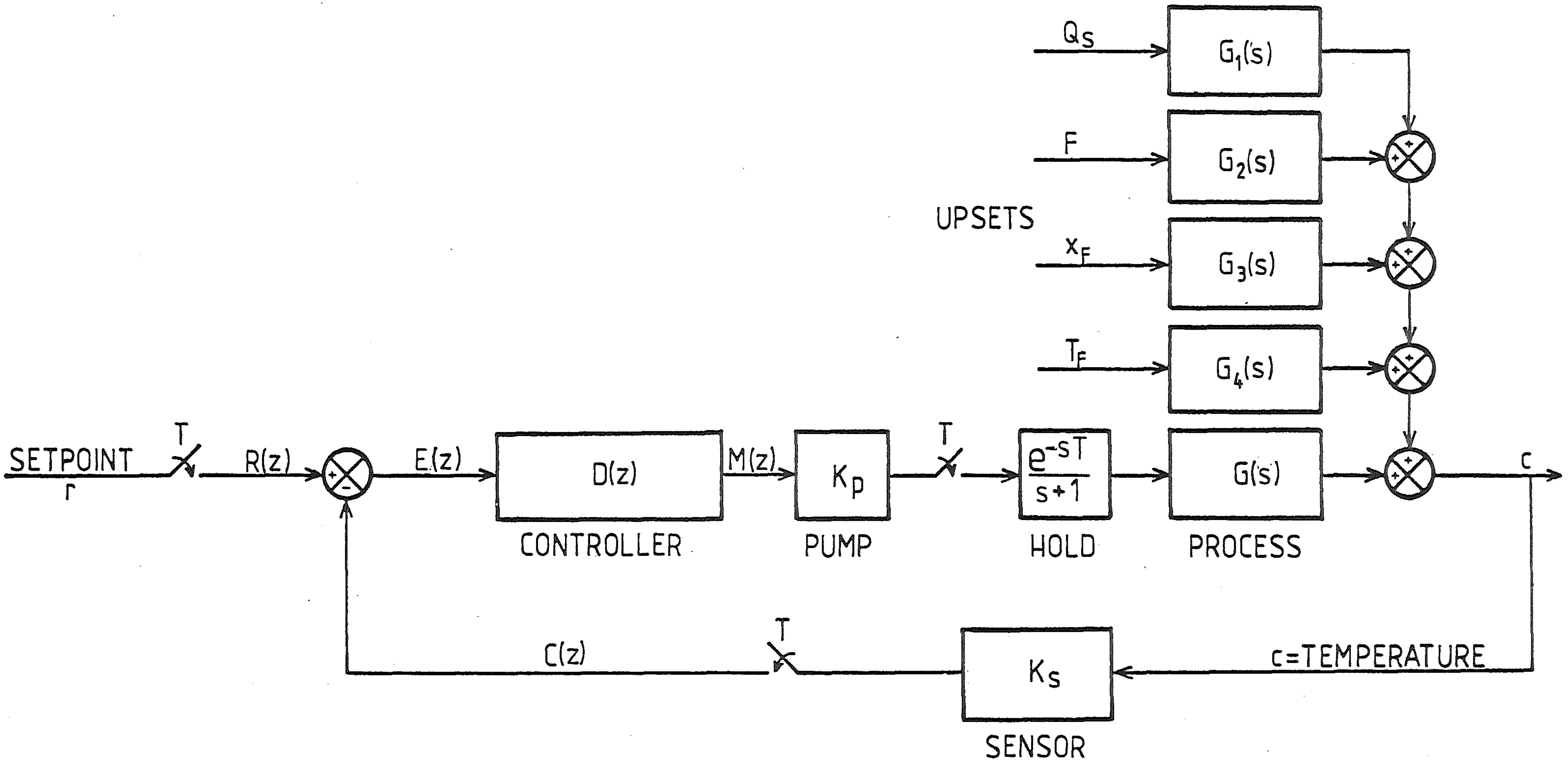


FIGURE 7-14 TEMPERATURE CONTROL LOOP

type of upset and the performance may not have been satisfactory. Two simple designs of the deadbeat type of controller were considered:

(i) Servo deadbeat controller.

For a unit step change in setpoint, the process was expected to reach the new setpoint in one sampling interval. The process was first order with time constant τ and a zero order hold was used. Hence (Smith 1972) :

$$R(z) = \frac{1}{1 - z^{-1}}$$

$$C(z) = \frac{z^{-1}}{1 - z^{-1}}$$

$$HG(z) = \frac{K(1 - b)z^{-1}}{1 - bz^{-1}}$$

$$b = e^{-T/\tau}$$

and

$$D(z) = \frac{1 - bz^{-1}}{K_s K_p K(1 - b)(1 - z^{-1})} \tag{7.15}$$

or

$$\Delta m_n = \frac{1}{K_s K_p K(1 - b)} (e_n - be_{n-1}) \tag{7.16}$$

which is the velocity type discrete equivalent of a PI controller.

(ii) Regulator deadbeat controller.

For a step upset, u , into $G_1(s)$, the process was required to return to the setpoint in two sampling intervals. The process was as for case (i). Hence

$$G_1(s) = \frac{K_1}{\tau_1 S + 1} \qquad \text{the upset dynamics}$$

$$NG(z) = \frac{K_1 u (1 - a)z^{-1}}{(1 - z^{-1})(1 - az^{-1})} \qquad a = e^{-T/\tau_1}$$

$$C(z) = K_1 U (1 - a)z^{-1}$$

$$D(z) = \frac{1}{K_p K_s K} \left[\frac{(1 - bz^{-1})(1 + a - az^{-1})}{(1 - z^{-1})(1 - az^{-1})}\right] \tag{7.17}$$

or

$$m_n = f(e_n, e_{n-1}, e_{n-2}, m_{n-1}, m_{n-2}) \tag{7.18}$$

and the velocity type form of the controller is

$$\Delta m_n = f(e_n, e_{n-1}, e_{n-2}, e_{n-3}, m_{n-1}, m_{n-2}, m_{n-3}) \tag{7.19}$$

There are obvious differences between the two controller designs (eqn. (7.16), (7.19)) but the performance of each would be similar. Shunta and Luyben (1972a, 1972b) investigated the application of similar controllers to a simulated process and concluded that the results were good for the disturbance for which they were designed but poor for other disturbances. In some cases the control was marginally better than conventional PI analog continuous control, but worse in others. They also designed a dual discrete controller which handled the load and setpoint upsets separately and which showed an improvement in control performance but was no better than conventional PI control. Setting up the dual controller was more difficult than a conventional controller, and required an accurate process model. Even incorporating decoupling into the dual controller showed only marginal improvement over conventional PI control. Hence the deadbeat type of controller was not considered in this project.

There are other problems associated with minimal deadbeat type controllers. Excessive control action can be demanded by the controller to provide the required response. This could be avoided by increasing the sample interval or incorporating constraints in another design procedure, e.g. the linear quadratic design procedures (Edgar et al (1973)). An alternate desired process response could be used to design another controller (Smith 1972), or additional poles and zeroes could be added to the controller to give the desired response. Both these methods require some experience on the part of the designer and could not be recommended as general practice.

Conclusion: While the synthesis of deadbeat type controllers is straightforward, the controller will in practice cause problems because of

(i) requiring excessive control action;

(ii) sensitivity to model parameter variations;

(iii) poor response to upsets other than that for which it was designed.

As expected, the discrete deadbeat controller for the servo problem proved to be a PI controller while the regulator problem produced a more complex controller. If the upset dynamics were similar to the plant dynamics ($G_1(s) \simeq G(s)$ in figure 7-14) then the regulator problem also reduced to a PI controller. On this basis, and considering that PI controllers were being used for the level controllers, discrete PI controllers were used on the temperature loops. The choice of one controller type also simplified progamming the microcomputer. As a first estimate the PI controllers were tuned using the tabulated values determined from the ITAE criterion (Smith (1972)). A deadtime equal to half the model sampling time was assumed in order to use this tuning method.

7.4.2 Closed Loop Column Responses

The recommended controller settings for the temperature control loops based on the ITAE criteria were:

T_1 loop $K_c = -107$ V/V $T_I = 1.6$ min $T = 10$s

$$\text{or} \quad \Delta m_n = -118.3\, e_n + 107 e_{n-1} \tag{7.20}$$

T_8 loop $K_c = +5.0$ V/V, $T_I = 0.8$ min $T = 10$s

$$\text{or} \quad \Delta m_n = +6\, e_n - 5\, e_{n-1} \tag{7.21}$$

The predicted controller tunings were tested on the column under closed loop control for a variety of feed upsets. Using the controllers described in section 5.8. Interactions between the loops made re-tuning necessary. The following controllers were found to give best overall control for a variety of feed upsets and setpoint changes.

T_1 loop $\quad K_c = -\ 2V/V \quad , T_I = 0.3$ min

$$\Delta m_n = -\ 3e_n + 2e_{n-1} \tag{7.22}$$

T_8 loop $\quad K_c = +\ 4V/V, \quad T_I = 0.8$ min

$$\Delta m_n = +\ 5e_n - 4e_{n-1} \tag{7.23}$$

For the T_1 loop , this represents a large reduction in gain, and an increase in integral action tending to a more loosely tuned loop. This change was made to slacken the control on the slower of the two loops and sacrifice some control performance for reduced interaction.

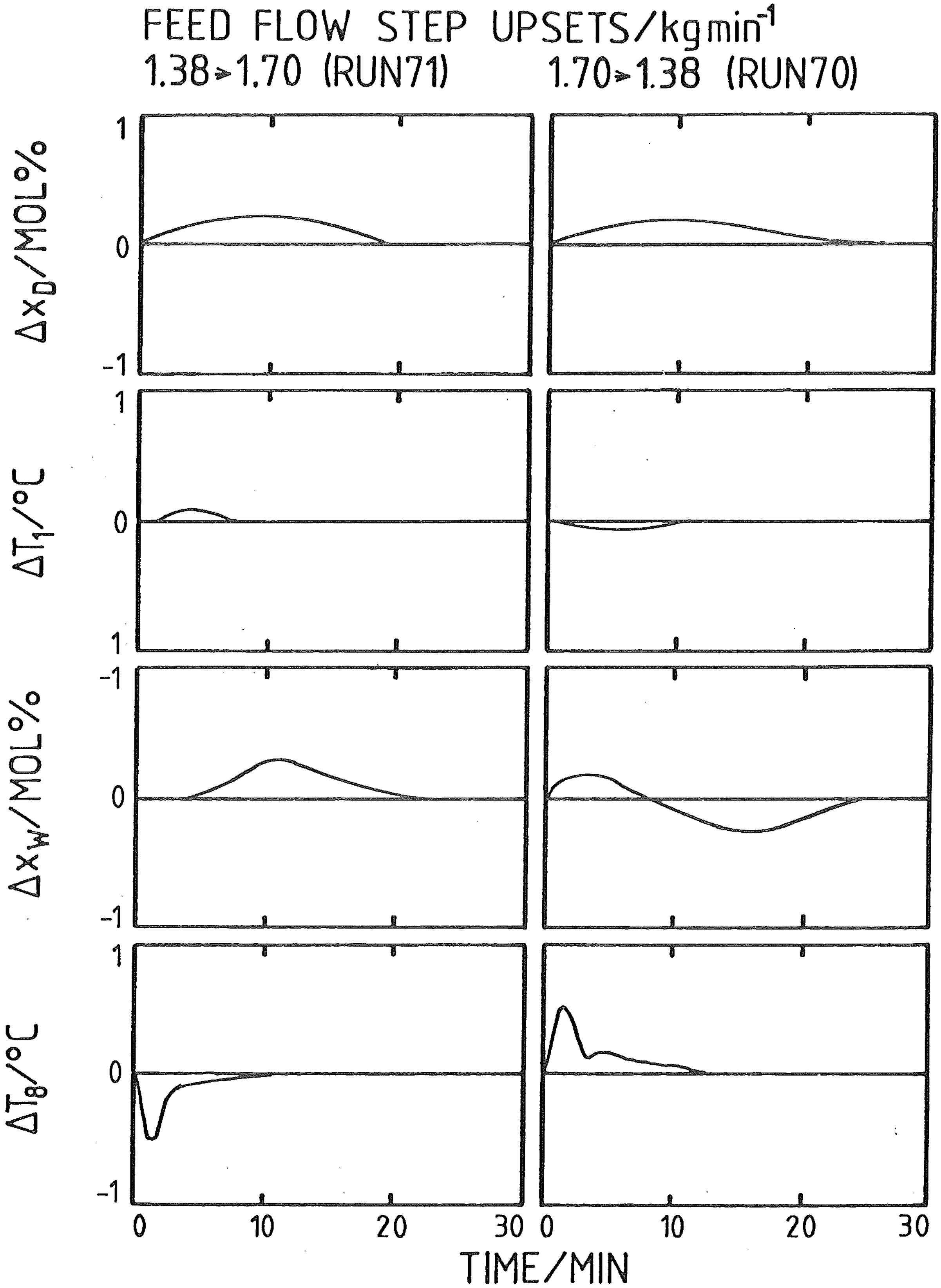
The results of these controllers to feed rate, feed composition and setpoint changes are shown in figures 7-15 to 7-18.

7.4.3 Discussion

The results in figures 7-15 to 7-18 show the responses of the column with temperature control on both ends of the column. The controllers used were the simplest possible - discrete PI eqivalents. The results therefore represented a basis for deciding whether significant improvement could be achieved by enhancing the control scheme, e.g. decoupling, feedforward control.

The precision of the measured deviations was limited by the performance of the temperature and composition sensors. These devices (see section 3.8) had a long term repeatability of ±0.2°C and ±0.4 mol% respectively, hence the deviations measured were well within the long term repeatability limits. For the short time duration (30 minutes) required for the closed loop responses, the stability of all measuring instruments was assumed to be sufficient to produce the results shown.

The choice of what constitutes good control in this situation is partly subjective. The commonly used criteria are minimum overshoot, settling time, rise time, decay ratio and the error integrals. The choice of one or more of these criteria depends on the type of response exhibited by the controlled loop (oscillatory or exponential) and by the requirements of the control loop (to minmise flow variations to other processing

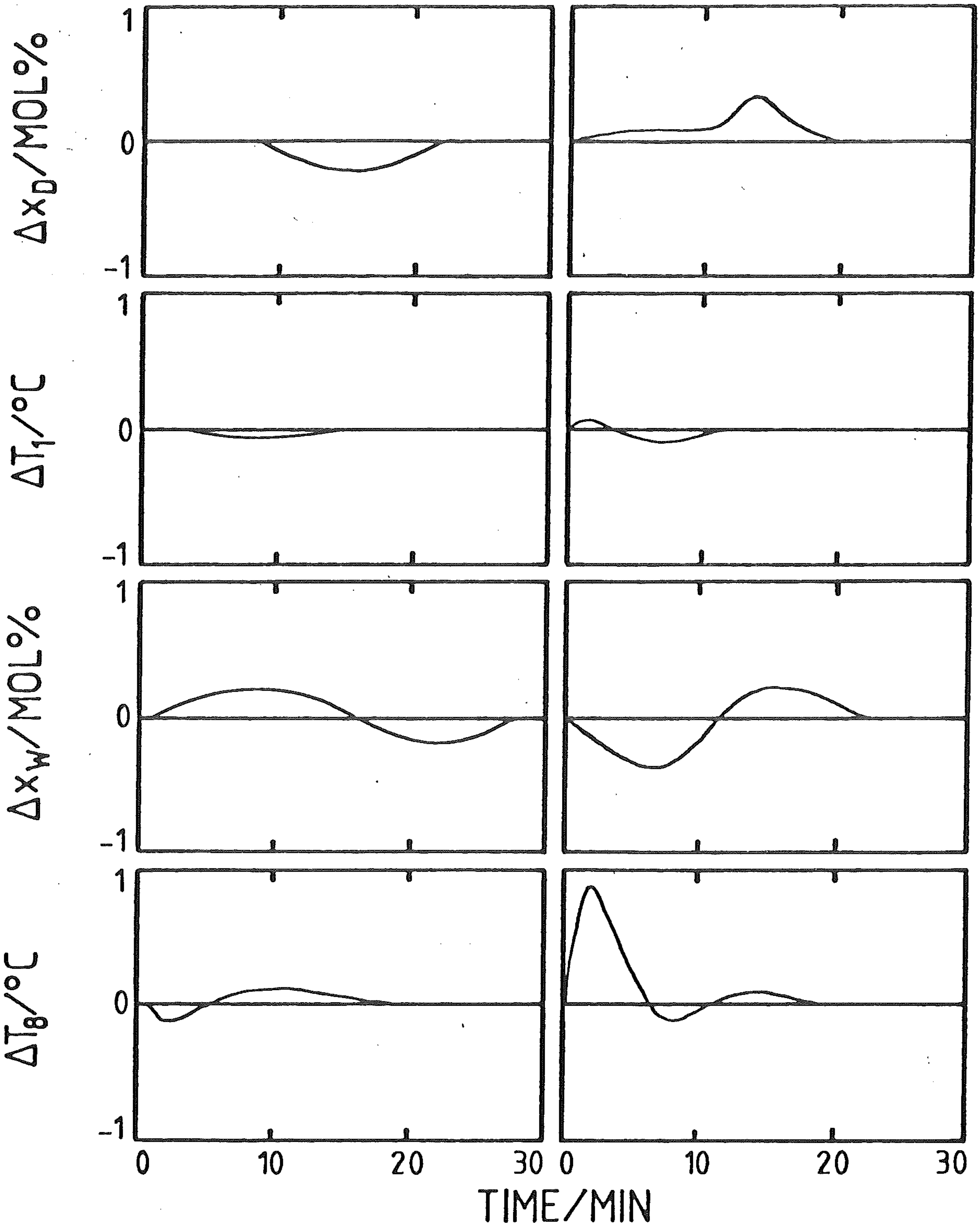


FEEDBACK CONTROLLERS

FIGURE 7-15 CLOSED LOOP RESPONSE

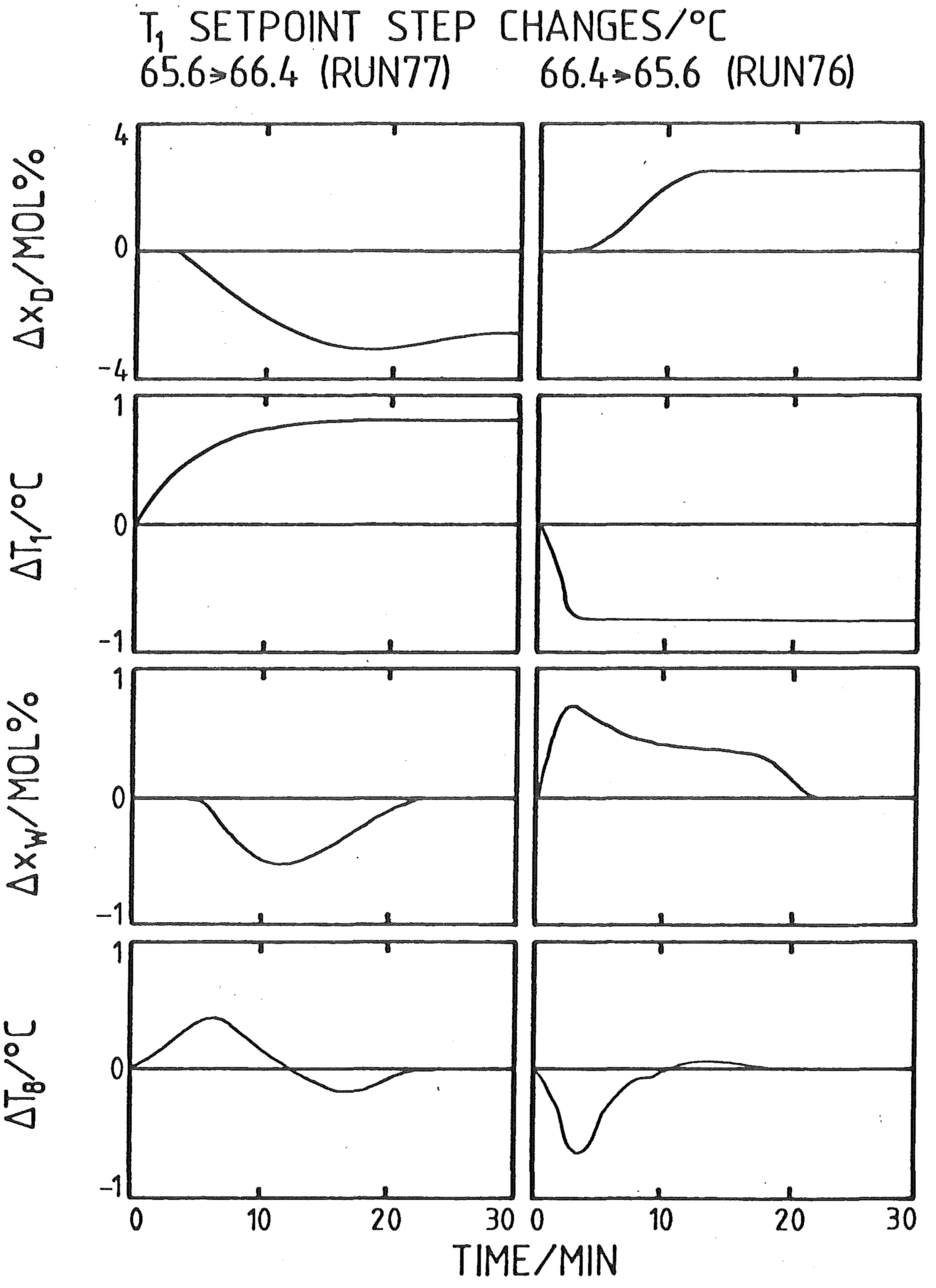
FEED COMPOSITION STEP UPSETS/m.f.

0.42→0.45 (RUN94) 0.46→0.42 (RUN93)



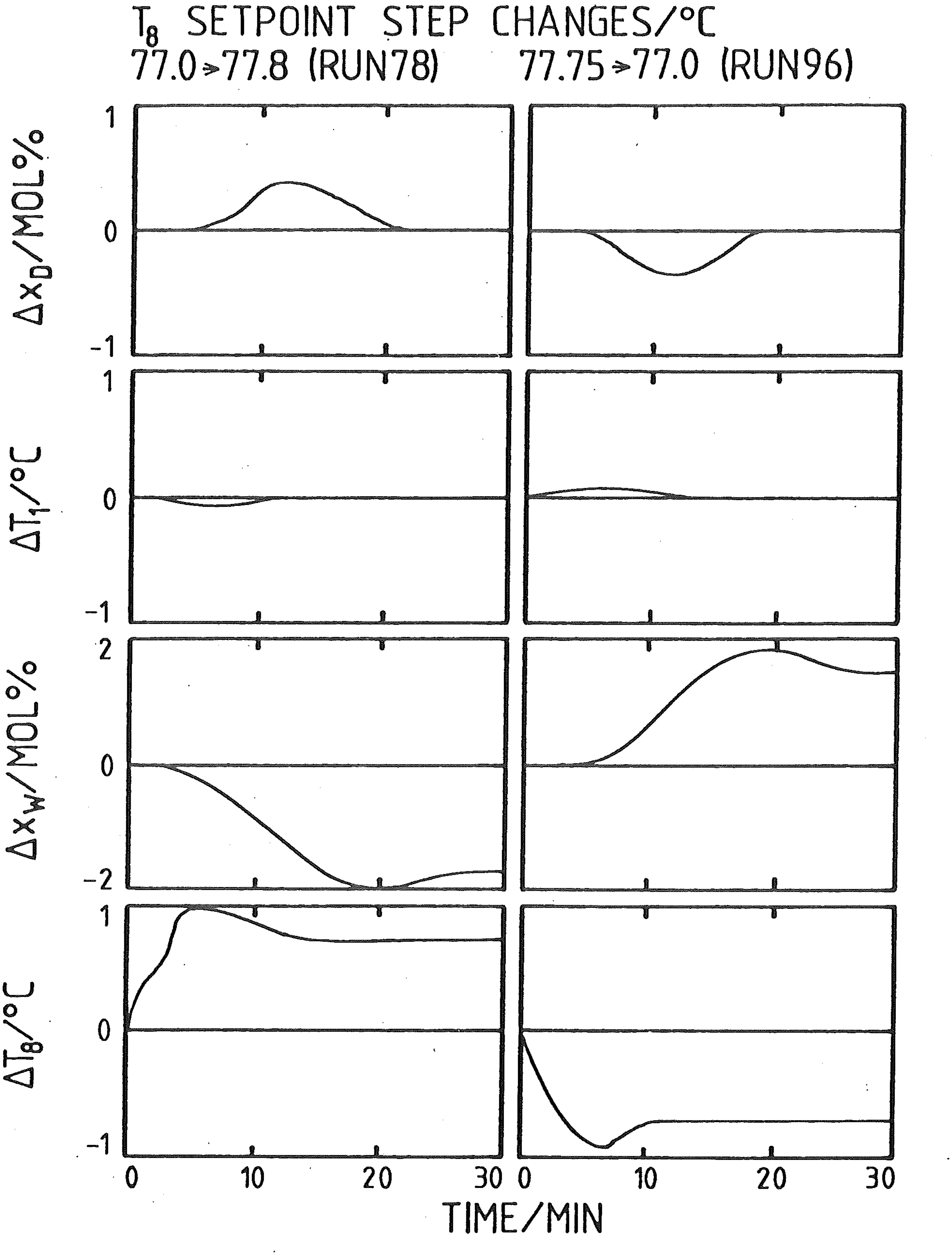
FEEDBACK CONTROLLERS

FIGURE 7-16 CLOSED LOOP RESPONSE



FEEDBACK CONTROLLERS

FIGURE 7-17 CLOSED LOOP RESPONSE



FEEDBACK CONTROLLERS

FIGURE 7-18 CLOSED LOOP RESPONSE

equipment, or to minimise off specification products). There is a point beyond which it is unjustified to try and make improvements to the performance of a control system.

The results shown for the distillation column under multiloop feedback control produced a maximum product deviation of 0.4 mol% for feed upsets and 0.7 mol% for setpoint changes. The product compositions were generally quick to respond but exhibited a long settling time. However for most of the settling time, the deviations were less than 0.2 mol%. This performance was considered excellent when compared to the results obtained by Svrcek (1967), Meyer et al (1978, 1979), Wood and Berry (1973) and Jafri et al (1965) on similar distillation columns.

As predicted in section 7.3.3, there was interaction between the two control loops causing some oscillation and overshoot at the bottom of the column. The final controller settings used were a compromise to give good control for a variety of upsets. The detuning of the top tray temperature controller resulted in a slower distillate composition response with correspondingly less interaction between the two loops. This is one way of overcoming the interaction problem.

The initial controller tuning for the two loops relied on the arbitrary selection of a deadtime equal to half the sample time (1 minute) for the fitted model. This deadtime along with the experimentally determined constant was used to set the controller parameters based on the ITAE method (Smith (1972)). The choice of deadtime was important, because as the ratio of deadtime to time constant tends to 0 , the controller gain tends to infinity and the integral time tends to 0 (infinite integral action). The initial choice of controller settings for the bottom tray temperature loop proved to be satisfactory, but those for top tray temperature loop were inadequate for the reasons previously mentioned.

The temperature controllers could be further fine tuned to respond better to a specific upset, e.g. feed rate changes only. However, it was important that the control system be robust enough to handle all possible upsets, and the trade off for this generality was a poorer controlled response in some situations.

The choice of temperature control tray location can have significant effects on the closed loop performance. Ideally, the sensors should be located at either end of the column for a constant pressure binary system but consideration of the ratios $\Delta T_1/\Delta x_D$ and $\Delta T_8/\Delta x_W$ (the sensitivity of temperature changes with respect to composition changes) could require that the sensing trays be located further into the column. The dynamic effects of the location of the sensing trays has been studied by Shunta and Luyben (1971) and Wahl and Harriot (1970). For this distillation column operating around x_D = 90 mol%, x_W = 5 mol%

$$\frac{\Delta T_1}{\Delta x_D} = 0.32 \frac{°C}{mol\%} \qquad \frac{\Delta T_8}{\Delta x_W} = 0.48 \frac{°C}{mol\%}$$

and the resolution of the temperature sensors was sufficient so that the control loops could maintain the product compositions within acceptable limits. Further, if the sensing elements were moved away from the ends of the column, the relationship between the temperature measured and the product composition became complex, and offsets occurred. Consider using the top and bottom tray temperatures for composition control:

(i) x_D control:

T_1 = temperature of liquid leaving tray 1

= $f(x_1)$

x_D = y_1 at steady state and hence

T_1 = $f(x_D)$ where the function involves the temperature/composition function, and the vapour/liquid equilibrium function only. If the efficiency of tray 1 is < 100%, then tray efficiency can be a factor in the function.

(ii) x_W control:

Consider the reboiler as in figure 7-19

$$L_8 x_8 = V_9 y_9 + Wx_W$$

$$y_9 = K x_W \quad \text{for} \quad x_W \to 0$$

$$x_8 = \left(\frac{V_9 . K + W}{L_8}\right) x_W$$

and $T_8 = f(x_8)$ (the temperature/composition function)

$$\therefore \quad x_W = f\ (L_8, V_9, W, K, T_8)$$

Consequently, any flow changes in the column will alter the relationship between bottoms composition and bottom tray temperature.

For small upsets to the column (±20% in feed rate, ±10% in feed composition), the offsets due to the temperature/composition functions described above were negligible; for larger upsets, trimming of the temperature loop setpoints was required to maintain the required product compositions. This could have been provided by cascade feedback loops using low gain integrating controllers as in figure 7-20. For these reasons trays 1 and 8 were chosen as the sensor trays.

The multi-loop control scheme could have been enhanced in several ways. The addition of derivative action to the controllers would have been the simplest improvement. There was however a significant amount of noise on the T_8 signal (±0.2°C) which would have caused problems with derivative action, and required the use of both hardware and software filters. The added complexity to the microcomputer software which used 16 bit integer arithmetic would not have produced significantly better control than that shown in figures 7-15 to 7-18.

A decoupler could have been used to separate the actions of the two controllers. Luyben (1970) showed the application of a simple decoupler on a binary distillation column produced improvements in the quality of control, but the improvements were not great, and the resulting transients were of the same nature and size as those obtained in

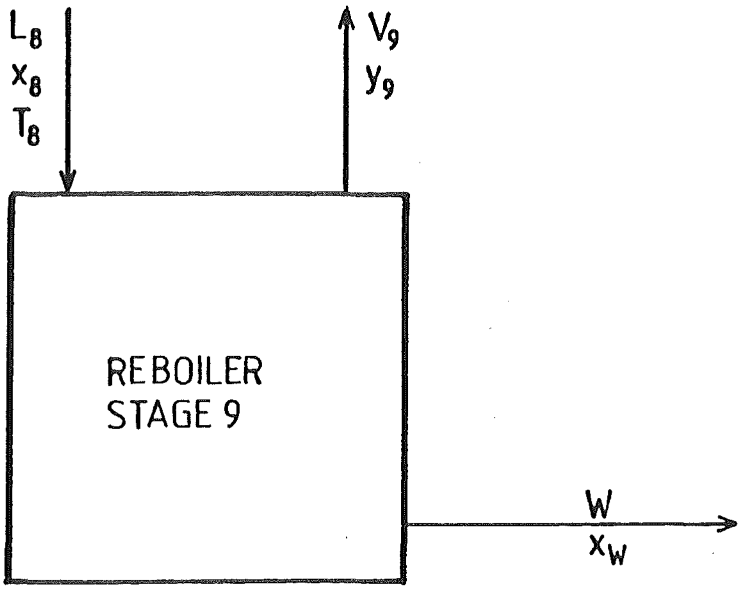
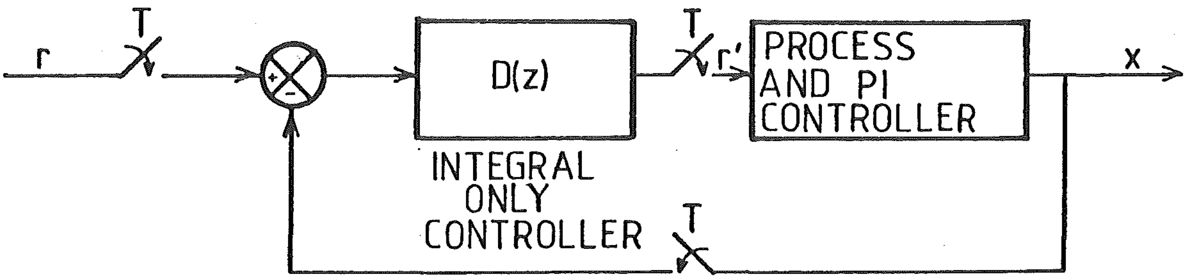


FIGURE 7-19 REBOILER FLOWS



r – COMPOSITION SETPOINT

r' – TEMPERATURE SETPOINT

x – PRODUCT COMPOSITION

FIGURE 7-20 CASCADE COMPOSITION CONTROL LOOP

this work with purely feedback control. The implementation of a decoupler would have required the use of lead/lag units, and caused problems with insignificant precision of the 16 bit integer arithmetic used. The improvements to be gained from the use of a decoupler were too small to warrant its inclusion.

Sastry et al (1977) have investigated a self-tuning controller, and Meyer et al (1978,1979) have investigated time delay compensation as improvements to conventional continuous PI control. In both cases, the product compositions were used as the control variables, and hence the PI controller performance was found to be inferior. Comparison of these results with those presented in this work show little or no improvement with the more sophisticated controllers.

7.5 CONCLUSION

The dynamics of a pilot scale atmospheric pressure binary distillation column have been investigated in both open loop and closed loop modes. The open loop responses have been approximated by simple discrete models (first or second order plus deadtime). Deadbeat discrete controllers based on the open loop models were investigated and found to be unsatisfactory, exhibiting poor control and requiring excessive control action. Discrete velocity-type PI controllers when correctly tuned produced acceptable product composition control when maintaining constant temperatures on the top and bottom trays in the column. The results obtained compared very favourably with other published work using more sophisticated controllers on similar columns (decoupling, sampled data, deadtime compensation). Interaction between the two temperature control loops was reduced by detuning the slower top tray temperature control loop.

The control achieved using simple digital controllers in a small microcomputer indicated that a dedicated distillation column controller could be produced at a competitive price compared with conventional analog controllers (see section 5.13). This control system could also be integrated into a much larger control scheme using a hierachy of control computers.

7.6 NOMENCLATURE

a	-	e^{-T/τ_1}
b	-	$e^{-T/\tau}$
c	-	process variable
C(z)	-	discrete process response
D(z)	-	discrete controller
e	-	continuous controller error = r-c
E(z)	-	error in the discrete controller
HG(z)	-	process pulse transfer function
K	-	process gain
K_1	-	upset gain
K_c	-	continuous controller gain
K_p	-	pump gain
K_s	-	sensor gain
L	-	liquid flow, mol min^{-1}
L_R	-	reflux flow, $\ell\,min^{-1}$
m	-	pump speed (≡ valve position)
M	-	interaction measure
M(z)	-	discrete controller output
N	-	dead time in number of sample intervals
NG(z)	-	disturbance pulse transfer function
P	-	process matrix
PI	-	proportion plus integral
Q_s	-	steam flow, kg min^{-1}
r	-	setpoint
s	-	Laplace domain operator
T	-	sampling interval

T_I	-	integral time
T_1,T_8	-	top and bottom tray temperature, °C
u	-	upset
V	-	vapour flow, mol min^{-1}
x	-	liquid composition
x_D	-	distillate composition
x_W	-	bottoms composition
y	-	vapour composition
z	-	z domain operator
θ	-	deadtime
τ	-	process time constant
τ_1	-	upset time constant

CHAPTER EIGHT

FEEDFORWARD CONTROL

CONTENTS

CHAPTER EIGHT

FEEDFORWARD CONTROL

8.1 INTRODUCTION

The major innovation in distillation column control has been the implementation of feedforward/feedback control systems. Precise control of distillation processes is difficult because

(i) distillation columns exhibit long time constants and deadtime;

(ii) interactions exist between control loops;

(iii) product composition analysers are generally not continuous;

(iv) column dynamics are a strong function of the operating conditions.

These problems can be overcome by the application of feedforward control using either analog elements or digital computers (MacMullen and Shinskey (1964), Lupfer and Johnson (1964), Luyben and Gerster (1964)).

The implementation of feedforward control requires the selection of a model which adequately describes the behaviour of the column over the full range of operating conditions and process disturbances. This model can be a rigorous one requiring a minicomputer to solve for the required control variables, or in the other extreme, it can be a black box model fitted by regression analysis. The latter approach generally has insufficient accuracy over the required operating range. In between these two extremes, come the shortcut procedures for producing first estimates of column conditions. The Fenske (1932) - Underwood (1946) - Gilliland (1940) approach is the common shortcut design method. Shinskey (1967) has suggested an alternative feedforward model based on a constant distillate to feed ratio, and an adaption of the Fenske (1932) minimum stages equation. It is possible to provide model adaption for this approach as detailed by Duyfjes and Van Der Grinten (1973).

The column models described are steady state and to correct for the dynamics of the column it may be necessary to introduce dynamic compensation into the feedforward controller. It is difficult to determine the exact compensating dynamics required, especially for a non-linear process like a distillation column, but lead/lag units can be used to approximate the required compensator.

8.2 A FEEDFORWARD CONTROLLER

Previous work by Williamson (1977) and others has shown the Gilliland correlation to be useful as a feedforward model. This approach was investigated using the computer model SSGW and the pilot scale column.

8.2.1 Derivation of the Controller Equations

The Gilliland correlation was suggested as a means of relating the reflux ratio to the number of theoretical plates in terms of the minimum number of stages and the minimum reflux ratio . The McCabe-Thiele analysis (1925) can be used to predict the minimum reflux ratio (R_m) and the Fenske (1932) equation to predict the minimum number of stages (N_m). The equations are:

Gilliland correlation

$$\exp\left(\frac{N_m}{N}\right) = a - b\left(\frac{R_m + 1}{R + 1}\right) \qquad a,\ b \text{ constants} \qquad (8.1)$$

McCabe Thiele analysis:

$$R_m = f(r_D,\ x_F,\ T_F,\ \text{vapour/liquid equilibrium}) \qquad (8.2)$$

Fenske equation:

$$N_m = \frac{\log\left(\frac{r_D(1-r_W)}{r_W(1-r_D)}\right)}{\log \alpha} \qquad (8.3)$$

N_m includes the reboiler.

Equation (8.1) can be rewritten as

$$R = \frac{b(R_m + 1)}{a - \exp(\frac{N_m}{N})} - 1 \qquad (8.4)$$

Given N, r_D, r_W, x_F T_F , a reflux ratio to produce the desired split can be predicted by equations (8.2), (8.3), (8.4). The other column control variables, Q_s the steam flow, must be calculated from balances around the column. Assuming equimolal overflow, the following equations result:

$$D = \frac{F(x_F - r_W)}{(r_D - r_W)} \tag{8.5}$$

$$W = F - D \tag{8.6}$$

$$R' = R\left(1 + \frac{C_{PR}(T_{RS} - T_R)}{\Delta H_{VR}}\right) \tag{8.7}$$

R' = internal reflux ratio

C_{PR} = average heat capacity of reflux

T_{RS} = saturation temperature of reflux

T_R = temperature of reflux

ΔH_{VR} = latent heat of vaporisation of reflux

$$L_R' = R'D \tag{8.8}$$

L_R' = internal reflux flow

at the feed tray

$$q = \frac{H_F - h'_F}{H_F - h_F} \tag{8.9}$$

q = feed condition

h'_F = actual feed enthalpy

$$L_s = L_R' + qF \tag{8.10}$$

$$V_s = L_s - W \tag{8.11}$$

$$Q_s = \frac{V\,\Delta H_{VW}}{\Delta H_{Vsteam}} \tag{8.12}$$

ΔH_{VW} = latent heat of vaporisation of bottoms

ΔH_{Vsteam} = latent heat of vaporisation of steam.

8.2.2 Simulation of the Feedforward Controller

The steady state column model, SSGW, was used to test the controller outlined above. Using experimental data from the pilot scale column, the parameters for the Gilliland correlation were determined. The results are shown in figure 8-1. To analyse the experimental data, R_m was calculated from equation 8.2, N_m from equation 8.3 using α = 4.2 (the geometric average for methanol/water taken at $x = 0.05$, $x = 0.90$), and N from a McCabe-Thiele analysis. The resulting correlation was

$$R = \frac{2.11(R_m + 1)}{3.54 - \exp\left(\frac{N_m}{N}\right)} - 1 \qquad (8.13)$$

The coefficients were found by a least squares fit.

The controller described by equations 8.5 to 8.13 was used to predict the reflux ratio and the steam flow to achieve given product compositions for a given set of column operating conditions. These variables were entered into the steady state model, and the model product composition predictions compared with the desired compositions. Figure 8-2 shows some typical results.

The predictions from the steady state feedforward controller described were unacceptable showing large offsets in both product compositions of up to 7 mol%. Figure 8-1 explained most of this offset; the Gilliland correlation did not represent the data well, and the scatter of points contributed to the errors in the controller predictions. The other assumptions made in the feedforward model -

(i) constant relative volatility for finding N_m,

(ii) equimolal overflow for finding Q_s,

(iii) constant number of theoretical plates,

also contributed to the inaccuracy of the controller. To improve the controller performance over a range of operating conditions, an adaption routine was added.

8.3 AN ADAPTIVE FEEDFORWARD CONTROLLER

8.3.1 The Adaption Routine

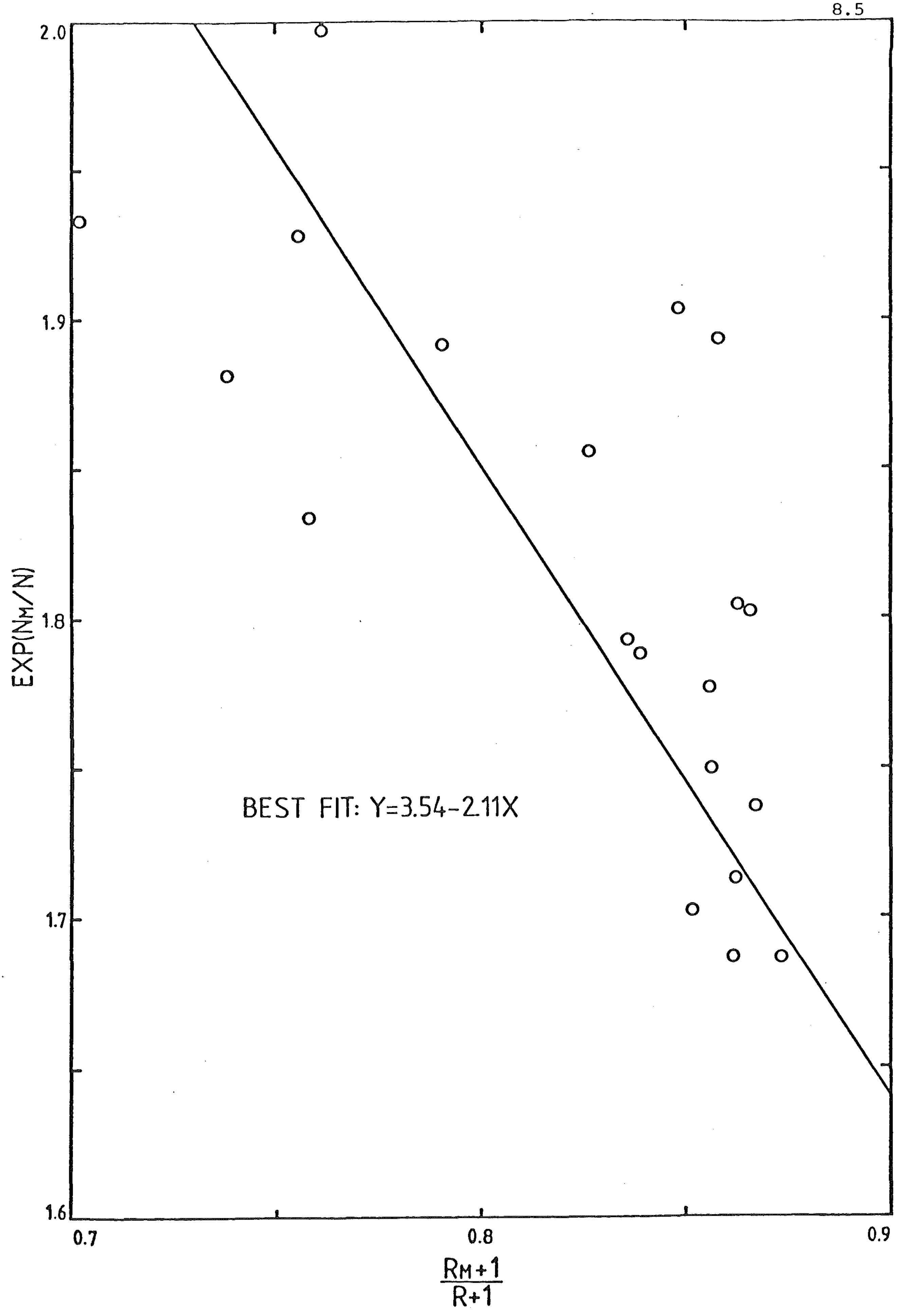


FIGURE 8-1 EXPERIMENTAL GILLILAND PLOT

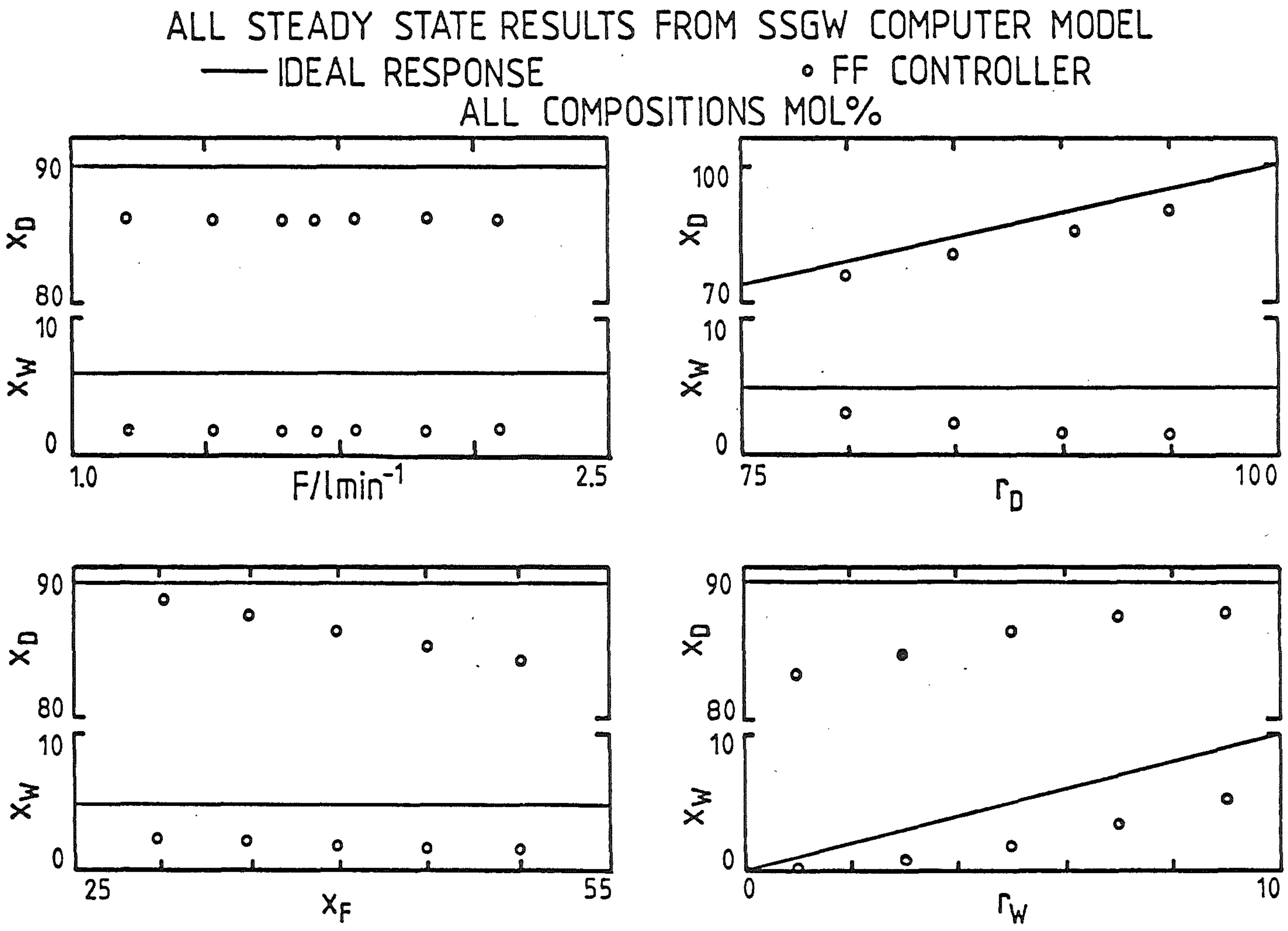


FIGURE 8-2 FEEDFORWARD CONTROLLER RESPONSES

The adaption routine modified a parameter in the feedforward model whenever the column reached steady state and showed some offset from the desired setpoints. The parameter to be adapted could be chosen from the following:

(i) N the number of theoretical trays in the column;

(ii) a, b, the Gilliland correlation ceofficients;

(iii) α , the relative volatility.

The first option was chosen because:

(i) the number of theoretical trays was a function of the column operating conditions and hence changed with load changes;

(ii) the adaption procedure was simple.

The number of theoretical trays was adopted by substituting x_D for r_D and x_W for r_W in the feedforward controller equations, and reorganising the Gilliland correlation as

$$N = N_m / \left[\ell n \left(a - b \left(\frac{R_m + 1}{R + 1} \right) \right) \right] \qquad (8.14)$$

The adaption procedure could be used iteratively with the controller described previously until convergence on the desired product compositions was achieved. The value of N obtained could be used for small load changes, until such time as the operating conditions changed sufficiently to produce off specification products, at which point the adaption procedure was repeated.

8.3.2 Simulation of the Adaptive Controller

The simulations performed on the feedforward controller were repeated with the adaptive controller. For three adaptions under constant operating conditions, the results were no better than those obtained without the adaption extension. The adaption procedure was stable and converged rapidly for all operating conditions. It was obvious that some of the assumptions involved in computing R_m , N_m and Q_S were invalid. To investigate, pseudo-methanol/water systems were used with the following characteristics.

(i) Ideal vapour/liquid equilibrium data ($\alpha = 4.2$).

(ii) N_m computed from the McCabe-Thiele plot.

(iii) Equimolal overflow ($\Delta H_V = 40$ kJ mol^{-1}).

Cases (i) and (ii) produced no improvement in the controller (with or without adaption), but case (iii) produced excellent results with a single adaption. Therefore it was concluded that the incorrect calculation of the boilup rate in the reboiler was causing the controller errors, and that the predictions based on equimolal overflow were inadequate for the methanol/water system. These conclusions are consistent with the results obtained in Chapter 6.

8.3.3 A Modified Adaptive Feedforward Controller

Computer simulations indicated that a more accurate method for estimating the required steam rate was necessary. It was also important that the method be quick and simple for inclusion into the microcomputer control system.

Consider the rectifying section of the column as shown in figure 8-3, and suppose the liquid stream loses i moles of the light component (1), and gains j moles of the heavy component (2), then for the liquid stream

$$L_{RF} = L_R - i + j \tag{8.15}$$

and for a heat balance (assuming adiabatic operation, and perfect solution behaviour)

$$i\ \Delta H_{V1} = j\ \Delta H_{V2} \tag{8.16}$$

A component balance for the light fraction over the liquid stream gives

$$L_R'\ x_D = L_{RF}\ x_{RF} + i \tag{8.17}$$

Solving (8.15), (8.16), (8.17) gives

$$L_{RF} = L_R' \left(\frac{\beta - x_D}{\beta - x_{RF}} \right) \tag{8.18}$$

where

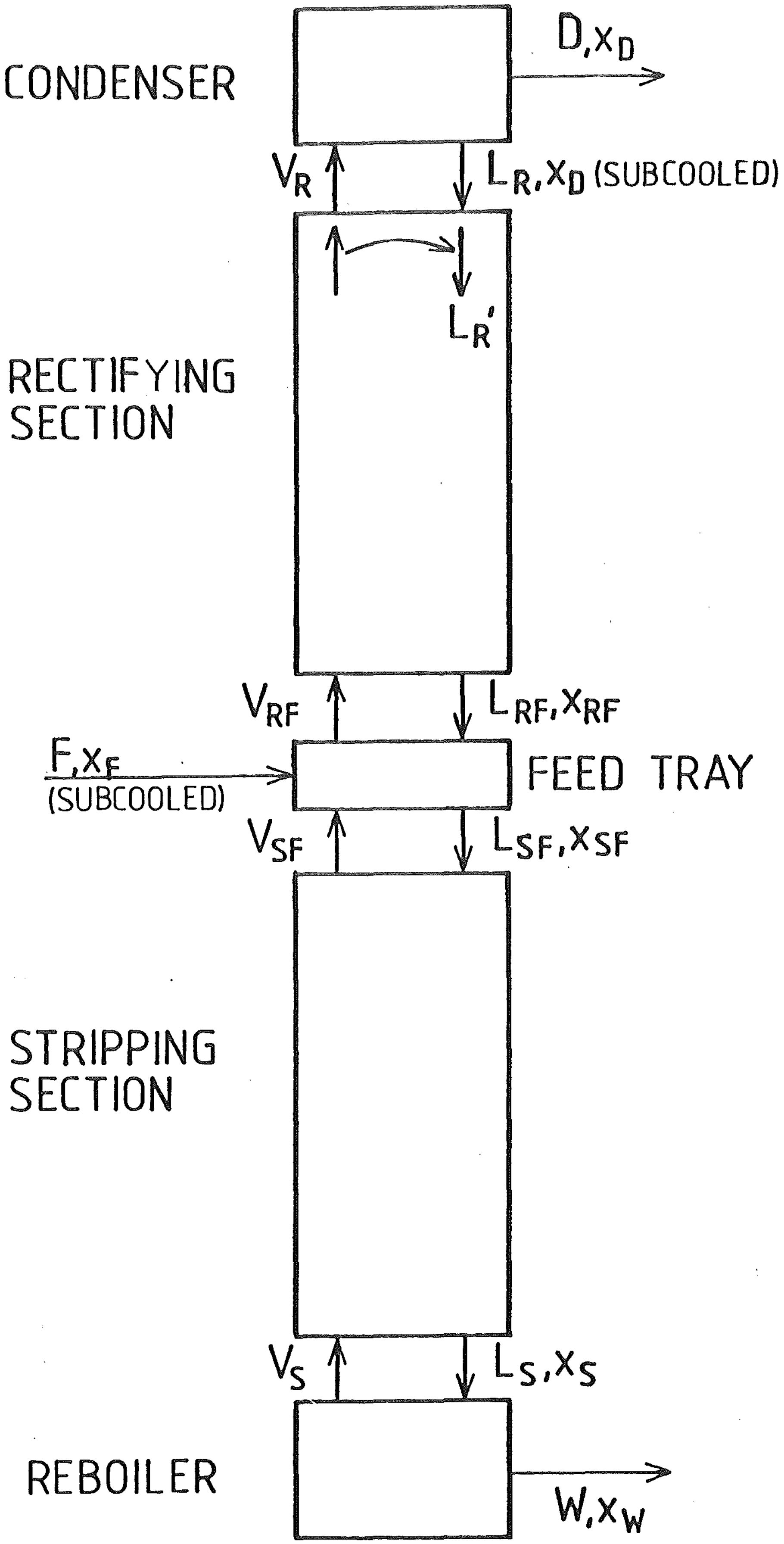


FIGURE 8-3 SIMPLE COLUMN MODEL

$$\beta = \frac{\Delta H_{V2}}{\Delta H_{V2} - \Delta H_{V1}} \qquad (8.19)$$

Now consider the stripping section where the liquid stream loses k moles of the light component (1) and gains m moles of the heavy component (2), then for the liquid stream

$$L_S = L_{SF} - k + m \qquad (8.20)$$

and for a heat balance (assuming adiabatic operation, and perfect solution behaviour)

$$k\,\Delta H_{V1} = m\,\Delta H_{V2} \qquad (8.21)$$

A component balance for the light fraction over the liquid stream gives:

$$L_S\, x_S = L_{SF}\, x_{SF} - k \qquad (8.22)$$

Solving (8.20), (8.21), (8.22) gives

$$L_S = L_{SF} \left(\frac{\beta - x_{SF}}{\beta - x_S} \right) \qquad (8.23)$$

and from equation (8.10)

$$L_{SF} = L_{RF} + qF \qquad (8.24)$$

If the average liquid composition on the feed tray is close to the feed composition, and the liquid composition change on the feed tray is small, then

$$x_{SF} \simeq x_{RF} \simeq x_F$$

and a modified procedure for determining the required boilup rate exists:

(i) $$L_R' = L_R \left[1 + \frac{C_{PR}\, \Delta T_R}{\Delta H_{VR}} \right]$$

(ii) $$L_{RF} = L_R' \left(\frac{\beta - r_D}{\beta - x_F} \right)$$

(iii) $$L_{SF} = L_{RF} + qF$$

(iv) $$L_S = L_{SF}\left(\frac{\beta - x_F}{\beta - x_S}\right)$$

(v) $$V_S = L_S - W$$

The only undefined variable is x_S, the composition of the liquid stream entering the reboiler. Consider the relationship of the latent heat of vaporisation to composition as given in Table 6-3 for the methanol/water system. For liquids below 0.20 m.f., the latent heat of vaporisation is approximately constant, and hence for trays operating below this composition, there will be equal transfer of both components, i.e. equimolal overflow. If the column is operated such that the liquid leaving the bottom tray is less than 0.2 m.f. methanol, then x_S in equation 8.3 can be approximated as 0.2. For the methanol/water system $\beta = 7.55$, hence equation (8.23) becomes

$$L_S = \left(\frac{7.55 - x_F}{7.35}\right) \qquad (8.25)$$

8.3.4 Simulation of Modified Controller

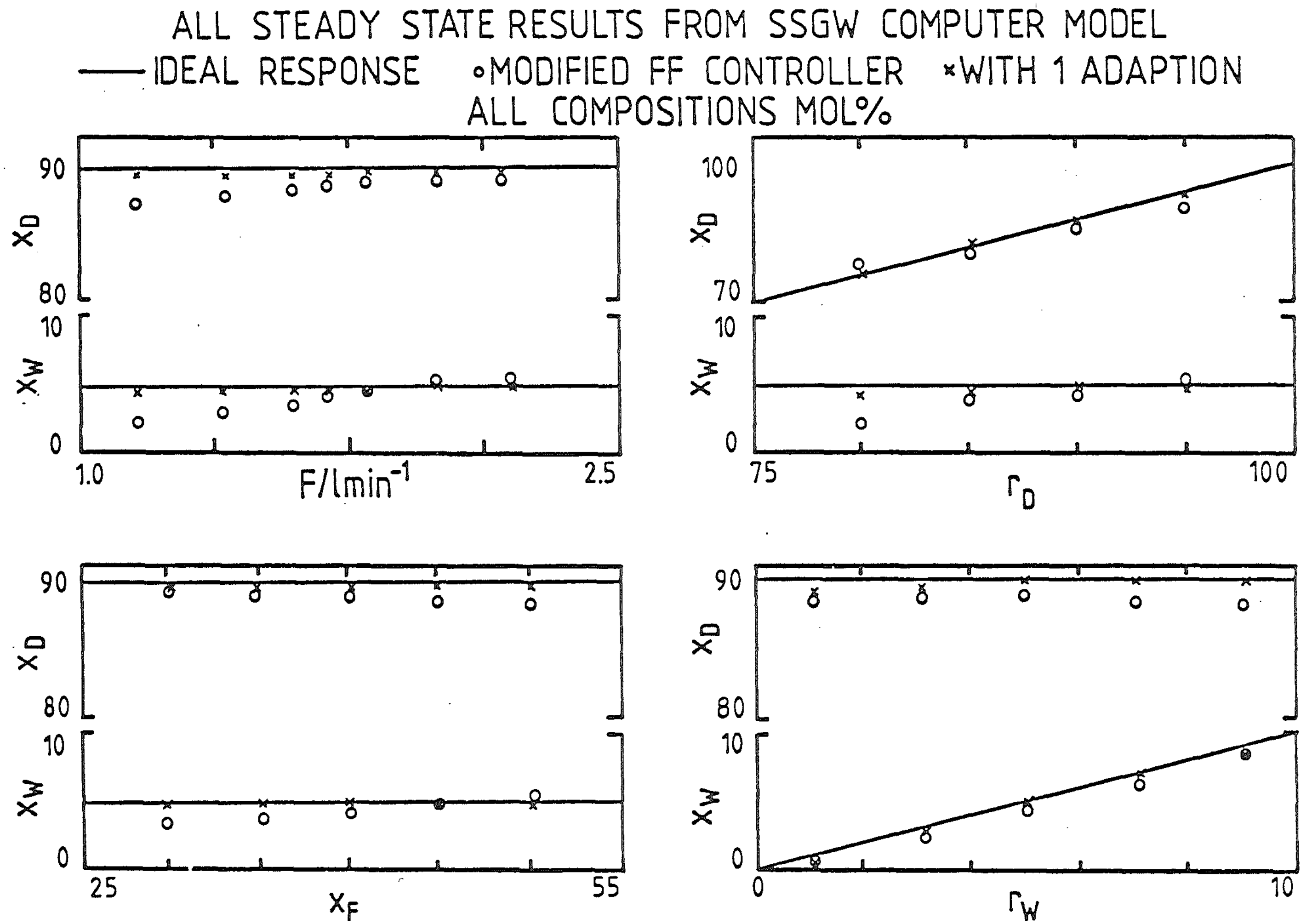
The modified adaptive feedforward controller was tested using the steady state column model, SSGW, and the parameters in section 8.2.2. The number of theoretical trays was taken as 6.5 initially. The steady state results for the controller, and the controller with one adaption are shown in figure 8-4. Both sets of results showed improvement over the results of figure 8-2, with the adaptive controller being better overall. The steady state offsets were less than 2 mol% for the modified controller, and less than 1 mol% for the adaptive modified controller (with one adaption). Investigations were made of the effects of

(i) the choice of coefficients for the Gilliland correlation;

(ii) the use of constant relative volatility in the Fenske equation;

(iii) the use of the Underwood equation for calculating the minimum number of stages.

These factors were shown to have negligible (< 0.2 mol%) effect



ALL STEADY STATE RESULTS FROM SSGW COMPUTER MODEL

—— IDEAL RESPONSE ∘MODIFIED FF CONTROLLER ×WITH 1 ADAPTION

ALL COMPOSITIONS MOL%

FIGURE 8-4 MODIFIED FEEDFORWARD CONTROLLER RESPONSES

on the steady state controller performance. A poor choice of coefficients in the Gilliland correlation required more adaptions before the desired specification products were met, but convergence was always obtained.

8.3.5 Sensitivity Analysis

A series of small perturbations (±5%) in the input parameters to the controller and the adapter were made to investigate the sensitivity of both routines. The linear models obtained from these tests are shown in figure 8-5.

FIGURE 8-5

Feedforward Controller Sensitivity Analysis

$$\begin{bmatrix} \Delta R \\ \Delta Q_S \end{bmatrix} = \begin{bmatrix} 0 & -.492 & .016 & -.930 \\ .952 & .330 & -.040 & -.442 \end{bmatrix} \begin{bmatrix} \Delta F' \\ \Delta x_F' \\ \Delta T_F' \\ \Delta N' \end{bmatrix}$$

Controller Sensitivity Model

$$\begin{bmatrix} \Delta N \end{bmatrix} = \begin{bmatrix} 0 & 3.46 & 0.120 & 36.7 & -1.35 & -5.6 \end{bmatrix} \begin{bmatrix} \Delta F' \\ \Delta x_F' \\ \Delta T_F' \\ \Delta x_D' \\ \Delta x_W' \\ \Delta R' \end{bmatrix}$$

Adapter Sensitivity Model

All compositions in m.f.

All temperatures in °C

All flows in ℓ min^{-1}

$$\Delta F' = \frac{\Delta F'}{F_{SS}} \quad \text{(SS = steady state) etc.}$$

The controller was found to be most sensitive to N , the number of theoretical stages, and to a lesser degree to the feed rate and composition. The adapter was found to be most sensitive to the estimate of the reflux ratio, and to the liquid compositions (x_F, x_D, x_W). These results show that errors in the feedforward controller (appearing as steady state offsets) will be likely in the practical application of the controller because of the difficulty of measuring flows and compositions accurately. The results of figure 8-4 were obtained using six significant digits (in all parameters) in a digital simulation, but in practice such accuracy would not be achieved.

8.4 IMPLEMENTATION OF THE MODIFIED ADAPTIVE FEEDFORWARD CONTROLLER

The addition of the APU system to the microcomputer (Chapter Four) simplified the development of a more complex control system using the adaptive feedforward controller described in the previous sections. The simple control system of Chapter 7 was modified so that composition control was achieved as outlined in figure 8-6. Feedback controllers were incorporated to account for the steady state offsets that were expected from the feedforward controller. Dynamic elements were added to both the feedforward controller (compensation) and the feedback controllers (decoupling) to counteract the column dynamics. A pair of integral-only trimming controllers on the feedback loop setpoints were incorporated to make the small adjustments to the tray temperature setpoints as required by changing column operating conditions. Some simplifications were made in programming the feedforward controller and adapter to improve its execution time and storage requirements (e.g. the calculation of the minimum reflux ratio).

8.4.1 The Feedforward Controller

The relevant equations have been presented in previous sections. The minimum number of stages was determined from the Fenske equation

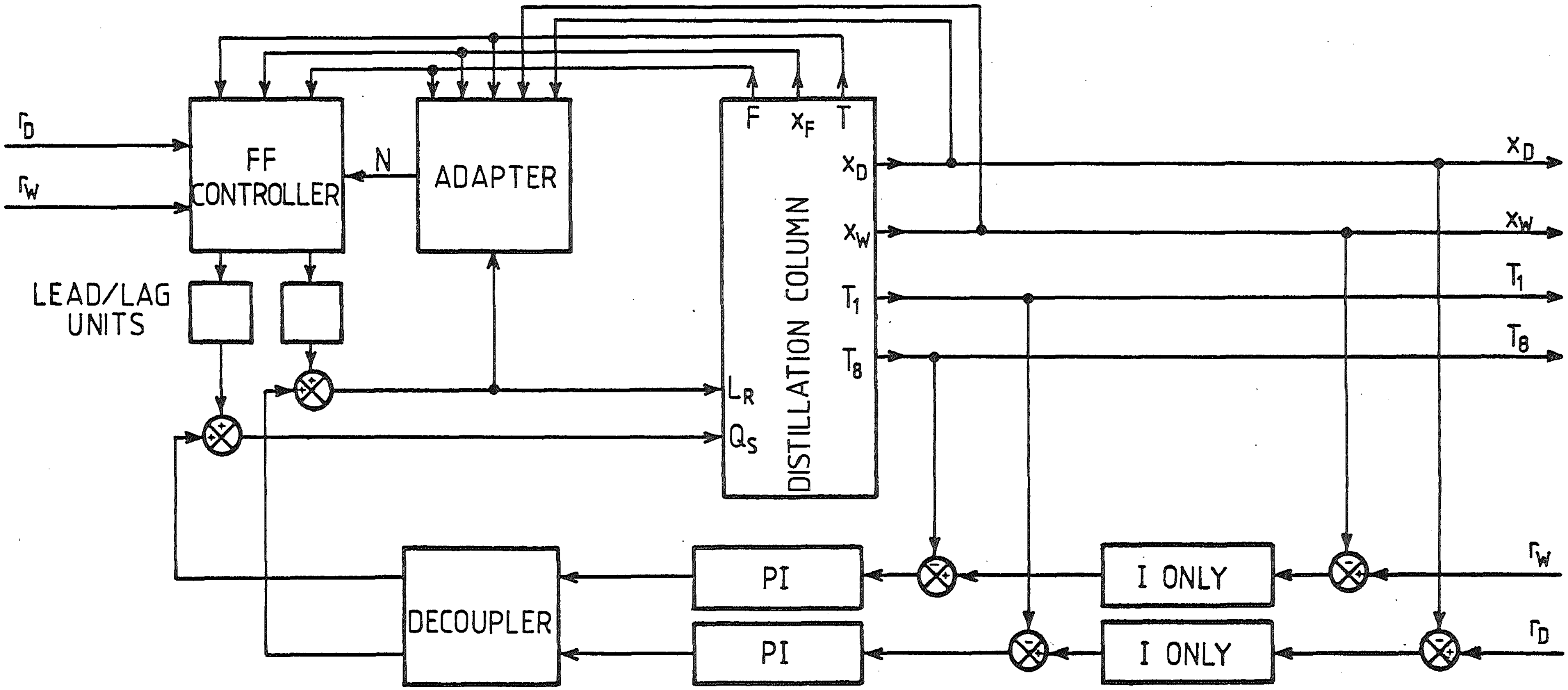


FIGURE 8-6 ADAPTIVE FEEDFORWARD/FEEDBACK CONTROL SYSTEM

(with $\alpha = 4.2$). The minimum reflux ratio was determined from a McCabe-Thiele diagram; at minimum reflux, a pinch point occurs at the intersection of the feed line and the operating line on the equilibrium line. By approximating the equilibrium line with a straight line over the feed composition range, the minimum reflux ratio can be determined analytically as follows:

feed line: $$q = 1 + 2E\text{-}3\ (85 - 23x_F - T_F) \tag{8.26}$$

(maximum error of 8% in q over the range $.3 \le x_F \le .5$, $20°C \le T_F \le 60°C$)

equilibrium line: $$y^* = .566x + .498 \tag{8.27}$$

(maximum error of 0.012 m.f. over the range $0.3 \le x \le 0.5$)

The intercept of the feed line and equilibrium line is at

$$x_i = \frac{\left(\frac{x_F}{q-1} + .498\right)}{\left(\frac{q}{q-1} - .566\right)} \tag{8.28}$$

$$y_i = .566x + .498 \tag{8.29}$$

and

$$R_m = \frac{(x_D - y_i)}{(y_i - x_i)} \tag{8.30}$$

The reflux ratio for a given separation can be found from equation (8.4), and the reflux and steam flows calculated according to sections 8.2.1 and 8.3.3. To convert the external reflux flow to the apparent internal reflux flow, the following relationship was used -

$$L_R' = 1.11\ L_R \tag{8.31}$$

(maximum error of 0.2% for $.8 \le x_D \le 1.0$, $20°C \le T_R \le 30°C$)

For methanol/water mixtures of less than 0.2 m.f., the molal latent heat of vaporisation was assumed constant at 40.8 kJ mol^{-1}. Assuming the latent heat of vaporisation of the supplied steam to be 2250 kJ kg^{-1} then

$$Q_S = V_S * \frac{40.8}{2250} = \frac{V_S}{.55} \text{ kg min}^{-1} \qquad (8.32)$$

A correction to steam flow for the estimated column surface losses was made by adding 0.07 kg min^{-1} to Q_S . This correction is discussed in Chapter 6.

The steam and reflux flows were converted to integers in the range 0-255 for output as single bytes to the D/A converters in the control system. The functions used for the conversion were:

$$\text{reflux density} = .957 - .170\ x_D \text{ kg } \ell^{-1} \qquad (8.33)$$

$$\text{reflux flow control byte} = 383L_V - 214 \qquad (8.34)$$

$$L_V = \text{reflux flow in } \ell\text{min}^{-1}$$

$$\text{steam flow control byte} = 202\ Q_S + 136 \qquad (8.35)$$

$$Q_S = \text{steam flow in kg min}^{-1}$$

Dynamic compensation was added to the controller to compensate for the column dynamics. The dynamics were approximated by the discrete digital equivalent of a lead/lag unit.

$$\frac{\tau_1 S + 1}{\tau_2 S + 1}$$

as

$$z_n = x_n + K_1(x_n - y_n) \qquad (8.36)$$

$$y_{n+1} = y_n + K_2(x_n - y_n) \qquad (8.37)$$

where x = input

y = intermediate value, lagging the input by τ_2

z = output leading y by τ_1

$$K_1 = \frac{\tau_1 - \tau_2}{\tau_2}, \quad K_2 = \frac{T}{\tau_2}$$

8.4.2 The Adapter

The same basic controller equations were used to adapt the feed-forward controller with x_D and x_W replacing r_D and r_W . The product compositions were determined from the refractometer data using the correlations given in Appendix II. The decision to adapt the controller should be made when

(i) both product compositions have been steady for T minutes

(ii) all inputs to the column have been steady for T minutes

(iii) all column controls have been steady for T minutes

where $T \simeq 10$ minutes.

To keep a log the the nine variables as listed above in the microcomputer would have required a large section of software and data storage, therefore adaption was actioned on an operator request. On setting a byte in the data base, the adaption routine adapted the controller on the next sample. This system removed the complicated logic of deciding when to adapt and prevented unnecessary adaptions.

8.4.3 Feedback Controllers

The P.I. controllers described in Chapter 5 were split into two sections. The level controllers continued to use the existing software, while the controllers to be coupled to the feedforward controller were reprogrammed using the APU system. The outputs of the feedback controllers were set to be in the range ±127 so that a trimming action could be applied in both directions to the feedforward controller output. A steady state decoupler was included of the form:

$$\Delta V_{reflux} = \Delta V_{reflux} + K_1 \Delta V_{steam} \tag{8.38}$$

$$\Delta V_{steam} = \Delta V_{steam} + K_2 \Delta V_{reflux} \tag{8.39}$$

where ΔV = change in control output.

From the open loop transfer functions fitted in Chapter 7, and the gains in the data acquisition system and control outputs, the following steady state model was calculated

$$\begin{bmatrix} \Delta T_1' \\ \Delta T_8' \end{bmatrix} = \begin{bmatrix} -.39 & .81 \\ -.88 & 1.7 \end{bmatrix} \begin{bmatrix} \Delta V_1 \\ \Delta V_4 \end{bmatrix} \tag{8.40}$$

$$= P \begin{bmatrix} \Delta V_1 \\ \Delta V_4 \end{bmatrix}$$

V = control output, T' = temperature data from A/D.

The steady state decoupler is the inverse of the process matrix P

$$D = \begin{bmatrix} 34.1 & -16.2 \\ 17.6 & -7.8 \end{bmatrix} \tag{8.41}$$

8.4.4 Trimming Controllers

In evaluating the feedback control system of Chapter Seven, small offsets after load changes were observed due to the changing relationship between the temperatures on the control trays and the product compositions. Integral-only controllers, as in figure 8-6, were included in the control system to take out this offset by adjusting the temperature controller setpoints. The setpoint controllers were loosely tuned to respond to long term offsets and not to product composition transients.

The refractometer which provided the measure of the product compositions was slow and estimates of distillate and bottoms product composition were available at 3 minute intervals, and hence the trimming controller required a low integral time.

The controllers were computed as

$$\begin{aligned} \Delta \text{Setpoint}_1 &= -2(r_D - x_D) \\ \Delta \text{Setpoint}_2 &= +2(r_W - x_W) \end{aligned} \tag{8.42}$$

which was equivalent to an integration rate of .003 °C min^{-1} for the individual setpoints.

8.4.5 Software

The control scheme of figure 8-6 was implemented in 3 subroutines:

(i) FFCTL - the feedforward controller, dynamic compensation and trimming controllers.

(ii) PIFB - the feedback controllers, decoupler and controller coupling.

(iii) ADAPT - the adapter.

The feedforward controller, the trimming controllers, and the adapter were made software selectable so that each could be disabled by zeroing a byte in the data base. Similarly, the dynamic compensation and decoupler could be disabled by setting the appropriate coefficients to zero. When the feedforward controller was disabled, the reflux and steam flow controls were set at 50% of the maximum output, so that the feedback controllers could continue to operate the column. With the software structured in this manner, a variety of control strategies could be implemented. The main program of the existing software was altered to include the new subroutines. The PI controller subroutine was modified to handle the level controls only, and the subroutines for controlling data acquisition, alarm checking, refractometer control, and the control variable outputs were retained. Flow diagrams of the new routines, and the changes to the main program control sequence are shown in figures 8-7, 8-8. The modified control program occupied 4K bytes of EPROM and 1K bytes of RAM.

8.4.6 Plant Tests

The feedforward control scheme was tested on the pilot plant distillation column using several configurations. The feedback controllers were tested alone, with and without decoupling. These controllers reproduced the results obtained in Chapter Seven with the integer-arithmetic-based controllers.

The addition of the steady state decoupler using the coefficients predicted in section 8.4.3 produced poor control. The process matrix P in equation (8.41) was almost singular, and hence the decoupler coefficients were large. Any errors in determining the matrix P would have had a significant effect on the decoupler performance. Re-tuning the decoupler coefficient showed little improvement. The control tray temperatures responded 10 to 20 times faster to steam flow changes than to reflux flow changes, consequently the bottom tray temperature loop responded quickly but was upset by the slower top tray temperature control loop via the decoupler. A dynamic decoupler (e.g. a lead/lag

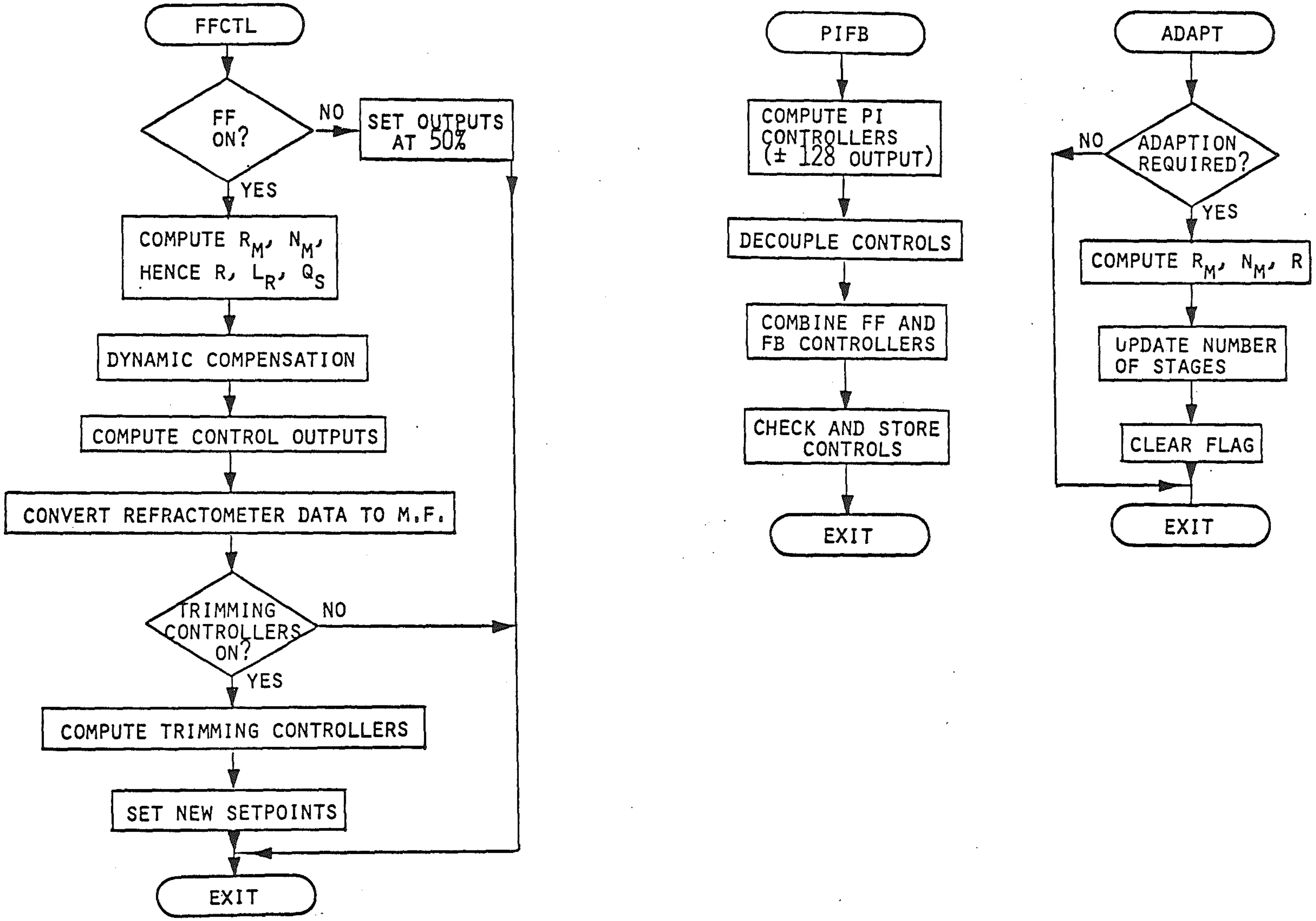


FIGURE 8-7 FEEDFORWARD/FEEDBACK CONTROLLERS

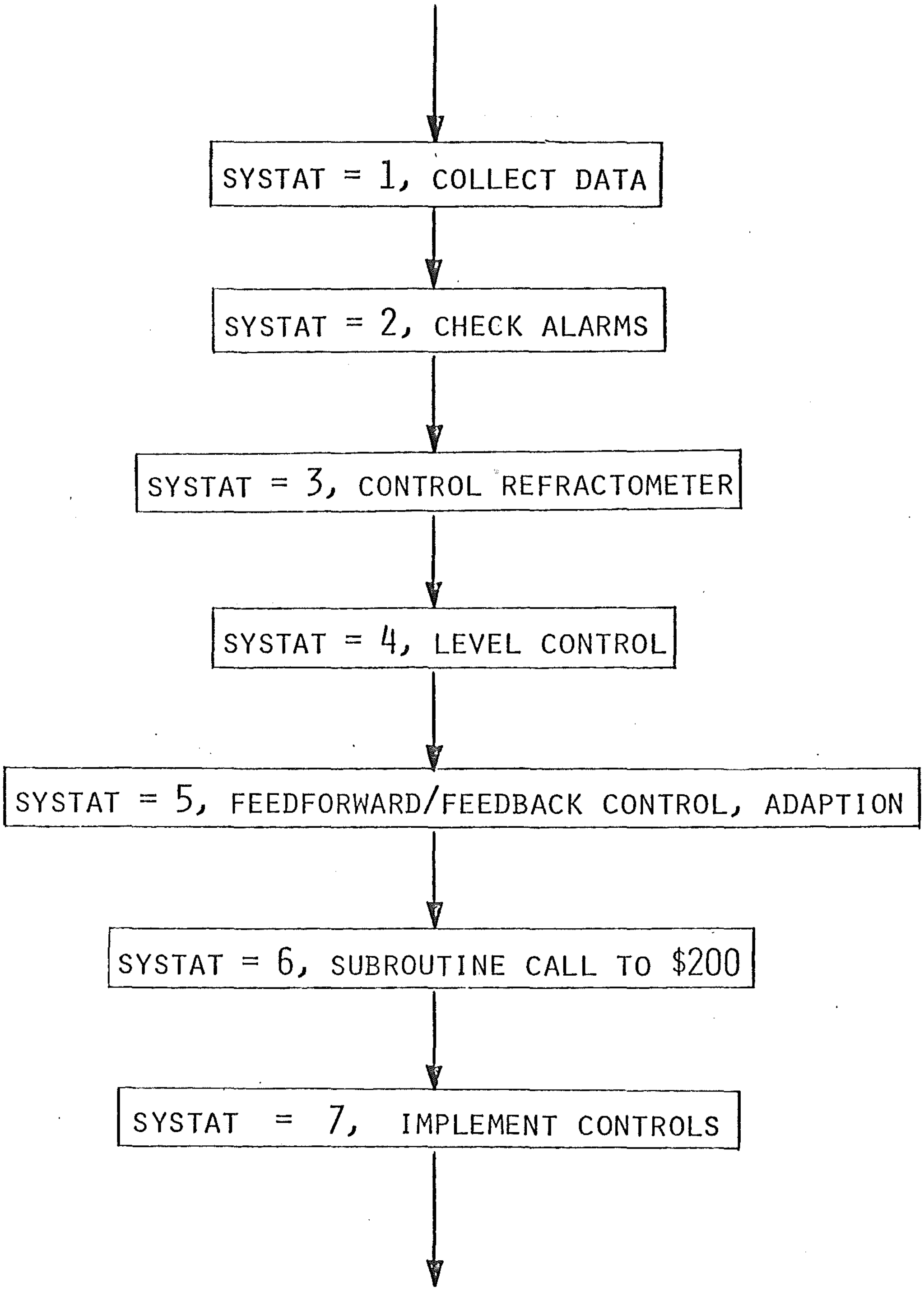

SYSTAT = 1, COLLECT DATA
SYSTAT = 2, CHECK ALARMS
SYSTAT = 3, CONTROL REFRACTOMETER
SYSTAT = 4, LEVEL CONTROL
SYSTAT = 5, FEEDFORWARD/FEEDBACK CONTROL, ADAPTION
SYSTAT = 6, SUBROUTINE CALL TO $200
SYSTAT = 7, IMPLEMENT CONTROLS

FIGURE 8-8 MAIN PROGRAM CONTROL SEQUENCE

unit) would have been a possible solution to this problem, but the control without a decoupler was satisfactory so the decoupler was disconnected.

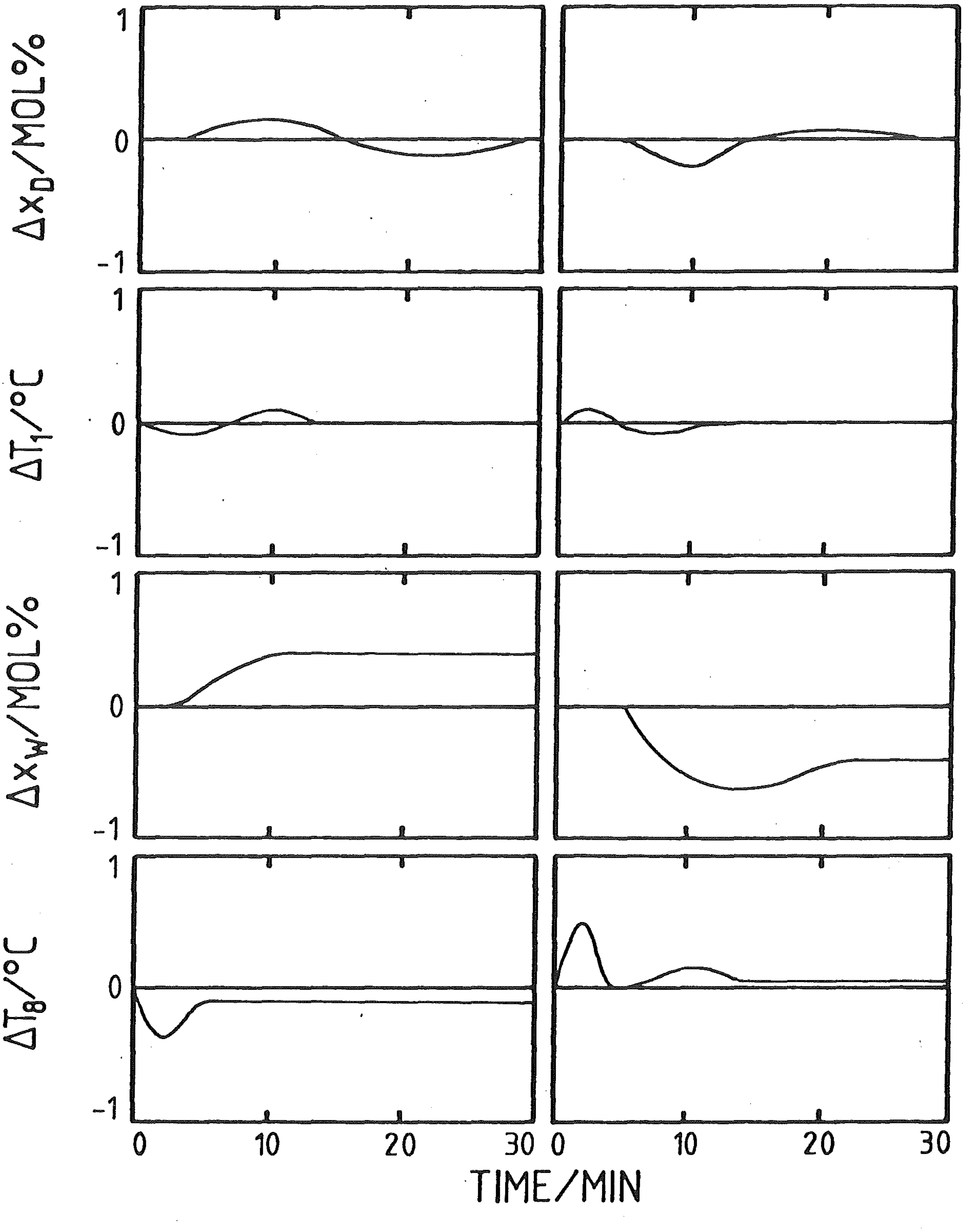
The feedforward controller was tested alone, with adaption and with dynamic compensation. Without adaption (assuming N = 6.5 from experimental measurements), large steady state offsets in the product compositions were observed (up to 5 mol%) for load and setpoint changes. The use of the adapter reduced the offset to less than 0.5 mol% on both products. The results in figures 8-9 to 8-12 show typical responses for the feedforward controller alone. The tests were made by adapting the controller until the desired setpoints were met, and then introducing the upset. The results show some offsets in the bottoms composition which could be reduced by further adaption.

The performance of the feedforward controller was excellent for the distillate composition without any dynamic compensation. The compensator was tuned by the method of Shinskey (1967), but the results were no better than those shown with no compensation (and in some cases worse).

The feedforward/feedback combination was tested using the same feedback controller settings used in Chapter 7. The results were very similar to those produced using feedback only control.

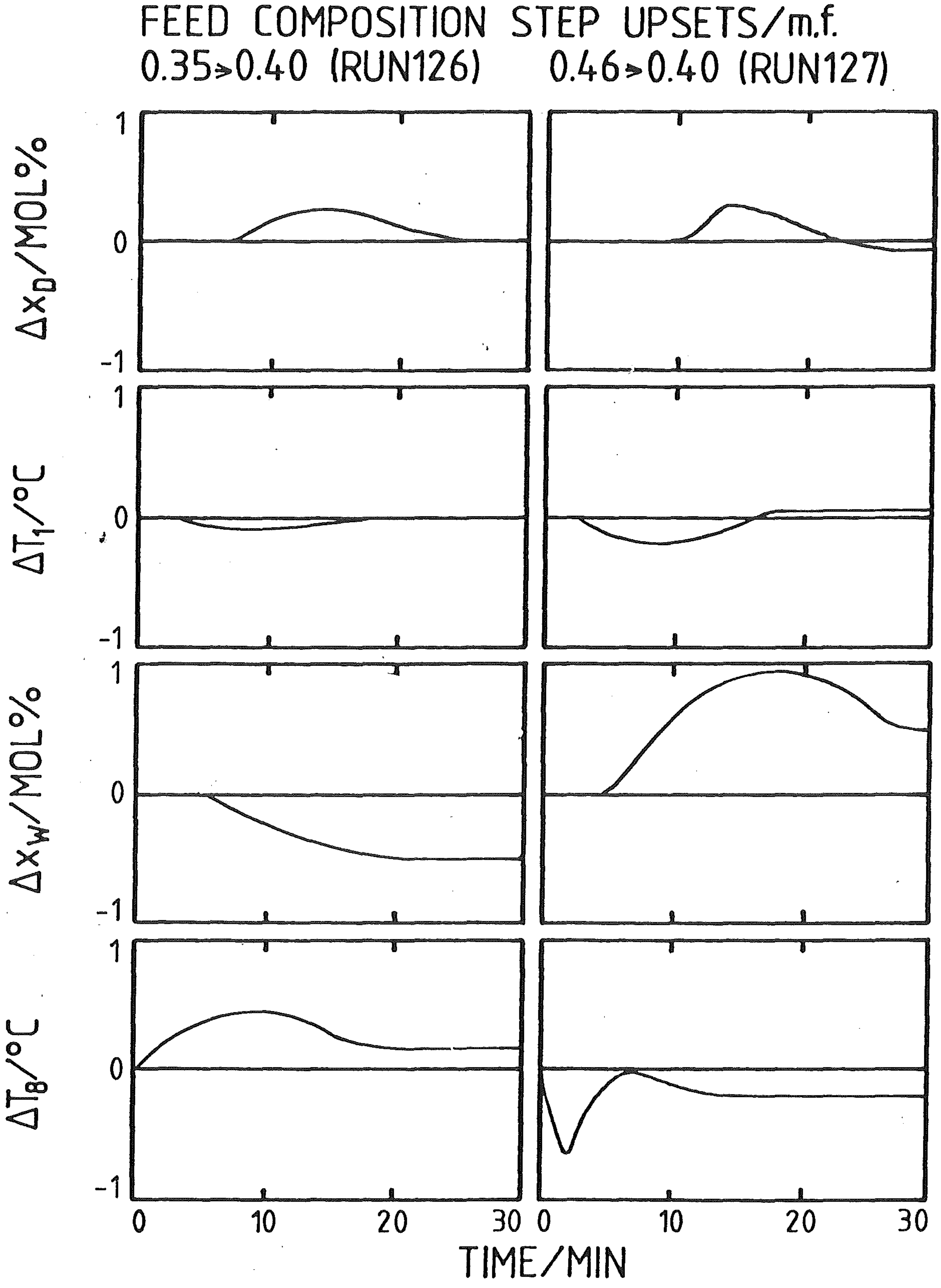
The trimming controllers helped to reduce the offsets due to the changing relationships between the control tray temperatures and the product specifications. The integral times predicted in section 8.4.4 were found satisfactory for load changes, but for setpoint changes, the temperature setpoints were ramped too slowly towards the new values, and degraded the system performance. Under these conditions, it was better to switch off the feedback and trimming controllers for setpoint changes, and to re-engage them with an estimate of the new temperature setpoints as the column approached the new steady state.

FEED FLOW STEP UPSETS/kg min^{-1}
1.54→1.72 (RUN124) 1.72→1.54 (RUN125)



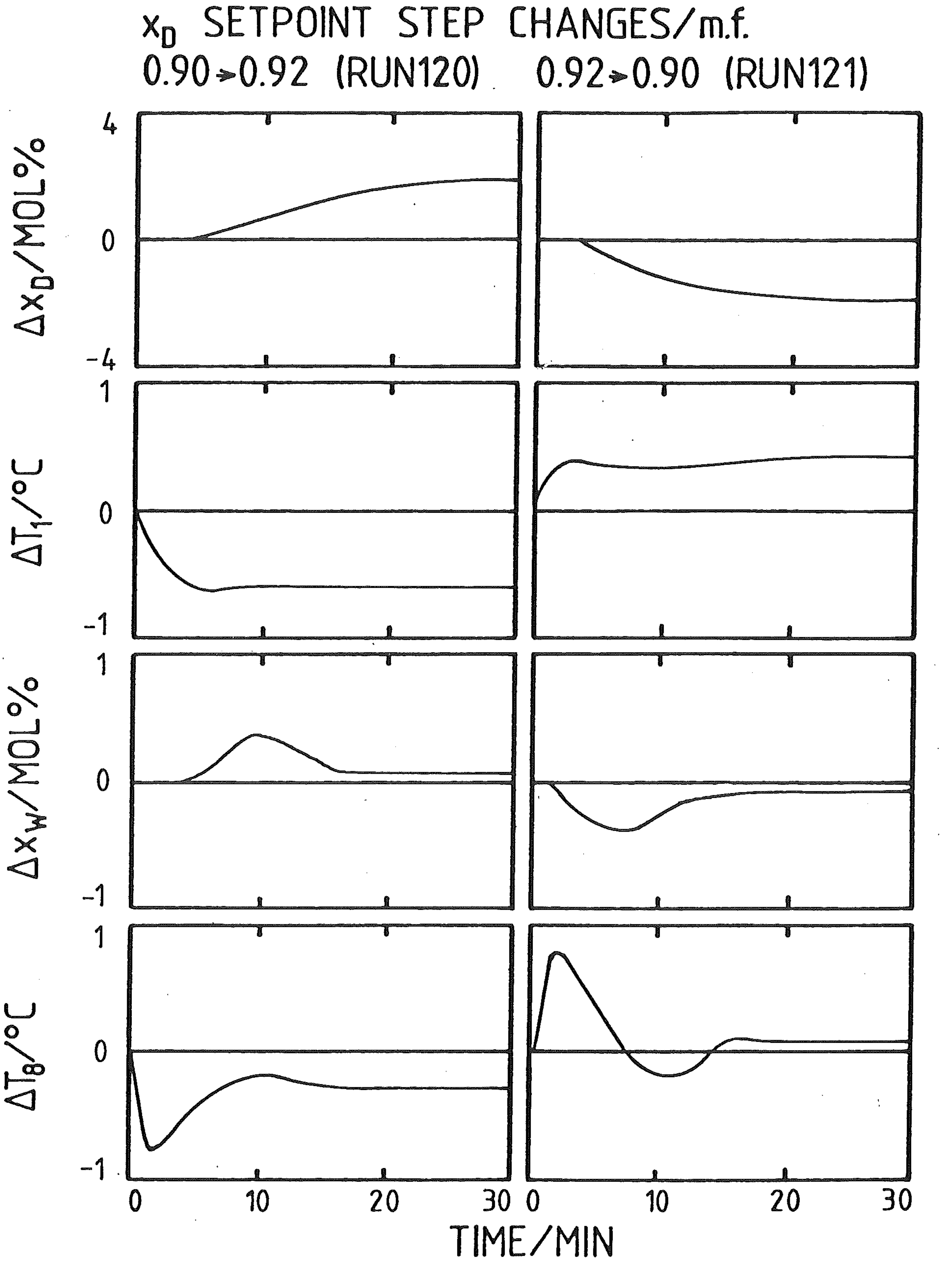
FEEDFORWARD CONTROLLER

FIGURE 8-9 CLOSED LOOP RESPONSE



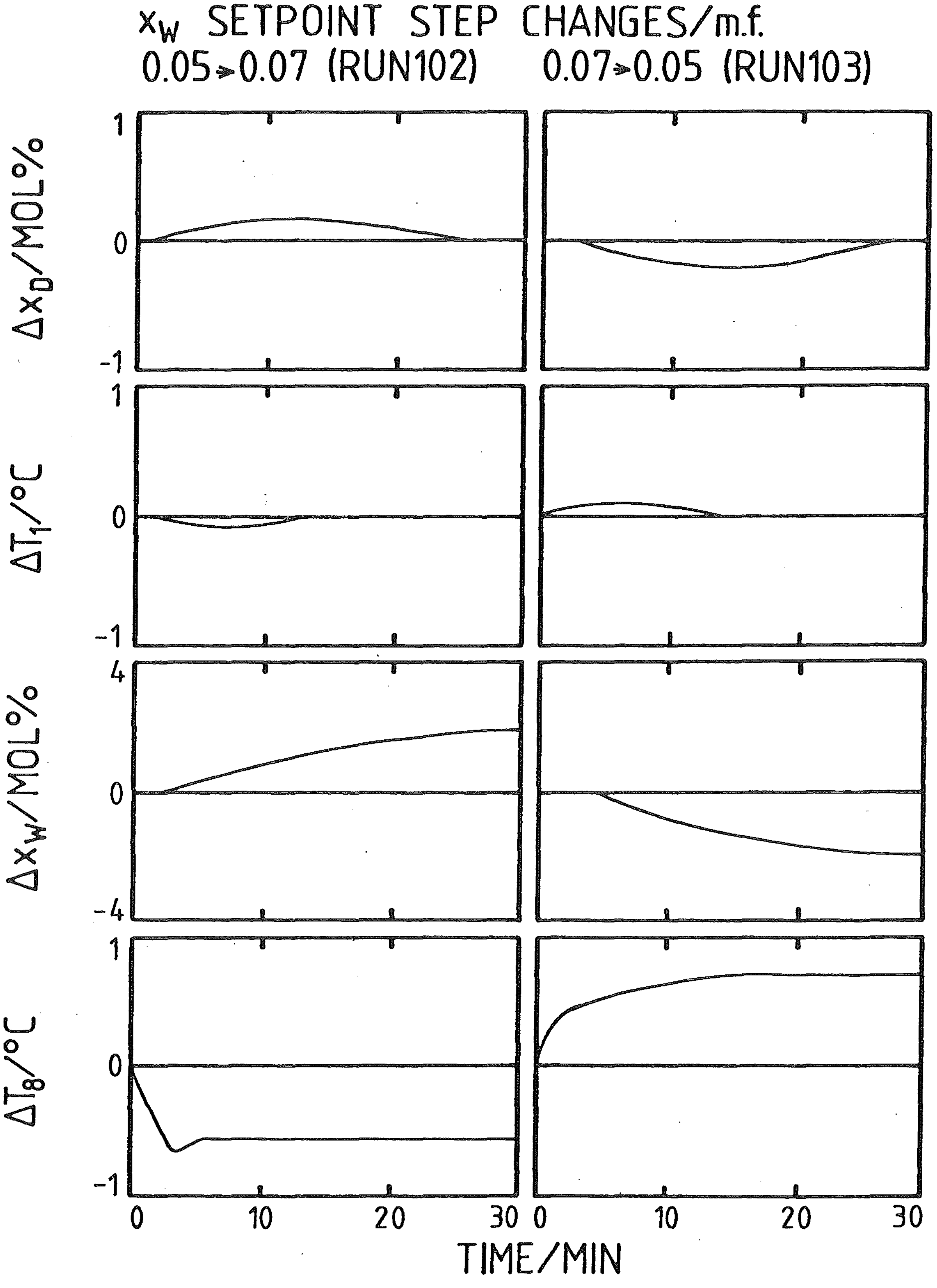
FEEDFORWARD CONTROLLER

FIGURE 8-10 CLOSED LOOP RESPONSE



FEEDFORWARD CONTROLLER

FIGURE 8-11 CLOSED LOOP RESPONSE



FEEDFORWARD CONTROLLER

FIGURE 8-12 CLOSED LOOP RESPONSE

8.5 COST COMPARISON OF THE MICROCOMPUTER-BASED FEEDFORWARD CONTROLLER WITH AN EQUIVALENT ANALOG SYSTEM

Section 5.13 showed the microcomputer based control system to be approximately equal in cost to an equivalent analog system. Using the same assumptions and costs, the following additional costs for adding the feedforward controller were calculated in $NZ (1979):

(i) Microcomputer System

(a)	Software Costs - 580 lines of assembler code @ 15 lines/day, code =		38.7 days
	Programmer @ $96/day, cost =		$3,715
(b)	Additional hardware - 8K Ram board =		$625
(c)	APU System - hardware =	$350	
	- labour =	$1,500	
	- software, 380 lines of assembler code	$2,429	
	Total =		$4,279
	Total Cost =		$8,619

(ii) Hardware System

Using the analog circuit developed by Morrin (1975) as a feedforward controller based on the Gilliland correlation, the following hardware was required

6 multipliers @ $600	$3,600
11 summers @ $500	$5,500
(based on the ACTIONPAK range of analog modules)	$9,100

This analog controller was a simplified version of the controller described in this chapter and did not include features such as the adapter, internal reflux correction, and the improved analysis for the steam flow prediction.

A comparison of the costs for the additional items required to implement the feedforward controller showed that two systems were almost equal in cost. The analog version was a simplified controller and did not include the adapter; therefore it would not be expected to perform as well as the microcomputer system. The microcomputer system was costed on a one-off basis and included the development costs of the hardware arithmetic processing unit. The flexibility and power of the microcomputer based feedforward controller was superior to that of the analog based alternative.

8.6 DISCUSSION

In all cases, the performance of the control system of figure 8-6 was found to be marginally better than that of the feedback control system described in Chapter Seven. The comments of section 7.4.3 with respect to the accuracy of the instrumentation apply equally here - the small deviations of the column variables during transients were of the same magnitude as the accuracy of the sensors, and hence the performance of the control system was as good as could be obtained with the hardware available.

The response of the feedforward control system to each of the upsets was different because of the column dynamics, and consequently the best overall control policy will involve compromises. The controlled responses obtained were satisfactory for this particular column, but it would be hard to justify the time and expense to install the feedforward/ feedback system in place of the conventional feedback system.

The Gilliland correlation was shown to be a reasonable column model provided the adaption routine was included. In tests, the adapter changed the estimate of the number of theoretical stages, N , from the experimental average of 6.5 within the range 5.7 to 6.7. The changes in N were small, but the sensitivity analysis showed the parameter to be the most critical in the controller. The simulation results showed similar trends to the experimental measurements. The analysis showed the usefulness of designing feedforward controllers around steady state models for distillation columns. This approach allowed the performance of the controller to be checked without requiring the solution of differential equations. The dynamic compensation for a feedforward controller is best implemented by simple blocks (e.g. a lead/lag) which can be tuned on-line to provide the best performance. Hence the benefits of dynamic simulations in this case were limited.

The dynamics of this particular column were such that dynamic compensation in the feedforward controller was not warranted. The responses obtained for feed rate upsets agreed with those predicted by Shinskey (1967) using the concepts of reflections within the column but the deviations during the transient response were small and acceptable. Experiments with various dynamic compensators, tuned as suggested by Shinskey (1967) produced no significant improvement in performance.

The steady state decoupler proved to be a problem in that it caused instabilities in the feedback system with a loss of control. The results without decoupling were satisfactory and suggested that the implementation of a dynamic decoupler would produce no significant improvement to the overall system performance.

Several problems were experienced with the control system. With feedforward only control, the bottoms composition tended to drift due to fluctuations in the steam flow rate. It was shown in Chapter 7 that

$$\Delta x_W (\text{mol\%}) = 32.5\ \Delta Q_S\ (\text{kg min}^{-1})$$

and hence a variation of ± 0.02 kg min^{-1} in the steam flow rate could produce a variation of ± 0.65 mol% in the bottoms composition. When feedback control was added to the steam flow control, the bottoms composition was maintained steady for periods of 4 to 6 hours.

Large steam flows caused high vapour velocities in the column, and broke the liquid seal on the rectifying section trays. Removing liquid from the top tray temperature sensor caused the temperature control loop to fail, and the column went out of control. This problem only arose when too much integral action was used in the steam flow controller, or when the feedback decouplers were used incorrectly.

The reflux pump was found to vary its delivery by approximately .05 $\ell \mathrm{min}^{-1}$ for a constant rotational velocity over a period of four hours, and was easily adjusted by the feedback controller.

The controller software was primitive in its operator communication, but effective. All conversions to the arithmetic processing unit floating point format had to be performed by hand (e.g. changing 1.0 to $0180, $0000). A more sophisticated operator interface would solve this problem. Alternately, if operated in a hierachical control scheme, the microcomputer could be operated via an interface in the supervisory computer.

The lack of suitable sensors for the feed flowrate and feed composition was a minor problem. These variables were measured and entered into the microcomputer data base by hand; in practice sensors on these variables would be used and interfaced to the data acquisition system.

The benefits that accrue from feedforward control are

(i) Large delays and lags can be handled without excessive deviations from the desired setpoints.

(ii) Corrective action can be taken when a load change occurs and not when the system responds.

(iii) Oscillatory responses can be avoided where disturbances occur at a frequency close to the natural frequency of the feedback loop.

When one or more of these factors occurs, feedforward control can improve system performance, but for this particular column, none of these factors was a problem. Harriot (1964) suggested that time constants greater than 1 hour make feedforward control attractive in distillation control. The scheme proposed here has been shown to work effectively, but to demonstrate its advantages over conventional feedback control, tests on a larger distillation column (with longer lags and delays) are required.

There are several other methods of design of feedforward controllers. The methods include the off-line approach (Burman and Maddox (1969), Distefano et al (1967)), the use of simple transfer functions (Wood and Pacey (1972)), the adaption and optimisation of a feedforward controller (Duyfjes and Van Der Grinten (1973)), and feed plate manipulation (Luyben (1968)). Luyben (1969) has also studied the extension of the feedforward controller to manipulating the setpoints of intermediate feedback tray temperature controllers directly. A number of the reported evaluations of feedforward controllers are based on simulated dynamic computer models which in some cases are oversimplified. If simulations are to be useful, they must also simulate the problems of the real world, such as the lack of precision in composition analysers and temperature sensors, the limits on valves and valve velocities, otherwise the results become academic.

A similar comment can be applied to the power of the computer used to solve the control problem - typically minicomputers with 16K or more of memory are used compared with the 5K of memory in a microcomputer in this system. It is essential that any control scheme be evaluated realistically using an economic analysis. The relative ease with which the column described in this work can be controlled has made it very difficult to distinguish between the possible control system variations.

The analysis used for the evaluation of the column flows, especially the vapour flow in the reboiler, produced values closer to the true flows than did the equimolal overflow method. This analysis could be applied to most binary systems where the latent heat of vaporisation was a linear function of composition (i.e. the binary system approached ideal solution behaviour). Methanol/water would be one of the worst binary systems (excluding effects such as azeotropes) for such a function due to the comparatively large heats of mixing (up to 1 kJ mol^{-1}). The flow calculation method could also be incorporated into the shortcut column design procedure using the Gilliland correlation to give a good estimate of the reboiler requirements.

8.7 CONCLUSION

The Gilliland correlation has been shown to be useful as the basis of a feedforward controller. The correlation was used both as a controller and an adapter to take account of changing operating conditions. Steady state simulations of the controller and its adapter showed that the assumption of equimolal overflow in computing the required boilup rate, given the other column variables, was invalid for the binary system methanol/water. A modification using a simple lumped parameter column model was proposed which successfully predicted the required reflux and steam flows for given product specifications in steady state simulations.

The adaptive feedforward controller was incorporated into the microcomputer control system with feedback control, and setpoint trimming controllers to match the feedback loop temperature setpoints to the desired product specifications. The scheme was successfully tested on the pilot plant scale column. Dynamic compensation for the feedforward controller, and decoupling of the feedback controller was found to be unnecessary for this particular column.

The performance of the complete feedforward/feedback control system was no better than that of a simple multiloop PI control system in terms of maximum overshoot and settling time. This could be explained by the reasonably fast dynamics of this column. On larger columns (with longer lags and delays) the feedforward/feedback system could be expected to perform better than the feedback only system.

The feedforward/feedback control scheme implemented in the microcomputer was costed and found to be comparable to a similar control scheme based on analog components; however, the microcomputer based system was superior in its flexibility, its ease of use, and its power (the adaption and trimming routines).

8.8 NOMENCLATURE

a,b	-	Gilliland correlation coefficients
C_{PR}	-	reflux liquid heat capacity, kJ mol K^{-1}
D	-	distillate flow, mol min^{-1}
F	-	feed rate, mol min^{-1}
h	-	liquid enthalpy, kJ mol^{-1}
H	-	vapour enthalpy, kJ mol^{-1}
L_R	-	external reflux flow, mol min^{-1}
L_R'	-	internal reflux flow, mol min^{-1}
L_{RF}, L_{SF}	-	internal column flows, mol min^{-1}
N	-	number of ideal stages including the reboiler
N_m	-	minimum number of stages including the reboiler
P	-	process matrix
q	-	feed condition
Q_S	-	steam flow kg min^{-1}
r	-	setpoint
R	-	external reflux ratio
R'	-	internal reflux ratio
R_m	-	minimum reflux ratio
T	-	temperature, °C
T_{RS}	-	saturated reflux temperature °C
T_R	-	reflux temperature
V	-	control output
W	-	bottoms flow, mol min^{-1}
x	-	liquid composition
x_S, x_{RF}, x_{SF}	-	internal column compositions
ΔH_V	-	latent heat of vaporisation
ΔH_{VR}	-	reflux latent heat of vaporisation
α	-	relative volatility

Subscripts

D	-	distillate
F	-	feed
W	-	bottoms

CHAPTER NINE

CONCLUSIONS

(1) A 225 mm diameter atmospheric pressure, sieve tray distillation column, with a capacity of 120 $\ell\,h^{-1}$ was built and commissioned. The column included eight trays (with an overall efficiency of 72%) and a thermosiphon reboiler. The condenser and reboiler capacities were 100 kW.

(2) A novel control system was developed for the column liquid flows. Variable speed motors and positive displacement vane pumps were designed and built to replace the conventional flow sensor, controller and control valve loop. Steam flow to the reboiler was controlled by a pneumatic loop with a remote setpoint.

(3) The column was fully instrumented using integrated circuit devices for sensing temperatures, levels and pressures. The accuracy of these devices was adequate, but better devices are available, e.g. thermistors. Composition sensing was performed using a multichannel on line refractometer.

(4) A microcomputer system was constructed using the Motorola M6800 family. An Am9511 arithmetic processing unit was added to improve its computational capacity. A link between the microcomputer and a minicomputer was developed, and the latter programmed as a development system (cross-assembler, download program, microcomputer supervisor).

(5) A 16 channel data acquisition system (12 bits) and a 4 channel output system (8 bits) were developed and interfaced to the column and the microcomputer. The microcomputer was also used to control and monitor the on-line refractometer.

(6) A steady state binary column model was programmed on a digital computer and verified against experimental data. The common assumptions of equimolal overflow and constant relative voltatility were shown to be

invalid for the methanol/water binary system.

(7) Step tests on the column showed that most responses of the product compositions, and top and bottom tray temperatures could be represented by first order lags and deadtime. The temperature responses showed very little deadtime, and hence column control was predicted to be good with simple feedback control.

(8) Feedback control (using 16 bit integer arithmetic, digital, PI controllers) was shown to produce good control within the accuracy of the instrumentation. Interactions between the temperature control loops required controller re-tuning.

(9) An adaptive feedforward controller was designed and tested on the steady state computer model and on the experimental column. It performed well but required feedback trimming to remove small offsets (0.005 m.f.). The performance of the combined feedforward/feedback system was no better than that achieved by feedback control alone due to the simple column dynamics.

(10) A dedicated microcomputer control scheme was shown to be feasible for a distillation column, and no more expensive than the conventional analog controllers it replaced. Inclusion in an overall plant control scheme by networking computers would provide additional benefits of optimisation and overview.

(11) The fast dynamics of the experimental column made it possible only to verify control schemes. Quantitative comparison favoured the feedback alone system since that was the most cost effective alternative.

RECOMMENDATIONS FOR FURTHER WORK

(1) An operator interface is required if the column is to be operated under microcomputer control alone (e.g. undergraduate laboratories). There is sufficient memory capacity in the microcomputer to include a sophisticated interface and simplify the column operation.

(2) Additional instrumentation is required to sense the feed flow rate and composition. More sensitive sensors on temperature (e.g. thermistors) and composition (e.g. capacitance sensors) could be investigated to improve the quality of control by improved resolution in the process variables.

(3) Investigation of the relationship between the Murphree vapour efficiency and composition for this column would provide important data for the steady state model.

(4) Constraints could be included in the control scheme to prevent the problem of excessive vapour velocity in the column. Relationships to determine the column flows necessary for stable operation are required.

(5) The feedforward/feedback control scheme described could be applied to other binary systems and to packed columns using the number of transfer units instead of the number of ideal stages. Multicomponent systems could be handled using a pseudo-binary equivalent system and modifications to the minimum reflux and minimum stages equations.

(6) Alternate control strategies could be applied including feed tray alteration and ratio control. Possible modifications to the feedback controllers include gain scheduling and self-tuning.

(7) The column dynamics could be altered by increasing the liquid holdup on the trays and in the reflux accumulator and reboiler to provide a more difficult control problem. A greater liquid depth on the trays would allow higher vapour rates to be used in the column and would increase the column capacity.

REFERENCES

AMERICAN MICRO DEVICES INC., The AM9511 Arithmetic Processing Unit. AM-PUBO 72, 1978.

AMERICAN MICROSYSTEMS INCORPORATED, Prototyping Board Manual, 1976.

AMUNDSON, N.R., PONTINEN, A.J., MCD Calculations on a Large Digital Computer. I.E.C., 50(5), 1958, p.731.

ANSELL, L.S., SAMUELS, H., FRISHE, W.C., Enthalpy-Concentration Chart for Methanol-Water Solutions. Chem. Eng., April 1951, p.133.

ARCHER, D.H., ROTHFUS, R.R., Dynamics of Distillation Units and Other Mass Transfer Equipment. C.E.P. Symp. Series 57(36), 1961, p.2.

BAILEY, S.J., Energy,Ecology Crises Spur On-Line Analysis. Control Eng. 25(8), 1978, p.27.

BAKOWSKI, S., Efficiency of Sieve Tray Columns. Brit. Chem. Eng. 14(7) 1969, p.945.

BHAT, P.V., WILLIAMS, T.J., Approximate Mathematical Models of Distillation Operations and Their Relation to Overall Column Control Methods. I.Ch.E. Symp. Series 6(32), 1969, p.22.

BOX, G.E.P., JENKINS, G.M., Time Series Analysis, Forecasting and Control. Holden Day, 1970.

BREDIG, G., BAYER, R., Z Physik Chem. 130(1), 1927.

BRISTOL, E.H., Designing and Programming Control Algorithms for DDC Systems. Control Eng. 24(1), 1977, p.24.

BRISTOL, E.H., On a New Measure of Interaction for Multivariable Process Control. I.E.E.E. Trans. Auto. Contr. AC-11(1), 1966, p.133.

BRISTOW, Q., OP Amps Multiply RC Time Constants. Electronics, April 24, 1972, p.102.

BROWN, G.G., MARTIN, H.Z., An Empirical Relationship Between Reflux Ratio and the Number of Equilibrium Plates in Fractioning Columns. Trans. A.I.Ch.E. 35, 1939, P.679.

BRUCE, R.G., FANNING, R.J., How Direct Digital Control Works. Hydrocarbon Processing, Dec. 1964, p.83.

BUFFHAM, B.A., On Inventory Times in Approximate Dynamic Distillation Modelling. A.I.Ch.E.J., 20(5), 1974, p.619.

BURMAN, L.K., MADDOX, R.N., Dynamic Control of Distillation Columns. I.E.C. Process Des. Dev. 8(4), 1969, p.433.

CARNAHAN, B., LUTHER, H.A., WILKES, J.O., Applied Numerical Methods. John Wiley, 1969.

CLAGGETT, E.H., Interpreters vs Compilers for On-Line Microcomputers. Control Eng. 24(8), 1977, p.24.

CORNEL, L.W., MONTONNA, R.E., Studies in Distillation PT II. I.E.C., 25(12), 1933, p.1331.

CUMMINGS, G.A., MILLER, G.S., Application of a Bipolar Microprocessor Chip Set to Control Systems. Proc. JACC, 1977, p.40.

DIGITAL EQUIPMENT CORPORATION, Peripherals Handbook, 1973, p.4-165.

DISTEFANO, G.P., MAY, F.P., HUCKABA, C.E., Transient Response and Feedforward Control of a Distillation Tower Subject to a Sequence of Upsets. A.I.Ch.E., 13(1), 1967, p.125.

DIXON, L.C.W., Non-Linear Optimisation. English University Press, 1972.

DOROSHEVSKY, A., POLANSKY, E., Z Physik Chem. 73, 1910, p.192.

DUYFJES, G., VAN DER GRINTEN, P., Application of a Mathematical Model for the Control and Optimisation of a Distillation Plant. Automatica, 9, 1973, p.537.

EDGAR, T.F., SCHWANKE, C.O., A Review of the Application of Modern Control Theory of Distillation Columns. Proc. JACC, 1977, p.1370.

EDGAR, T.F., VERMEYCHUK, J.G., LAPIDUS, L., The Linear-Quadratic Control Problem: A Review of Theory and Practice. Chem. Eng. Comm. 1, 1973, p.57.

ERBAR, J.H., MADDOX, R.N., Latest Score: Reflux vs Trays. Petr. Ref. 40(5), 1961, p.183.

FENSKE, M.R., Fractionation of Straight Run Pennsylvania Gasoline, I.E.C., 24, 1932, p.482.

FLÜGGE, W., Handbook of Engineering Mechanics. McGraw-Hill, 1962.

GALLANT, R.W., Physical Properties of Hydrocarbons. Gulf Publishing Co., 1968.

GARELICK, E.L., Characterising Signal Conditioners with Plug-In-Memory. Control Eng. 24(9), 1977, p.62.

GERSTER, J.A., BABER, M.F., Experimental Transient Response of a Pilot Plant Distillation Column Pt. II. A.I.Ch.E.J., 8(3), 1962, p.407.

GERSTER, J.A., LUYBEN, W.L., VERNEUIL, V.S., Experimental Transient Response of a Pilot Plant Distillation Column Pt. IV. A.I.Ch.E.J. 10(3), 1964, p.357.

GILLESPIE, D.M., Hierachial Language for Process Control. Proc. JACC, 1977, p.113.

GILLILAND, E.R., Multicomponent Rectification - Estimation of the Number of Theoretical Plates as a Function of the Reflux Ratio. I.E.C., 32, 1940, p.1220.

GILLILAND, E.R., MOHR, C.M., Transient Behaviour in Plate Tower Distillation of a Binary Mixture. C.E.P., 59(9), 1962, p.59.

GILLILAND, E.R., ROBINSON, C.S., Elements of Fractional Distillation. McGraw-Hill, 1950.

GREEN, S.J., VENER, R.E., Vapour-Liquid Equilibria of Formaldehyde-Methanol-Water. I.E.C., 47(1), 1955, p.103.

GUISTI, A.L., OTTO, R.E., WILLIAMS, T.J., Direct Digital Computer Control. Control Eng., Jun. 1962, p.104.

GUSTAVSSON, I., Survey of Applications of Identification in Chemical and Physical Processes. Automatica, 11, 1975, p.3.

HARRIOT, P., Process Control. McGraw-Hill, 1964.

HAY, J.M., JOHNSON, A.I., A Study of Sieve Tray Efficiencies. A.I.Ch.E.J. 6(3), 1960, p.373.

HOLLAND, C.D., Multicomponent Distillation. Prentice Hall, 1963.

HUCKABA, C.E. et al, Experimental Confirmation of a Predictive Model for Dynamic Distillation. C.E.P. Symp. Series 61(5), 1965, p.126.

HUCKABA, C.E., MAY, F.P., FRANKE, F.R., Analysis of Transient Conditions in Continuous Distillation Operations. C.E.P. Symp. Series 59(46), 1963, p.38.

HUGHES, H.E., MALONEY, J.O., The Application of Radioactive Tracers to Diffusional Operations. C.E.P. 48, 1952, p.192.

INTERNATIONAL CRITICAL TABLES, 5, 1928, p.115.

JAFRI, M.N., GLINSKY, G.S., WOOD, R.K., The Transient Behaviour and Control Characteristics of a Continuous Binary Column. Trans. I.Ch.E., 43, 1965, PT56.

KHRISHNAMOORTHY, V., EDGAR, T.F., Identification and Estimation of Discrete Models for Distillation Columns. Proc. JACC, 1977, p.857.

KUO, B.C., Analysis and Synthesis of Sampled Data Control Systems. Prentice-Hall, 1963.

KURZ, H., ISERMANN, R., Feedback Control Algorithms for Adaptive Control. Proc. JACC, 1977, p. 815.

LAMB, D.E., PIGFORD, R.L., RIPPEN, D.W.T., Dynamic Characterisation and Analog Simulation of Distillation Columns. C.E.P. Symp. Series 57(36), 1961, p.132.

LEWIS, W.K., MATHESON, G.L., Studies in Distillation. I.E.C., 24, 1932, p.494.

LUPFER, D.E., JOHNSON, M.L., Automatic Control of Distillation Columns to Achieve Optimum Operation. ISA Trans. 3(2), 1964, p.165.

LUYBEN, W.L., Feed Plate Manipulation in Distillation Feedforward Control. I.E.C. Fund. 7(3), 1968, p.502.

LUYBEN, W.L., GERSTER, J.A., Feedforward Control of Distillation Columns. I.E.C. Process Des. Dev. 3(4), 1964, p.374.

LUYBEN, W.L., Distillation Decoupling. A.I.Ch.E.J., 16(2), 1970, p.198.

MACMULLEN, E.C., SHINSKEY, F.G., Feedforward Analog Control Computer Control of a Superfractionator. Control Eng. March 1964, p.69.

MARINO, P.A. et al, Sinusoidal and Pulse Response of a Plate Distillation Column by Reflux Upset. A.I.Ch.E.J., 14(6), 1968, p.866.

MARTIN, J.J., Analytical Solution of Equilibrium Stage Operations. A.I.Ch.E.J., 9(5), 1963, p.646.

MASON, W.A., Number of Trays vs Reflux Ratio. Petr. Ref. 38(5), 1959, p.237.

MCCABE, W.L., THIELE, E.W., Graphical Design of Fractioning Columns. I.E.C., 17, 1925, p.605.

MEYER, C., SEBORG, D.E., WOOD, R.K., An Experimental Application of Time Delay Compensation Techniques to Distillation Control. I.E.C. Process Des. Dev., 17(1), 1978, p.62.

MEYER, C., WOOD, R.K., SEBORG, D.E., Experimental Evaluation of Analytical and Smith Predictors for Distillation Column Control. A.I.Ch.E.J. 25 (1), 1979, p.24.

MIKHAIL, S.Z., KIMEL, W.R., Densities and Viscosities of Methanol-Water Mixtures. J. Chem. Eng. Data 6(4), 1961, p.533.

MOCZEK, J.S., OTTO, R.E., WILLIAMS, T.J., Approximate Models for the Dynamic Response of Large Distillation Columns. C.E.P. Symp. Series 61(55), 1965, p.136.

MORRIN, P.C., Predictive Distillation Control Using an Analog Computer. B.E. Project Report, Univ. of Canterbury, 1975.

MOSLER, H.A., KOPPEL, L.B., COUGHANOWR, D.R., Process Control by Digital Compensation. A.I.Ch.E.J., 13(4), 1967, p.768.

MOSTAFA, H.A., Effect of Concentration on Distillation Plate Efficiency. Trans. I.Ch.E., 57(1), 1979, p.55.

MOTOROLA, Microprocessor Application Manual, 1975.

NATIONAL Semiconductor Transducer Catalog and Handbook, 1974.

NATIONAL Semiconductor MOS/LSI Databook, 1977, p.6-2.

OTHMER, D.F., BENENATI, R.F., Composition of Vapours from Boiling Binary Solutions. I.E.C., 37, 1945, p.299.

PERRY, J.H., Chemical Engineers Handbook. 4th Edition. McGraw-Hill, 1963.

PHILLIPS LTD., Okometer-R Industrial Refractometer Instruction Manual.

PLEWES, A.C., JARDINE, D.A., BUTLER, R.M., The Integral Heats of Vaporisation of Alcohol-Water Mixtures. Can. J. Tech. 32(4), 1954, p.133.

Q.V.F. Process Plant and Piping Catalog, 1970.

RADEMAKER, O., RIJNSDORP, J.E., MAARLEVELD, A., Dynamics and Control of Continuous Distillation Units. Elsevier Scientific, 1975.

ROSENBROCK, H.H. et al. Computers and Direct Digital Control. Control, March, 1965, p.125.

ROTAMETER MANUFACTURING CO. LTD., Calibration Charts for Metric Series Rotameters. 1960.

SARGENT, R.W.H., MURTAGH, B.A., The Design of Plate Distillation Columns for Multicomponent Mixtures. Trans. I.Ch.E., 47, 1969, T85.

SARIDIS, G.N., Comparison of Six-On-Line Identification Methods. Automatica, 10, 1974, p.69.

SARKARNY, G., ROZSA, P., TETTAMANTI, K., The Analytic Calculation of the Number of Theoretical Plates. Periodica Polytechnica, 1970, p.321.

SASTRY, V.A., SEBORG, D.E., WOOD, R.K., Self-Tuning Regulator Applied to a Binary Distillation Column. Automatica, 13, 1977, p.417.

SCHINDLER, M., Focus on Software. Electronic Design, 6, 1979, p.103.

SCHWANKE, C.O., EDGAR, T.F., HOUGEN, J.O., Problems with Application of Multivariable Control Theory to Distillation Columns. 69th Annual A.I.Ch.E. Meeting, Chicago, 1976.

SHIH, Y., Integral Action in the Optimal Control of Linear Systems with Quadratic Performance Index. I.E.C. Fund. 9(1), 1970, p.35.

SHINSKEY, F.G., Process Control Systems. McGraw-Hill, 1967.

SHUNTA, J.P., LUYBEN, W.L., Dynamic Effects of Temperature Control Tray Location in Distillation Columns. A.I.Ch.E.J., 17(1), 1971, p.92.

SHUNTA, J.P., LUYBEN, W.L., Sampled Data Feedback Control of a Binary Distillation Column. Can. J.Ch.E., 50, 1972, p.365.

SHUNTA, J.P., LUYBEN, W.L., Sampled Data Non-Interacting Control for Distillation Columns. Chem. Eng. Sci., 27, 1972, p.1325.

SINGH, R., Equilibrium Stages in Continuous Binary Distillation. Brit. Chem. Eng., 11(9), p.1025, 11(10), p.1226, 1966.

SKROKOV, M.R., The Benefits of Microprocessor Control. Chem. Eng., Oct. 11, 1976, p.133.

SMITH, C.L., Digital Computer Process Computer Control. Intext, 1972.

SUNDARESAN, K.R., KRISHNASWAMY, P.R., Estimation of Time Delay, Time Constant Parameters in Time, Frequency, and Laplace Domains. Can. J.Ch.E., 56, 1978, p.257.

SVRCEK, W.Y., Ph.D. Thesis, University of Alberta, Edmonton, 1967.

TAO, R.T., et al, Applications of Microprocessors in Control Problems. Proc. JACC, 1977, p.8.

THIELE, E.W., GEDDES, R.L., Computation of Distillation Apparatus for Hydrocarbon Mixtures. I.E.C., 25, 1933, p.289.

TOU, J.T., Digital and Sampled Data Control Schemes. McGraw-Hill, 1969.

TREYBAL, R.E., Mass Transfer Operations. 2nd Edition, McGraw-Hill, 1968.

UCHIDA, S., KATO, H., J. Chem. Ind., Japan, 37, 1934, p.525.

UNBEHAUEN, H., SCHMID, C.H.R., BOTTINGER, F., Comparison and Application of DDC Algorithms to a Heat Exchanger. Automatica, 12(5), 1976, p.393.

UNDERWOOD, A.J.V., Fractional Distillation of Multicomponent Mixtures: Calculation of Minimum Reflux Ratio. J. Inst. Petrol. 32, 1946, p.614.

VAN WINKLE, M., TODD, W.G., Optimum Fractionation Design by Simple Graphical Methods. C.E. Sept. 20, 1971, p.136.

WAHL, E.F., HARRIOT, P., Understanding and Prediction of the Dynamic Behaviour of Distillation Columns. I.E.C. Process Des. Dev., 9(3), 1970, p.396.

WANG, J.C., HENKE, G.E., Tridiagonal Matrix for Distillation. Hydro. Proc., 45(8), 1966, p.155.

WEIGAND, W.A. et al. Calculational Methods for the Response Time to Step Inputs for Approximate Models of Distillation Columns. A.I.Ch.E.J. 18, 1972, p.1243.

WELTY, J.R., WILSON. R.E., WICKS, C.E., Fundamentals of Momentum, Heat, and Mass Transfer. John Wiley, 1969.

WILLIAMS, T.J., The Status of Studies of the Dynamics of Mass Transfer Operations. C.E.P. Symp. Series 59(46), 1963, p.1.

WILLIAMSON, C.J., Verification of a Distillation Control Model. B.E. Project Report, Univ. of Canterbury, 1977.

WOOD, R.K., BERRY, M.W., Terminal Composition Control of a Binary Distillation Column. Chem. Eng. Sci. 28, 1973, p.1707.

WOOD, R.K., PACEY, W.C., Experimental Evaluation of Feedback, Feedforward and Combined Feedforward-Feedback Binary Distillation Column Control. Can. J.Ch.E., 50, 1972, p.376.

APPENDIX I

CIRCUIT DIAGRAMS

LIST OF FIGURES

Am9511 Arithmetic Processor Unit Interface

The Am9511 was designed to be compatible with the Intel 8080 family of microprocessors, and consequently, its timing signals were not consistent with the Motorola 6800 bus. A printed circuit board was developed to interface the Am9511 and a 1K byte EPROM to the microcomputer (figure 1).

The biggest timing problem was the correct arrangement of the read and write signals. It was necessary for the read and write signals to be delayed after the APU chip was selected for a period of not less than 50ns for the 2MHz Am9511. This delay was achieved with RC networks on the Schmidt inputs of 74LS221 monostables. The other major timing requirement of the read/write controls was that they turn off at least 60ns before the chip select line ($\overline{c/s}$) was released. The 74LS221 monostables were adjusted to ensure that this condition was met.

The microcomputer address bus was fully buffered and decoded. The desired memory addresses for the APU and EPROM were set by straps on the board. The APU was arranged to occupy two adjacent memory locations, and each of its four functions was selected by the A0 address line and the R/W line as shown in Table I-1.

TABLE I-1 APU ADDRESSING

Function	A0	R/W
Data write to APU	1	0
Data read from APU	1	1
Command to APU	0	0
Read APU status	0	1

The access time for a read on the APU stack was quoted as 2μs which was too slow for an M6800 operating at 1MHz. Hence the microprocessor was delayed by holding its ϕ_2 clock phase high until the data was available. A monostable triggered from the APU pause line held the $\overline{\text{MEM RDY}}$ line on the CPU board low for approximately 4µs, sufficient time for data from the APU to become available to the microprocessor (see figure 1A).

The clock for the APU was a free running astable multivibrator driving a one shot monostable. The astable was adjusted to provide the required frequency, while the one shot was adjusted to give the correct duty cycle. This arrangement was easier to set up than the conventional circuit using two one shots triggering each other.

All input and output lines were fully buffered using standard three state buffers. The data bus buffers were gated on to the data bus when the chip was addressed so that the buffer appropriate to the pending data transfer direction was selected.

Power for the APU board was obtained from the microcomputer power supply shown in figure 2. The supply rails were decoupled on board with suitable capacitors.

A 2708 EPROM was added to the board to store a control program for the APU. The circuit diagram of the board appears in figure 1, and the timing diagram for APU reads and writes is given in figure 1A.

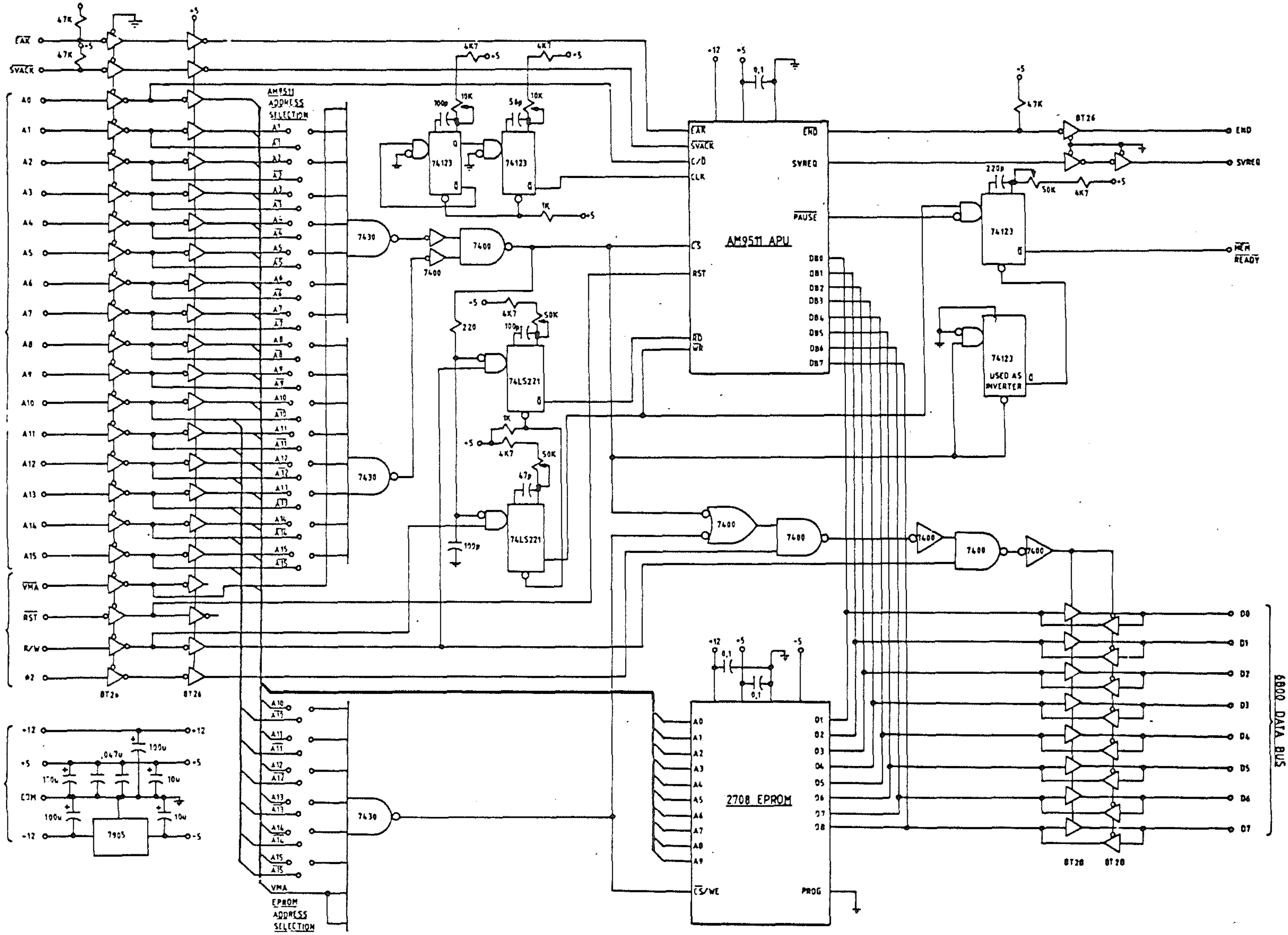
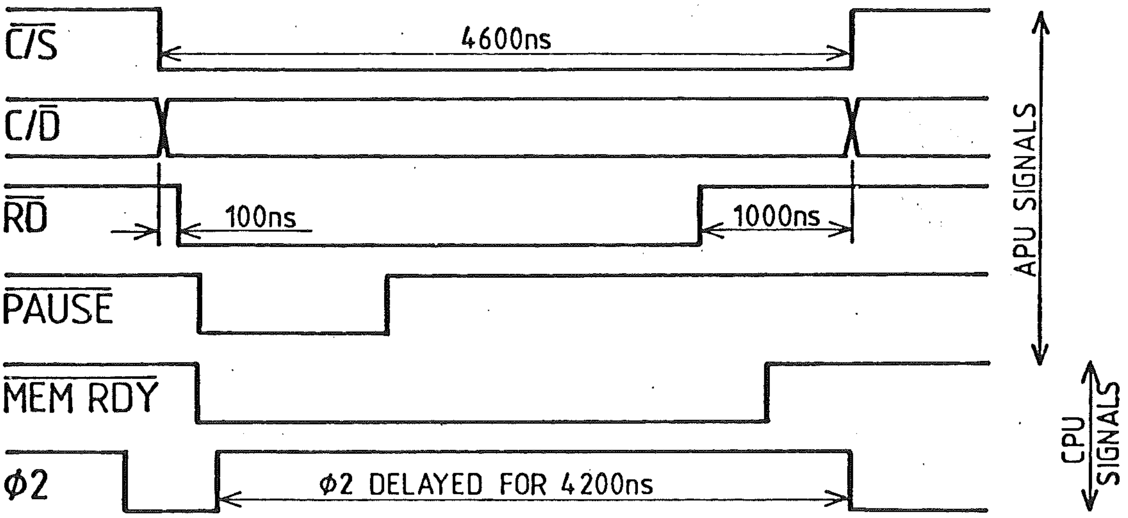
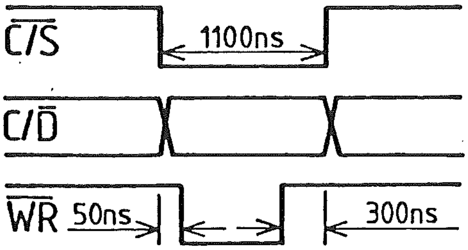


FIGURE 1 ARITHMETIC PROCESSOR BOARD



APU READ CYCLE



APU WRITE CYCLE

SCALE:3000ns

FIGURE 1A APU TIMING DIAGRAM

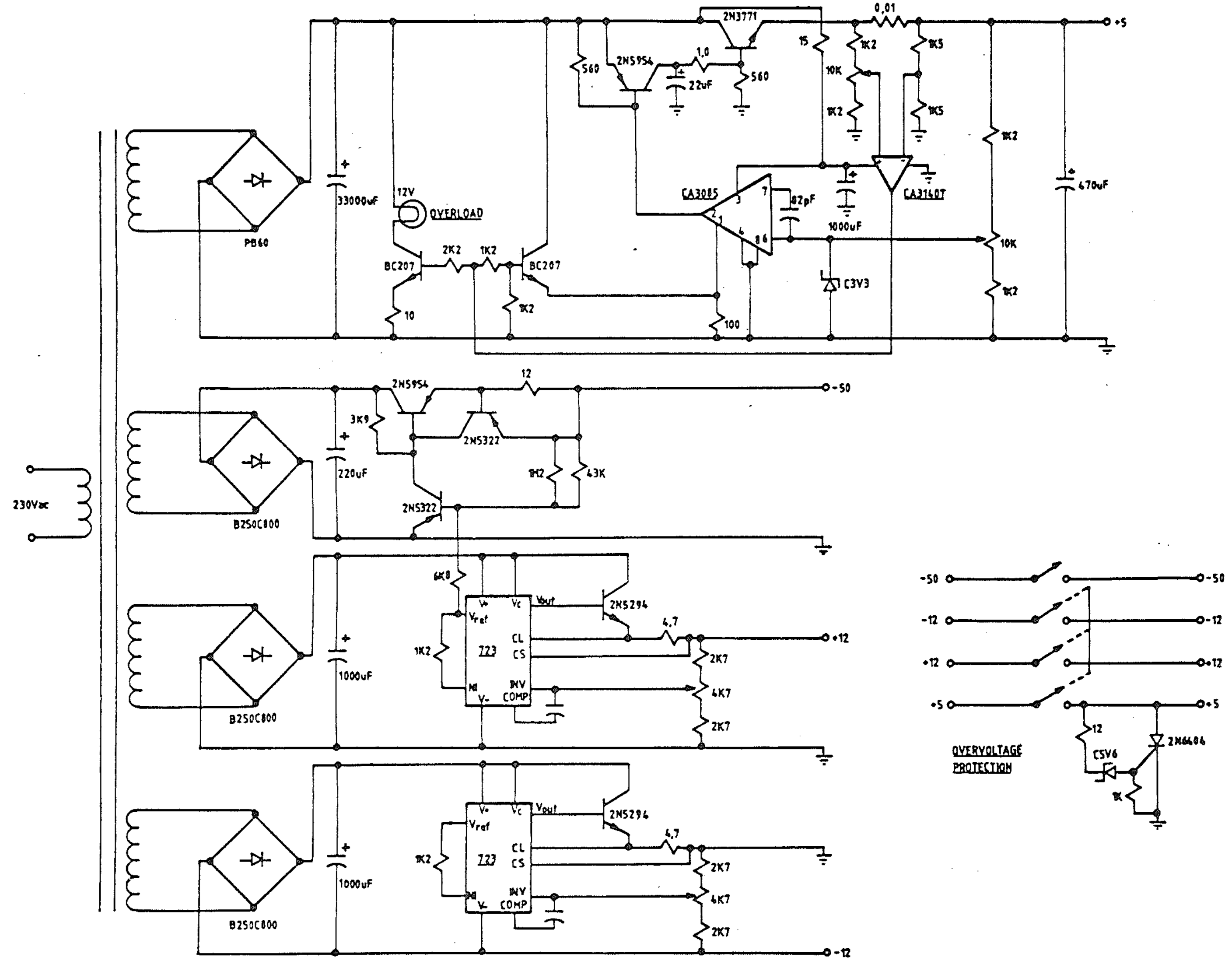


FIGURE 2 MICROCOMPUTER POWER SUPPLY

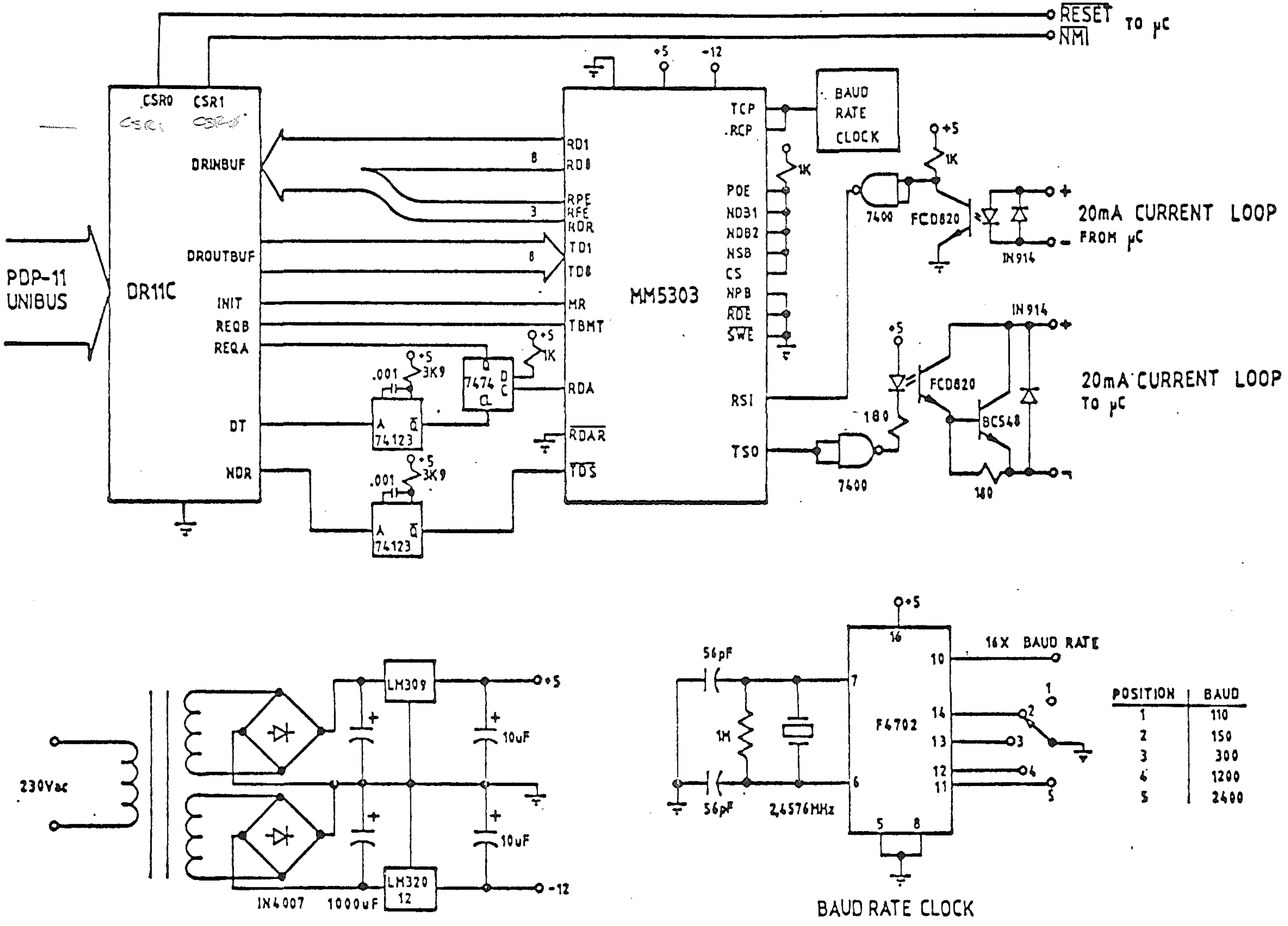
Minicomputer-Microcomputer Interface

A hardware interface was constructed using a UART to link a PDP-11 parallel interface (DR11) to the EVK300 board via a serial 20mA current loop as shown in figure 3. The interface was constructed using the standard Digital Equipment serial peripheral format (D.E.C. (1973)). Details of the DR11 interface can be found in the PDP-11 Peripherals Handbook (D.E.C. (1973)), and the UART details can be found in the National Semiconductor MOS/LSI Handbook (National Semiconductor (1977)).

For output from the PDP-11 to the microcomputer, data was strobed into the UART from the DR11 output register by the NEW DATA READY signal on the DR11. The UART then commenced to serialise and send the byte. When the UART output buffer was empty, it raised a control line which in turn set REQB in the DR11 status register indicating 'output buffer empty' to the PDP-11. By enabling the REQB interrupt the UART could be interrupt driven by the PDP-11.

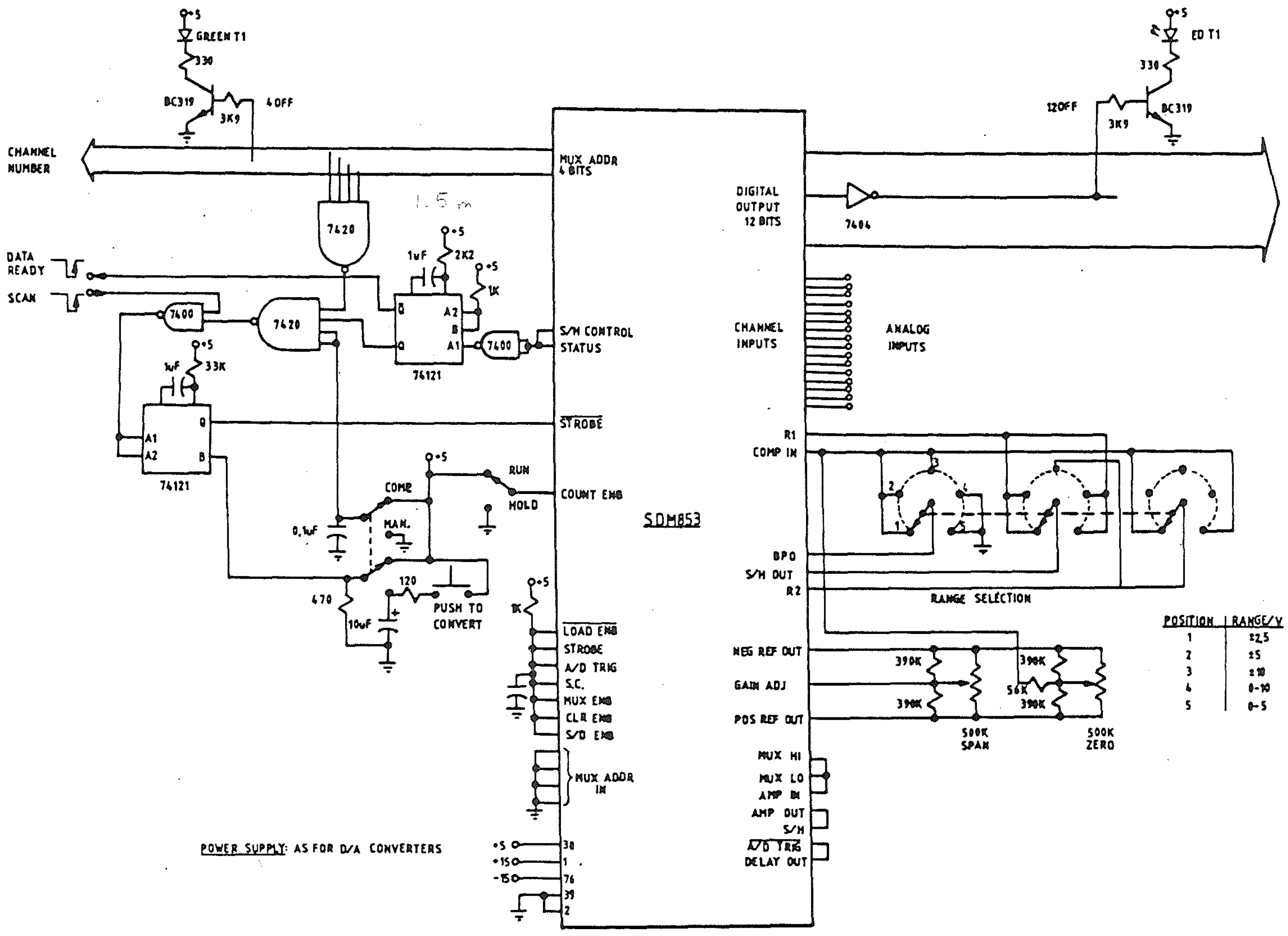
For input from the microcomputer to the PDP-11, the 'data received' line of the UART was connected to the REQA line of the DR11, and both raised high on receipt of a data byte. When the user program read the data byte from the DR11 input register, the DR11 DATA TAKEN line was used to clear the UART for receipt of the next character. The three UART error lines ROR, RFE and RPE were connected to bits 13, 14, 15 of the DR11 input register and were available to the users program. The receipt of characters could also operate under interrupt control. The two output control lines CSRO, CSR1 on the DR11 interface were connected to the $\overline{\text{RESET}}$ and $\overline{\text{NMI}}$ lines of the microcomputer to give the minicomputer overall control of the microcomputer.

See —
p111-6
[illegible]



POSITION	BAUD
1	110
2	150
3	300
4	1200
5	2400

FIGURE 3 MINICOMPUTER-MICROCOMPUTER INTERFACE



POSITION	RANGE/V
1	±2,5
2	±5
3	±10
4	0-10
5	0-5

FIGURE4 DATA ACQUISITION SYSTEM

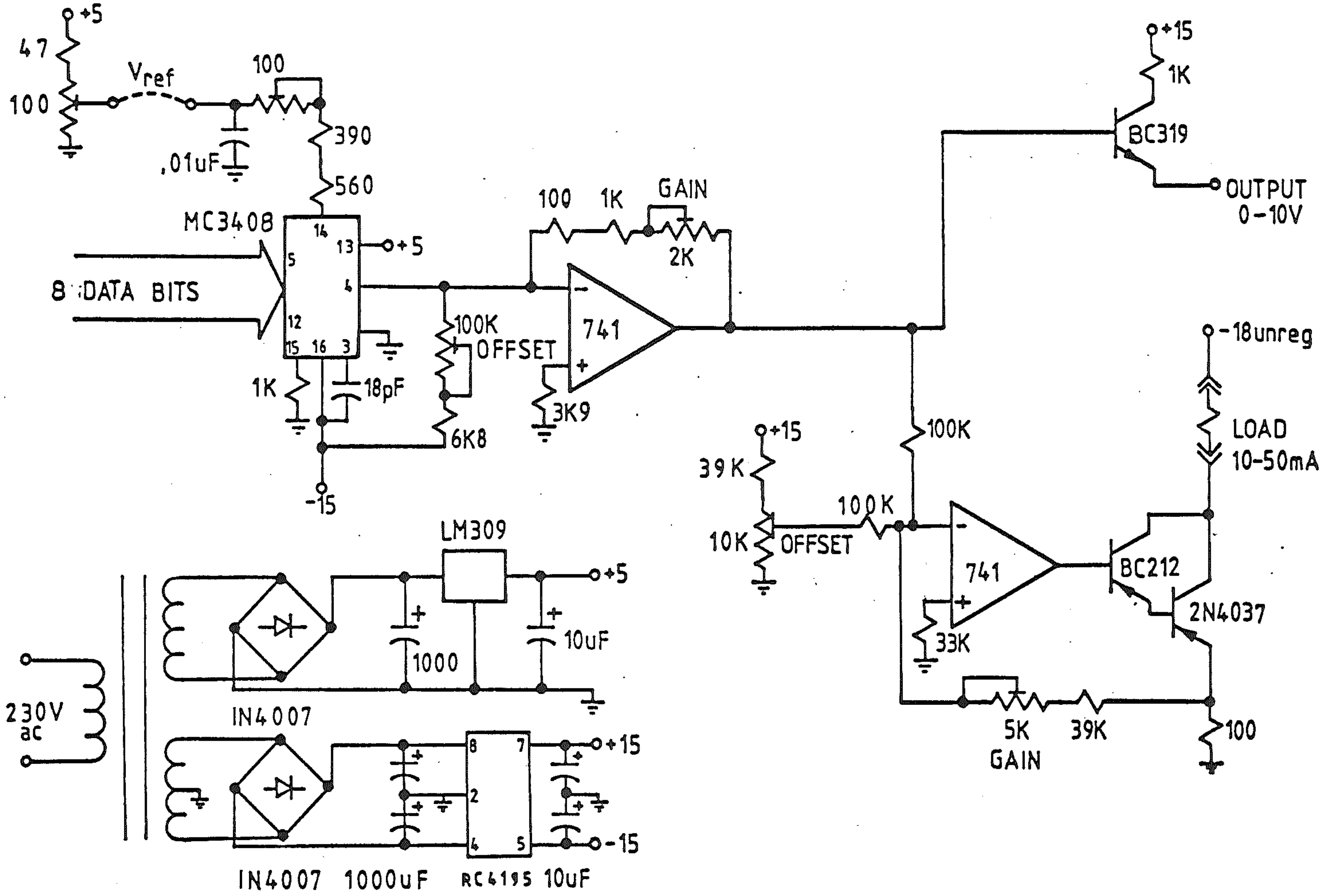
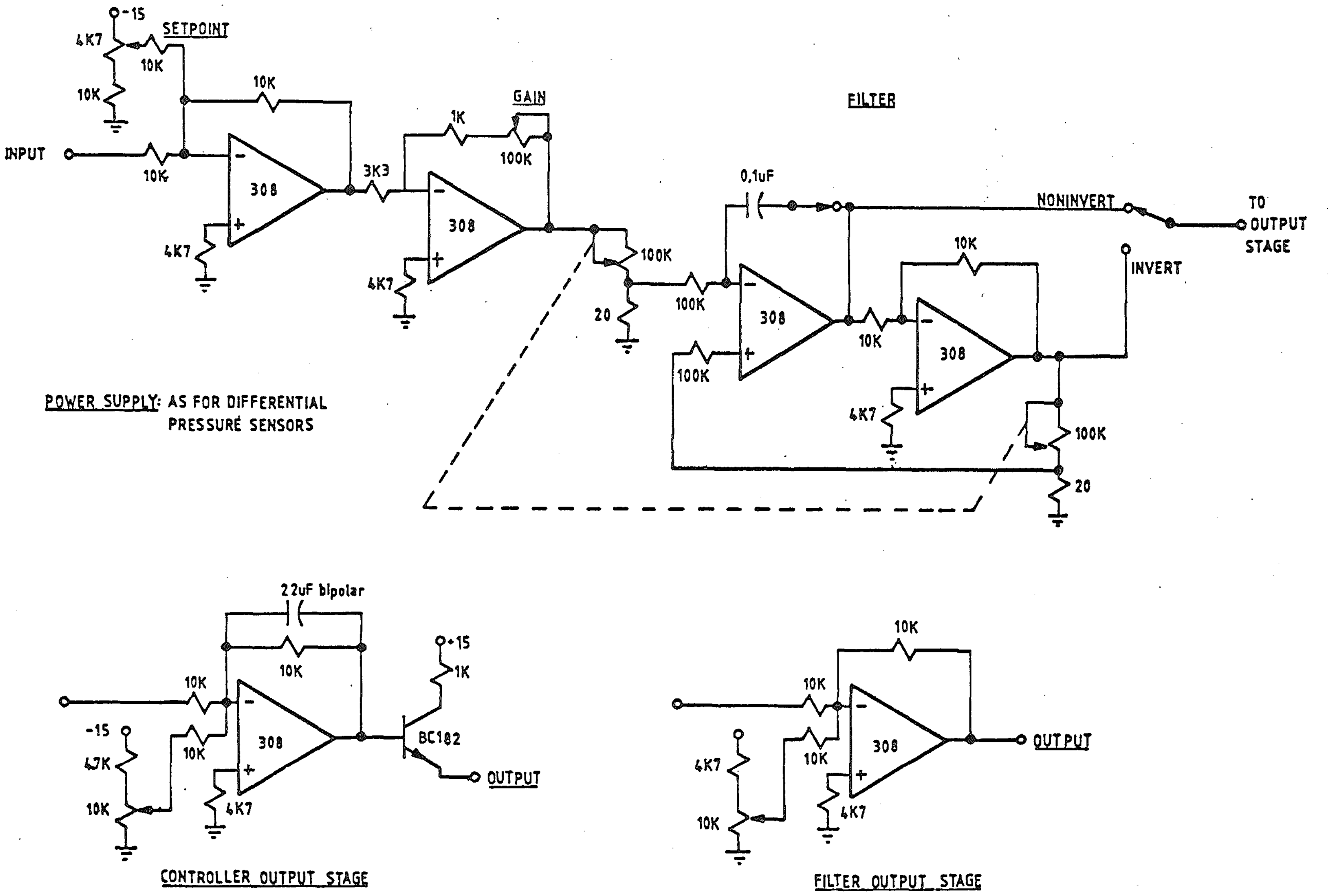


FIGURE 5 CONTROL OUTPUT SYSTEM



POWER SUPPLY: AS FOR DIFFERENTIAL PRESSURE SENSORS

CONTROLLER OUTPUT STAGE

FILTER OUTPUT STAGE

FIGURE 6 ANALOG CONTROLLER/FILTER

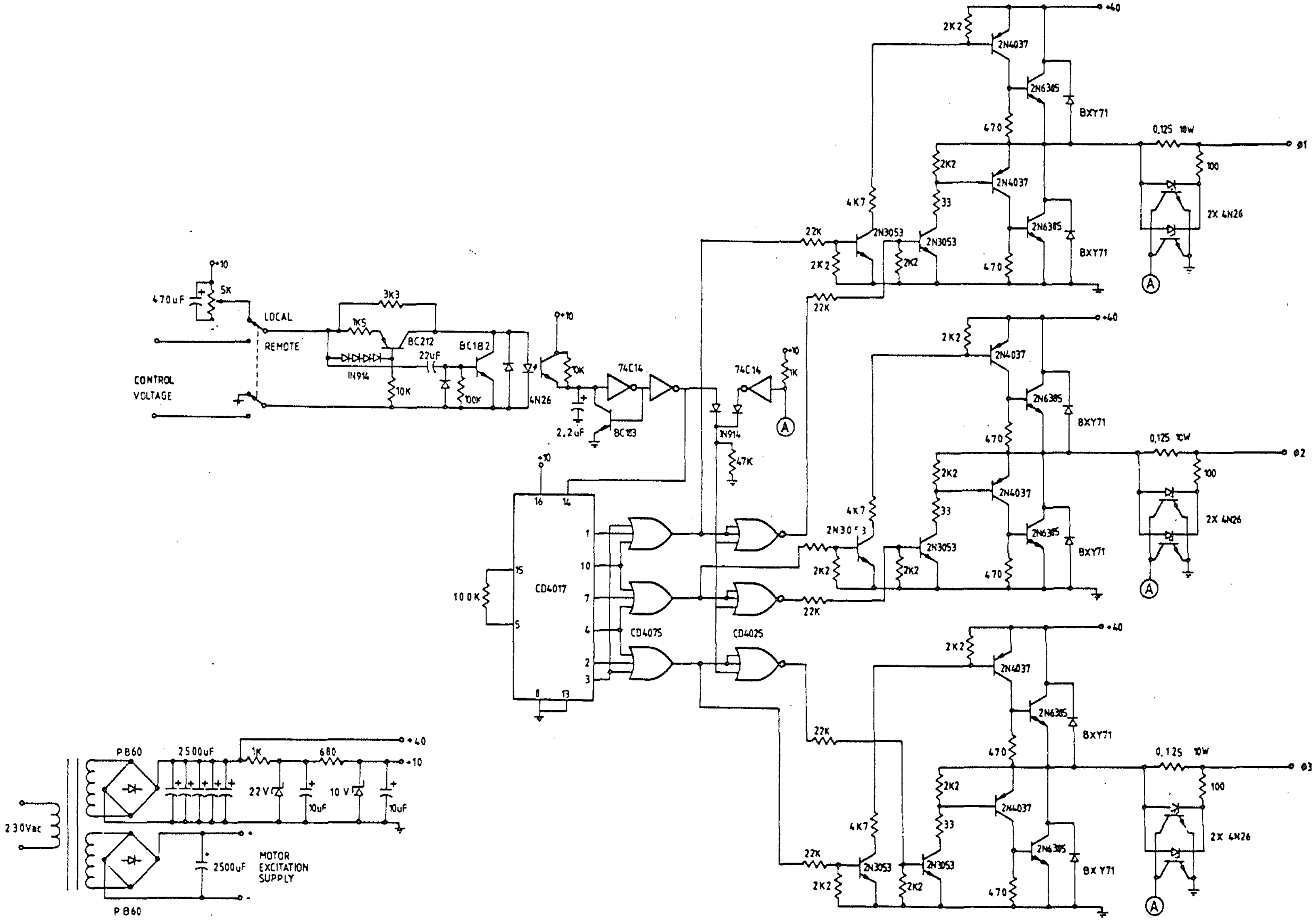


FIGURE7 MOTOR SPEED CONTROLLER (THREE PHASE INVERTER)

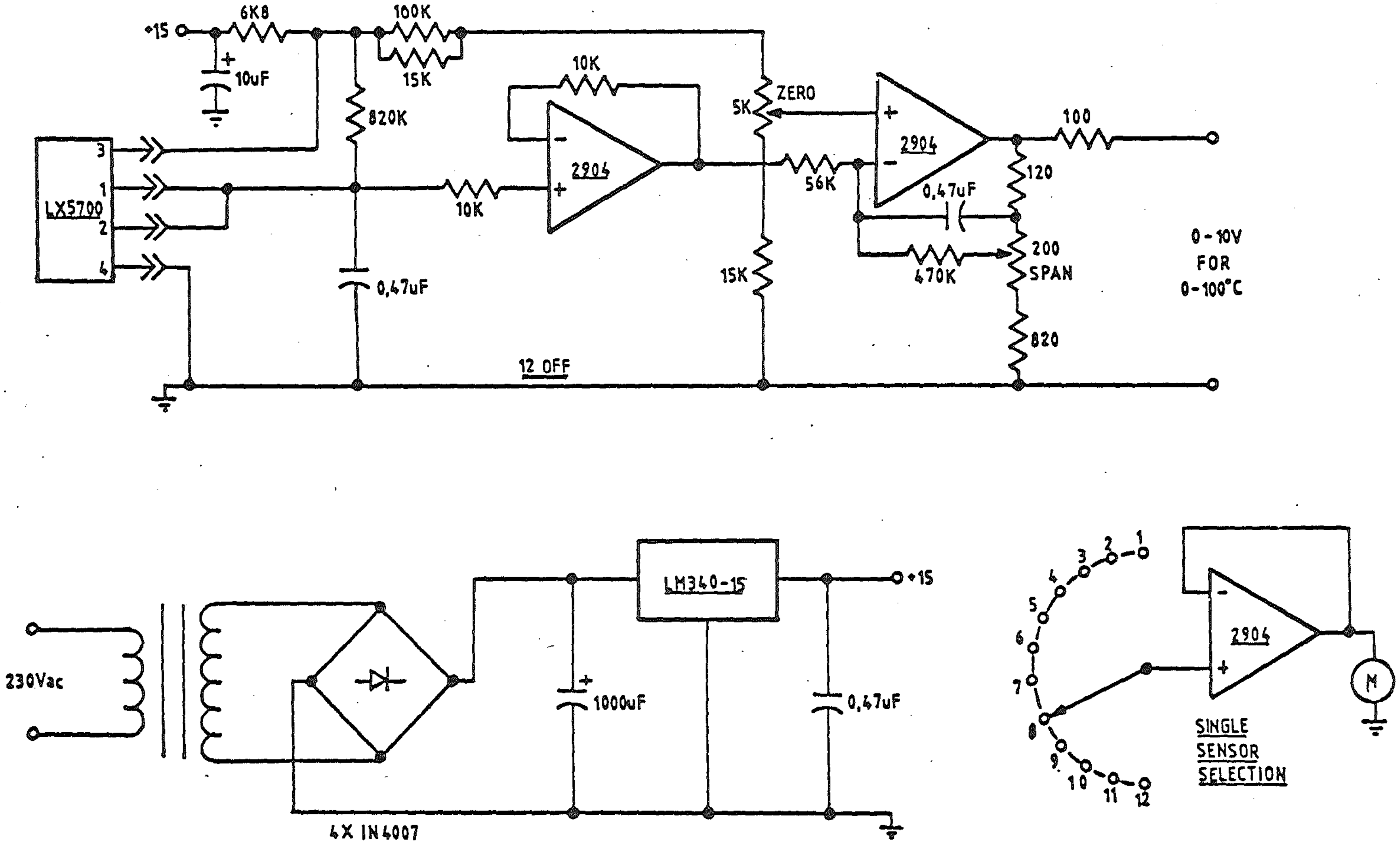


FIGURE8 TEMPERATURE SENSORS

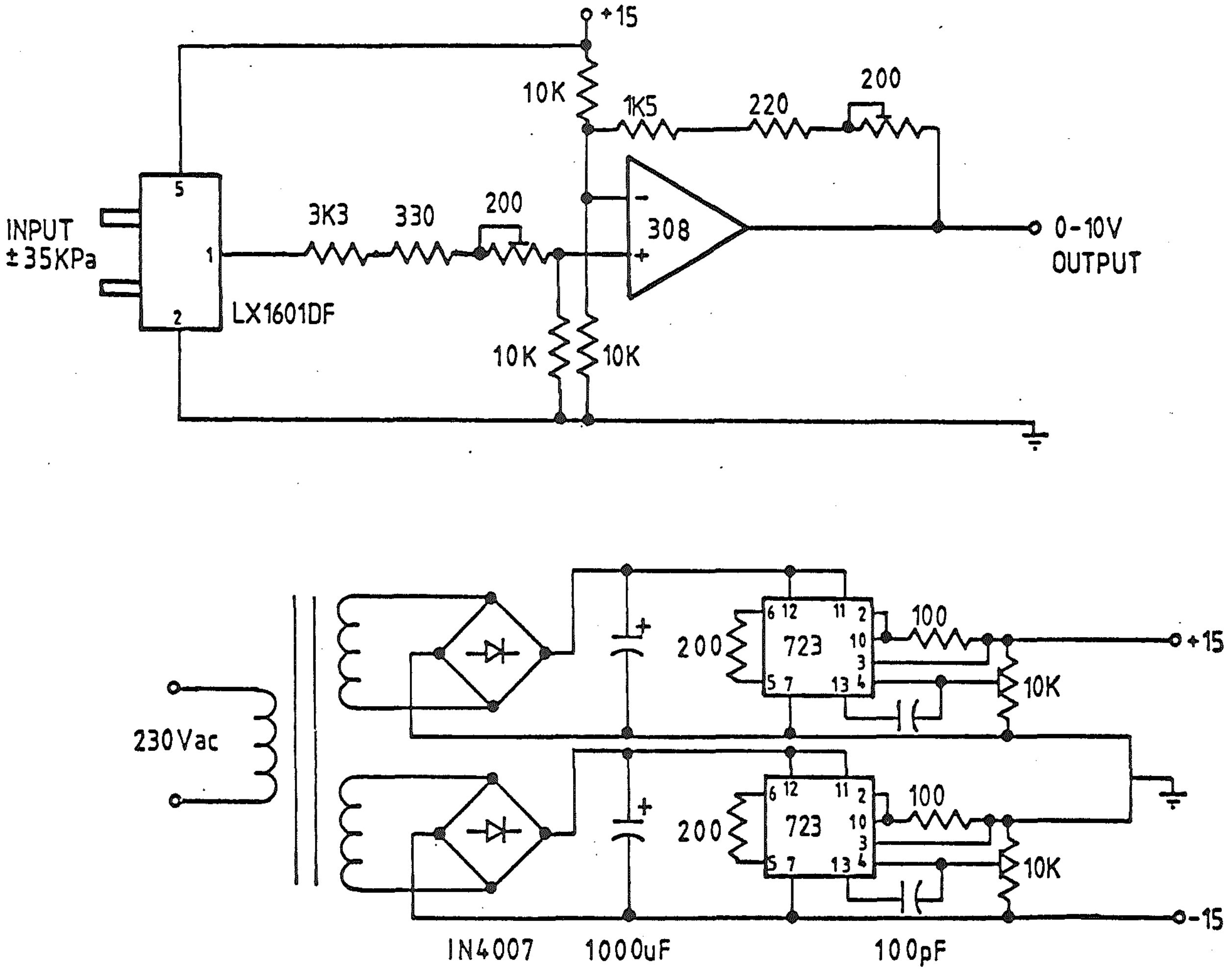
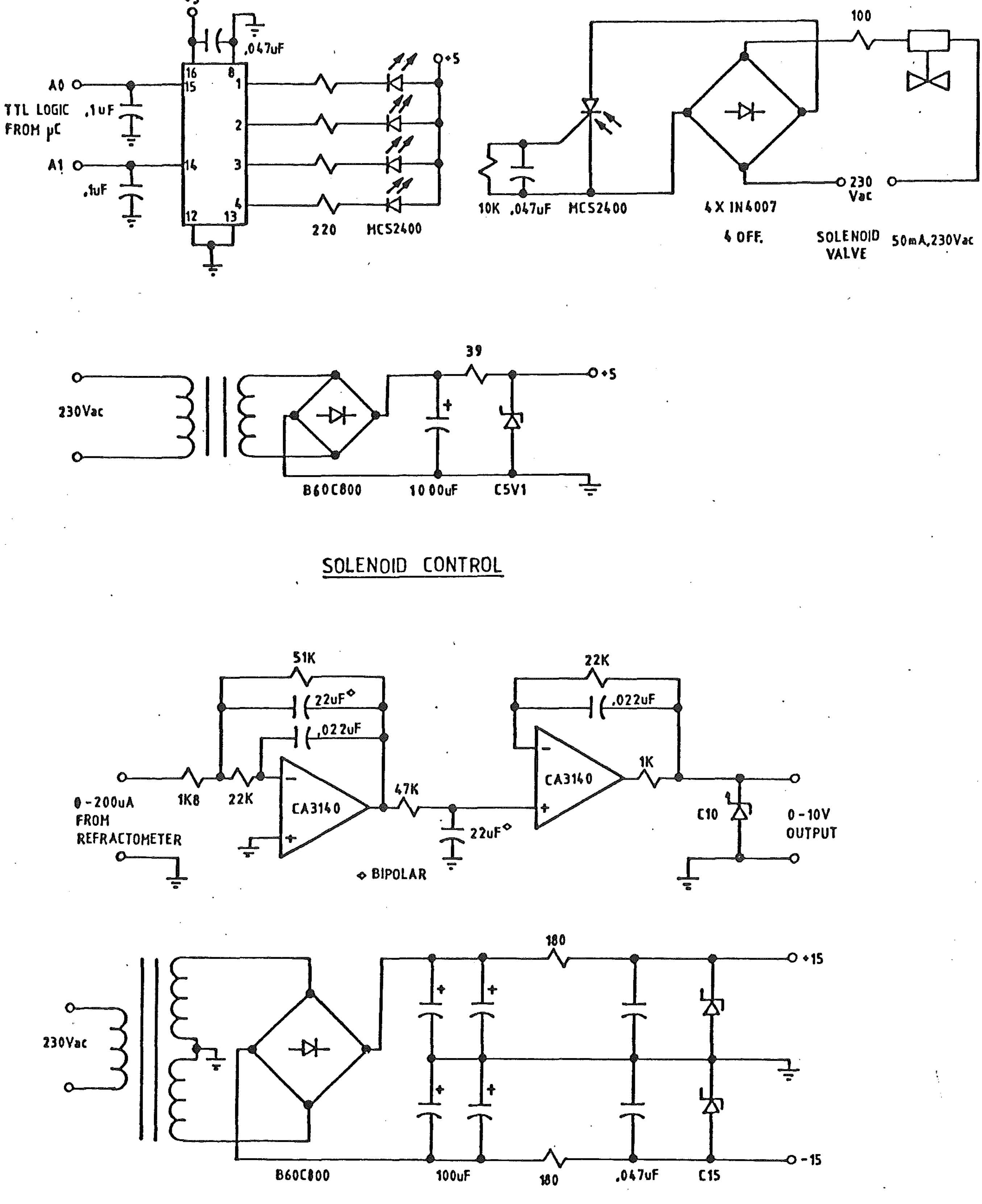


FIGURE 9 DIFFERENTIAL PRESSURE SENSORS



SOLENOID CONTROL

REFRACTOMETER AMPLIFIER

FIGURE 10 REFRACTOMETER ADDITIONS

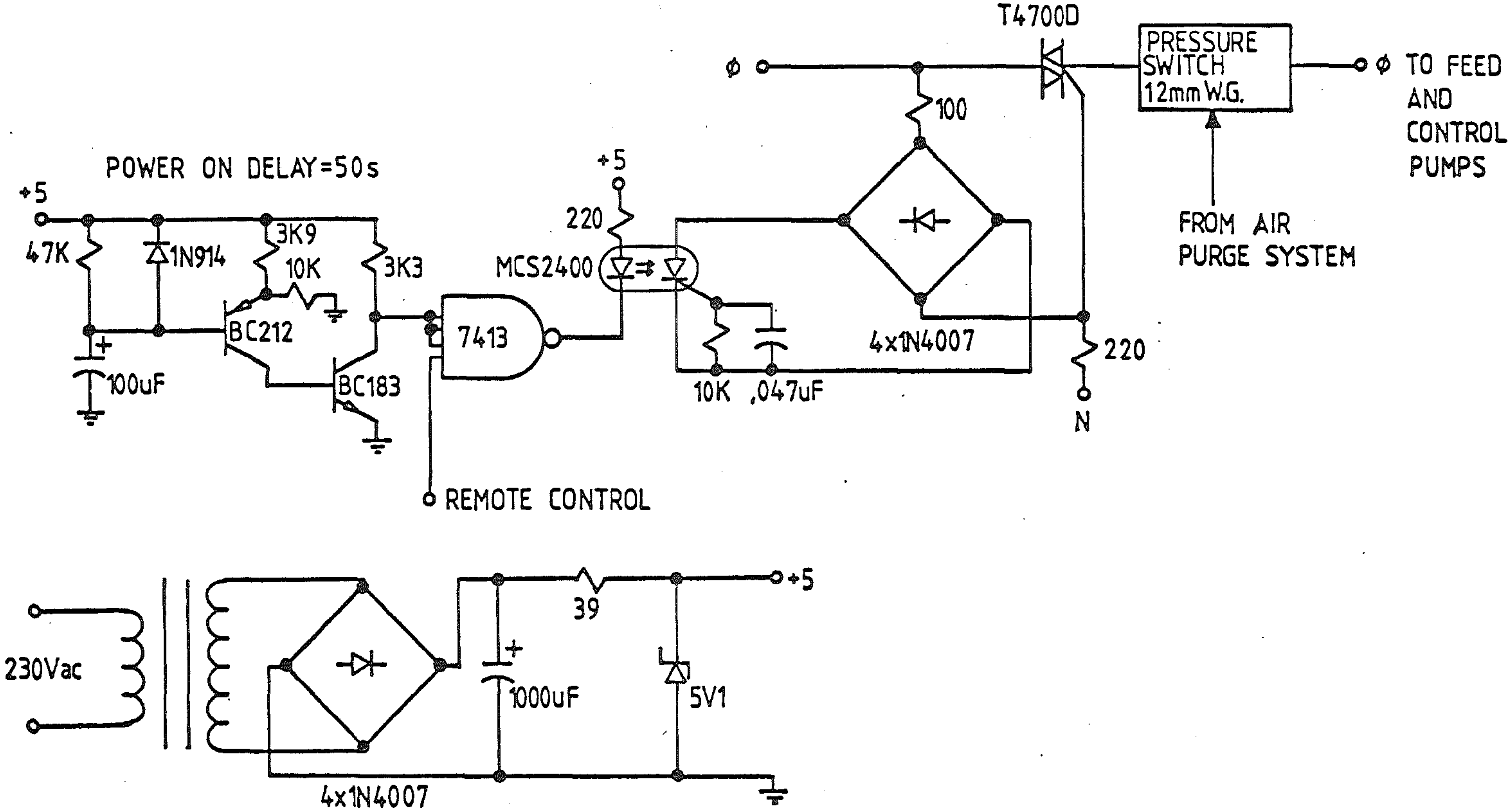


FIGURE 11 INTERLOCK SYSTEM FOR MOTOR PROTECTION

APPENDIX II

COLUMN CALIBRATIONS

II.1 TEMPERATURE PROBE CALIBRATION

The temperature probes were individually calibrated in a water bath using standard mercury/glass thermometers. The results were correlated using a quadratic polynomial with a maximum deviation of 0.2°C between the correlation and the experimental values.

LOCATION	$T = a + bV + cV^2$ V = measured voltage, T m°C		
	a	b	c
Tray 1	-.01	10.48	-.0456
2	-.01	10.34	-.0313
3	0	10.21	-.0206
4	0	10.22	-.0212
5	0	10.31	-.0303
6	0	10.46	-.0460
7	0	10.29	-.0285
8	0	10.27	-.0253
Reflux Drum	0.01	10.05	-.0684
Feed	0.0	10.19	-.0170
Refractometer Sample	0.0	10.19	-.0170

II.2 CALIBRATION OF ABBÉ REFRACTOMETER FOR METHANOL/WATER AT 25°C

Mole Fraction Methanol	Refractive Index
0	1.3328
.100	1.3368
.200	1.3399
.308	1.3413
.401	1.3413
.510	1.3401
.585	1.3389
.700	1.3367
.800	1.3334
.874	1.3310
1.00	1.3270

EFFECT OF TEMPERATURE ON REFRACTIVE INDEX OF METHANOL/WATER

		Refractive Index	
Mole Fraction Methanol	T = 20°C	T = 30°C	T = 40°C
0	1.3330	1.3324	1.3311
.104	1.3371	1.3359	1.3342
.207	1.3411	1.3390	1.3363
.308	1.3427	1.3400	1.3378
.404	1.3430	1.3401	1.3373
.516	1.3418	1.3390	1.3359
.622	1.3399	1.3372	1.3331
.709	1.3375	1.3357	1.3321
.803	1.3335	1.3323	1.3291
.937	1.3325	1.3294	1.3253
1.00	1.3300	1.3261	1.3322

Data measured using Abbé rectractometer

II.3 OKOMETER REFRACTOMETER CALIBRATION (25°C)

	Refractometer Output/V	Refractive Index	Mole Fraction Methanol
Distillate Range	1.00	1.3270	1.00
	1.51	1.3280	.965
	1.55	1.3285	.950
	1.68	1.3290	.940
	1.73	1.3292	.930
	1.86	1.3295	.920
	1.94	1.3298	.915
	2.05	1.3300	.910
	2.17	1.3302	.905
	2.43	1.3309	.885
	2.66	1.3311	.880
	2.77	1.3312	.875
	2.95	1.3320	.855
	3.29	1.3324	.840
	3.70	1.3327	.830
Feed Range	7.20	1.3387	.600
	7.63	1.3397	.535
	7.90	1.3405	.475
	8.11	1.3413	.400
	8.20	1.3414	.360
Bottoms Range	5.06	1.3332	.010
	5.33	1.3340	.020
	5.56	1.3345	.030
	5.69	1.3348	.040
	6.02	1.3351	.050
	6.40	1.3360	.075
	6.59	1.3362	.080

6.80	1.3369	.100
7.09	1.3375	.120
7.27	1.3380	.135
7.50	1.3388	.160
7.60	1.3392	.175
7.80	1.3399	.200

All measurements were made at 25°C. The refractive indices were measured on an Abbé refractometer, and the liquid compositions determined from the calibration data in Table II-2.

The data for the ranges 0 - 0.2, and 0.8 - 1.0 m.f. methanol were fitted with polynomials for ease of interpolation. The results were:

Range 0 - 0.2 m.f. methanol

$$Y = .344 - .150X + .0168X^2$$

Range 0.8 - 1.0 m.f. methanol

$$Y = 1.117 - .124X + 1.300E\text{-}2X^2$$

where Y = liquid mole fraction methanol

X = refractometer output/V

The corresponding calibrations for use in the microcomputer where 0-10V corresponds to 0-4095 are:

Range 0 - 0.2 m.f. methanol - bottoms composition

$$x_W = .344 - 3.666E\text{-}4*I + 1.001E\text{-}7*I^2$$

Range 0.8 - 1.0 m.f. methanol - distillate composition

$$x_D = 1.117 - 3.030E\text{-}4*I - 7.725E\text{-}8*I^2$$

I = refractometer data from data acquisition system.

II.4 ROTAMETER CALIBRATIONS

The rotameters in the bottoms, distillate and reflux lines were calibrated using water at 20°C. The calibrations were not in agreement with the correlations predicted by the manufacturer (Rotameter (1960)). A correction table was prepared to convert rotameter readings from water to methanol based on the manufacturers correlation. For intermediate compositions, linear interpolation based on the mixture mass fraction was used.

The experimental flow measurements were fitted with polynomials with a maximum deviation of ± .02 kg min^{-1}.

(i) Reflux Rotameter (Water @ 20°C)

Scale Reading/cm	Flow/kg min^{-1}
3.0 ± 0.1	.60 ± .02
4.0	.71 ± .02
5.0	.81 ± .02
6.0	.92 ± .02
7.0	1.02 ± .02
8.0	1.14 ± .02
9.0	1.26 ± .02
10.0	1.37 ± .03
12.0	1.61 ± .03
14.0	1.85 ± .03
16.0	2.12 ± .03
18.0	2.38 ± .03
20.0	2.66 ± .03
22.0	2.97 ± .03

Correlation: $\text{Flow/kg min}^{-1} = .0012x^2 + .0927x + .315$

x = rotameter reading/cm

(ii) Distillate Rotameter (Water @ 20°C)

Scale Reading/cm	Flow/kg min^{-1}
3.0 ± 0.1	.58 ± .01
4.0	.69 ± .02
5.0	.78 ± .02
6.0	.90 ± .02
7.0	.99 ± .02
8.0	1.11 ± .02
9.0	1.22 ± .03
10.0	1.35 ± .03
12.0	1.58 ± .03
14.0	1.84 ± .03
16.0	2.08 ± .03
18.0	2.35 ± .03
20.0	2.63 ± .03
22.0	2.85 ± .03

Correlation: Flow/kg min^{-1} = $.00089x^2 + .0992x + .267$

x = rotameter reading/cm

(iii) Bottoms Rotameter (Water @ 20°C)

Scale Reading/cm	Flow/kg min^{-1}
2.0 ± 0.1	.30 ± .01
3.2	.36 ± .02
4.0	.41 ± .02
5.0	.47 ± .02
6.0	.53 ± .02
7.0	.60 ± .02
8.0	.67 ± .02
9.0	.74 ± .02
10.0	.82 ± .02
12.0	.99 ± .02
14.0	1.18 ± .03
16.0	1.38 ± .03
18.0	1.60 ± .03
20.0	1.83 ± .03
22.0	2.08 ± .03

Correlation: Flow/kg min^{-1} = $.002x^2 + .0041x + .211$

x = rotameter reading/cm

(iv) Correction for Pure Methanol (@ 20°C) on Distillate and Reflux Rotameters

Scale Reading/cm	Additional Flow/kg min^{-1}
2.0	.02
3.0	.03
4.0	.04
5.0	.06
6.0	.07
7.0	.07
8.0	.06
9.0	.05
10.0	.03
11.0	.01

II.5 PUMP CALIBRATIONS

(i) Feed Pump-Calibrated Using Water at 20°C

Stroke Setting	Flow/kg min^{-1}
2.0 ± .01	.28 ± .05
3.0	.47
4.0	.63
5.0	.80
6.0	.98
7.0	1.18
8.0	1.36
9.0	1.52
10.0	1.76
11.0	1.90

Corrections for mixtures of methanol and water were made using the ratio of the mixture density to the density of water at 20°C, because the feed pump was a positive displacement piston pump.

(ii) Control Calibrations

Reflux Pump:

Microcomputer output byte = 383 L_V - 214

(0-255)

L_V = Reflux flow/ℓ min^{-1}

(0.90 m.f. methanol, 20°C)

Steam Flow:

Microcomputer output byte = 202 Q_S + 136

(0-255)

Q_S = Steam flow/kg min^{-1}

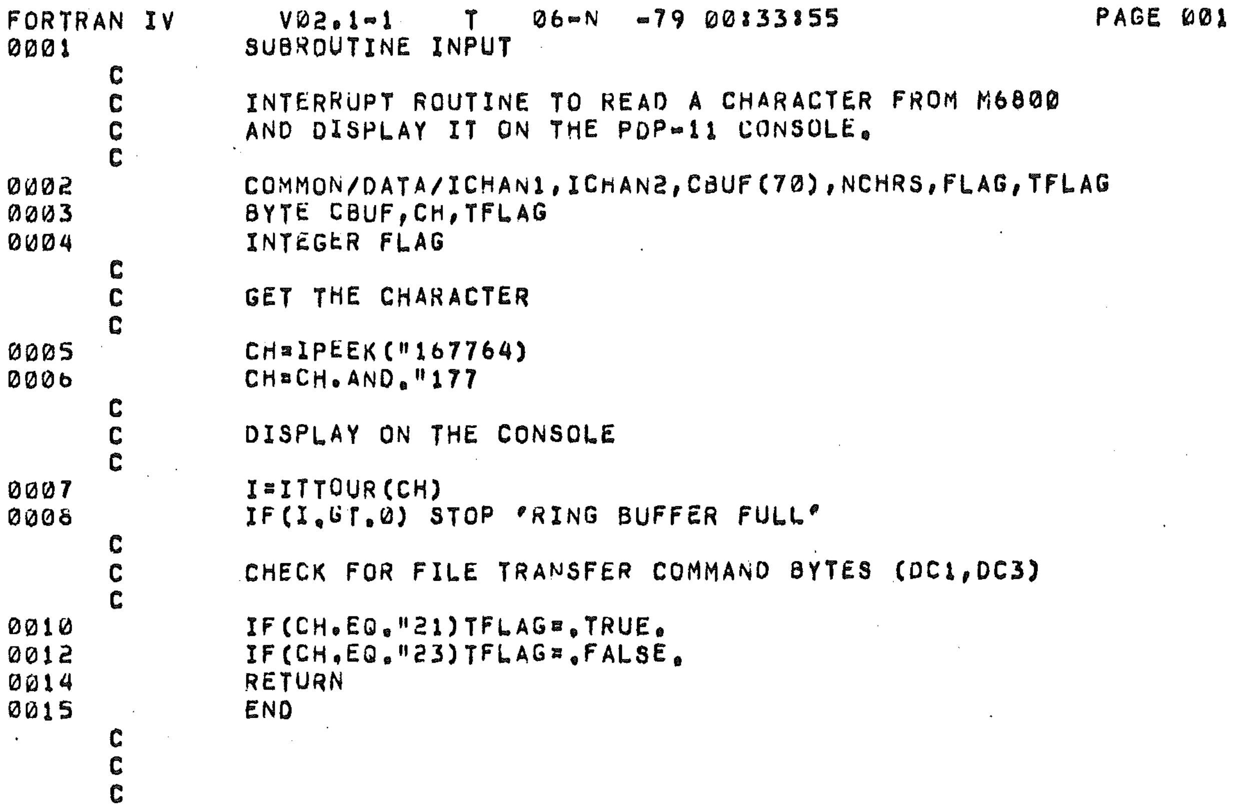
APPENDIX III

PDP-11 SOFTWARE

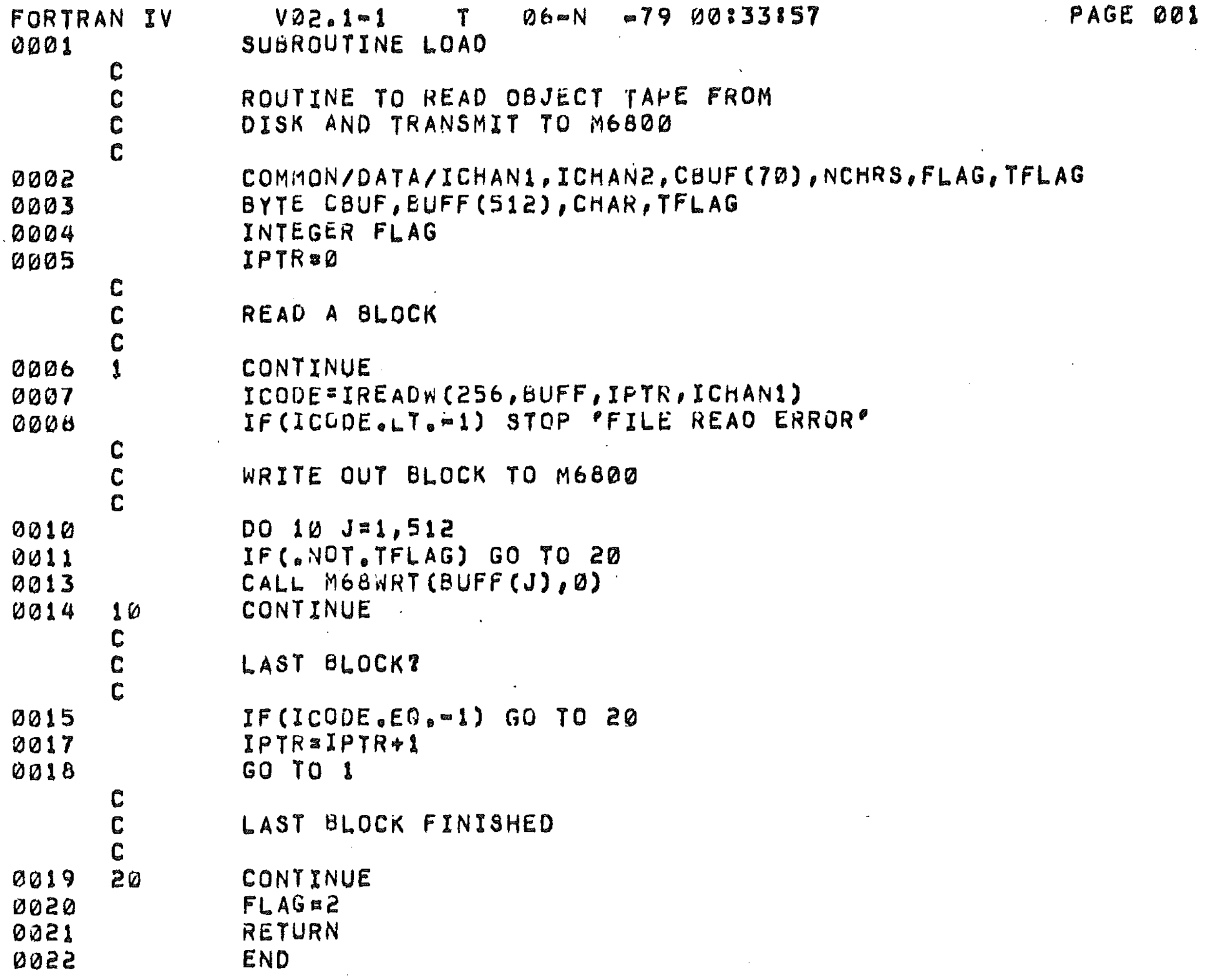
The program MGW was used to interface the PDP-11 minicomputer to the microcomputer via the hardware interface described in Appendix I. The program was written in RT-11 FORTRAN using the System Subroutine Library and an assembler subroutine. A listing follows.

```
FORTRAN IV      V02.1-1    T   06-N  -79 00:33:49                PAGE 001
      C
      C               *********
      C               *  MGW  *
      C               *********
      C
      C         A PROGRAM TO INTERFACE TO M6800
      C         VIA THE DR11/UART INTERFACE
      C
      C         GRANT WILSON
      C
      C         JULY 1978
      C
0001            COMMON/DATA/ICHAN1,ICHAN2,CBUF(70),NCHRS,FLAG,TFLAG
0002            BYTE CBUF,TITLE(12),CHAR,TFLAG
0003            INTEGER DBLK(4),FLAG,IASR(3)
0004            EXTERNAL INPUT
0005            DATA TITLE/'R','K','0',6*' ','L','D','A'/
0006            DATA IASR/"167760,3,0/
0007            CALL IPOKE("167760,3)
0008            FLAG=0
0009            TFLAG=.FALSE.
0010            CALL DEVICE(IASR)
0011            TYPE 9999
0012  9999      FORMAT(/' PROGRAM RUNNING')
      C
      C         SET UP AND ENABLE INTERRUPT ROUTINE
      C
0013            I=INTSET("320,7,0,INPUT)
0014            IF(I.NE.0) STOP 'INT SETUP ERR'
0016            CALL IPOKE("167760,"103)
      C
      C         WAIT HERE FOR COMMAND STRING
      C
0017  2         READ(5,100) NCHRS,(CBUF(J),J=1,NCHRS)
0018  100       FORMAT(Q,70A1)
      C
      C         CHECK FOR RESET,NMI AND STOP
      C
0019            IF(NCHRS.LT.2) GO TO 3
0021            IF(CBUF(1).NE.'R'.OR.CBUF(2).NE.'E') GO TO 3
0023            CALL M68RST
0024            GO TO 2
0025  3         IF(CBUF(1).EQ.'S'.AND.CBUF(2).EQ.'T') GO TO 999
0027            IF(CBUF(1).NE.'N'.OR.CBUF(2).NE.'M') GO TO 4
0029            CALL M68NMI
0030            GO TO 2
      C
      C         CHECK STRING FOR "L" OR "."
      C
0031  4         IF(CBUF(1).NE.'L') GO TO 10
      C
      C         GET FILE NAME OF DISK FILE,AND LOOK IT UP
      C
0033            WRITE(7,200)
0034  200       FORMAT(/' ENTER FILE NAME: DEV=RK0,EXT=LDA ')
0035            DO 5 K=4,9
0036  5         TITLE(K)="40
0037            READ(5,202)NT,(TITLE(J),J=4,3+NT)
```

```
FORTRAN IV      V02.1-1    T   06-N  -79 00:33:49                  PAGE 002
0038   202      FORMAT(Q,12A1)
0039            CALL IRAD50(12,TITLE,DBLK)
0040            IF(IFETCH(DBLK).NE.0) STOP 'BAD FETCH'
       C        ... GET CHANNEL FOR DISK READ ...
0042            ICHAN1=IGETC(M)
0043            IF(ICHAN1.EQ.-1) STOP 'NO CHANNELS AVAILABLE'
0045            IF(LOOKUP(ICHAN1,DBLK).LT.0) STOP 'BAD LOOKUP'
0047            FLAG=1
       C
       C        OUTPUT COMMAND STRING TO M6800
       C
0048   10       CONTINUE
0049            CBUF(NCHRS+1)="15
0050            DO 20 J=1,NCHRS+1
0051            CALL M68WRT(CBUF(J),0)
0052   20       CONTINUE
       C
       C        CHECK FOR FILE TRANSFER COMPLETE,IF SO
       C        THEN CLOSE THE FILE.
       C
0053            IF(FLAG.EQ.0) GO TO 2
0055   24       IF(FLAG.EQ.1.AND..NOT.TFLAG) GO TO 24
0057            CALL LOAD
0058            FLAG=0
0059            CALL CLOSEC(ICHAN1)
0060            CALL IFREEC(ICHAN1)
0061            GO TO 2
       C
       C        DISABLE INTERRUPTS ON FINISH
       C
0062   999      CONTINUE
0063            CALL IPOKE("167760,2)
0064            STOP 'PROGRAM TERMINATED'
0065            END
       C
       C
       C
```



```
FORTRAN IV      V02.1-1    T   06-N  -79 00:33:55               PAGE 001
0001            SUBROUTINE INPUT
      C
      C         INTERRUPT ROUTINE TO READ A CHARACTER FROM M6800
      C         AND DISPLAY IT ON THE PDP-11 CONSOLE.
      C
0002            COMMON/DATA/ICHAN1,ICHAN2,CBUF(70),NCHRS,FLAG,TFLAG
0003            BYTE CBUF,CH,TFLAG
0004            INTEGER FLAG
      C
      C         GET THE CHARACTER
      C
0005            CH=IPEEK("167764)
0006            CH=CH.AND."177
      C
      C         DISPLAY ON THE CONSOLE
      C
0007            I=ITTOUR(CH)
0008            IF(I.GT.0) STOP 'RING BUFFER FULL'
      C
      C         CHECK FOR FILE TRANSFER COMMAND BYTES (DC1,DC3)
      C
0010            IF(CH.EQ."21)TFLAG=.TRUE.
0012            IF(CH.EQ."23)TFLAG=.FALSE.
0014            RETURN
0015            END
      C
      C
      C
```



```
FORTRAN IV      V02.1-1    T   06-N   -79 00:33:57                  PAGE 001
0001             SUBROUTINE LOAD
      C
      C          ROUTINE TO READ OBJECT TAPE FROM
      C          DISK AND TRANSMIT TO M6800
      C
0002             COMMON/DATA/ICHAN1,ICHAN2,CBUF(70),NCHRS,FLAG,TFLAG
0003             BYTE CBUF,BUFF(512),CHAR,TFLAG
0004             INTEGER FLAG
0005             IPTR=0
      C
      C          READ A BLOCK
      C
0006  1          CONTINUE
0007             ICODE=IREADW(256,BUFF,IPTR,ICHAN1)
0008             IF(ICODE.LT.-1) STOP 'FILE READ ERROR'
      C
      C          WRITE OUT BLOCK TO M6800
      C
0010             DO 10 J=1,512
0011             IF(.NOT.TFLAG) GO TO 20
0013             CALL M68WRT(BUFF(J),0)
0014  10         CONTINUE
      C
      C          LAST BLOCK?
      C
0015             IF(ICODE.EQ.-1) GO TO 20
0017             IPTR=IPTR+1
0018             GO TO 1
      C
      C          LAST BLOCK FINISHED
      C
0019  20         CONTINUE
0020             FLAG=2
0021             RETURN
0022             END
```

```
MICRO AND INTERFACE CONTROL     MACRO V03.0286-NOV-79 00:08:49 PAGE 1

                                .LIST   TTM
                                .TITLE  MICRO AND INTERFACE CONTROL
                        ;
                        ;       TO CONTROL THE 6800 MICROCOMPUTER
                        ;
                        ;       CALLS: CALL M68RST - RESETS THE MICRO
                        ;              CALL M68NMI - ISSUES NMI INT TO MICR(
                        ;
                                .GLOBL  M68RST,M68NMI
                167760  MICROCS=167760
                        ;
                        ;       RESET CONTROL ROUTINE
                        ;
        000000  042737  M68RST: BIC     #2,@#MICROCS    ;TURN CSR1 OFF
                000002
                167760
        000006  012700          MOV     #10000.,R0      ;WAIT A WHILE
                023420
        000012  000240  1$:     NOP
        000014  005300          DEC     R0
        000016  001375          BNE     1$
        000020  052737          BIS     #2,@#MICROCS    ;TURN CSR1 ON
                000002
                167760
        000026  000207          RTS     PC
                        ;
                        ;       NMI CONTROL ROUTINE
                        ;
        000030  042737  M68NMI: BIC     #1,@#MICROCS    ;TURN CSR0 OFF
                000001
                167760
        000036  012700          MOV     #100.,R0        ;WAIT A WHILE
                000144
        000042  000240  2$:     NOP
        000044  005300          DEC     R0
        000046  001375          BNE     2$
        000050  052737          BIS     #1,@#MICROCS    ;TURN CSR0 ON
                000001
                167760
        000056  000207          RTS     PC
                        ;
                        ;       TO SEND DATA TO THE M6800 MICROCOMPUTER
                        ;       ATTACHED TO THE PDP-11 VIA A DR11+UART
                        ;
                        ;       CALL: CALL M68WRT(B1,B2,...,0)
                        ;       B1,B2 = BYTES OF DATA TERMINATED WITH ZERO
                        ;
                                .MCALL  .RCTRLO,.PRINT
                                .GLOBL  M68WRT
                167760  UARTCS=167760
                167762  UARTWR=UARTCS+2
                        ;
        000060  005725  M68WRT: TST     (R5)+           ;BUMP UP R5
        000062  005000  1$:     CLR     R0
        000064  013701  2$:     MOV     @#UARTCS,R1     ;GET UART C/S
                167760
        000070  100411          BMI     3$              ;READY ?
```

```
MICRO AND INTERFACE CONTROL      MACRO V03.0286-NOV-79 00:08:49 PAGE 1-1

     47 000072  005200            INC     R0                ;NO
     48 000074  020027            CMP     R0,#32000.        ;LONG ENOUGH ?
                076400
     49 000100  000771            BR      2$                ;NO
     50 000102                    .RCTRLO                   ;YES - PRINT MSG
     51 000104                    .PRINT  #MSG
     52 000112  000207            RTS     PC
     53 000114  113537      3$:   MOVB    @(R5)+,@#UARTWR   ;SEND DATA BYTE
                167762
     54 000120  105775            TSTB    @(R5)             ;CHECK NEXT BYTE
                000000
     55 000124  001356            BNE     1$                ;<>0 CONTINUE
     56 000126  000207            RTS     PC
     57                     ;     ERROR MESSAGE
     58 000130     077      MSG:  .ASCIZ  /? MICRO - NO RESPONSE ?/
        000131     040
        000132     115
        000133     111
        000134     103
        000135     122
        000136     117
        000137     040
        000140     055
        000141     040
        000142     116
        000143     117
        000144     040
        000145     122
        000146     105
        000147     123
        000150     120
        000151     117
        000152     116
        000153     123
        000154     105
        000155     040
        000156     077
        000157     000
     59                           .EVEN
     60         000001            .END
```

APPENDIX IV

MICROCOMPUTER SOFTWARE

IV.1 OPSYS

OPSYS is a skeletal operating system as described in Chapter 5. A listing follows.

```
        NAM     MICROCOMPUTER OPERATING SYSTEM
*
*       MINI OPERATING SYSTEM FOR
*       M68ØØ MICROCOMPUTER.
*       RESIDES IN EPROM #3
*       RESET START IS $E6ØØ
*       NMI START IS $E63Ø
*
*       GRANT WILSON
*       3Ø/3/79
*       MODIFIED VERSION
*       ADDING CONFIGURE,JUMP-TO-PROTO,ZERO RAM
*       ROUTINES. CAN BE USED WITH
*       A TERMINAL OR THE PDP-11.
*
*       DEFINITIONS
*
ACIAC   EQU     $FBCE           ACIA CONTROL REGISTER
ACIAD   EQU     $FBCF           ACIA DATA REGISTER
ACIAS   EQU     $FBCE           ACIA STATUS REGISTER
SWI3Ø   EQU     $FØD1           USER SWI HANDLER
SWIHAN  EQU     $FØBE           RSRSR SWI HANDLER
PROTO   EQU     $FØ4A           PROTO START ADDRESS
*
*       RSRSR DEFINITIONS
*
TABX    EQU     $Ø3
P2HEX   EQU     $ØF
P4HEX   EQU     $1Ø
PRINTA  EQU     $11
PMSG    EQU     $12
VALAN   EQU     $13
INPUTA  EQU     $14
CONHB   EQU     $15
*
*       PIA ASSIGNMENTS
*
*       D.A.M. MODÚLE
DMSB    EQU     $FBCA
DCSR1   EQU     DMSB+1
DLSB    EGU     DCSR1+1
DCSR2   EQU     DLSB+1
*       D/A PIAS
V4      EQU     $FBCØ
V3      EQU     V4+2
V2      EQU     V3+2
V1      EQU     V2+2
V4CSR   EQU     V4+1
V3CSR   EQU     V3+1
V2CSR   EQU     V2+1
V1CSR   EQU     V1+1
```

```
*
*
*                      MONITOR RAM
*                THIS DATA BASE FOLLOWS THE SAME
*              LAYOUT AS THE AMI PROTO DATA BASE.
*
        ORG     $FF90
BASE    EQU     *
BOS     EQU     *-1             BOTTOM OF STACK
BUF     RMB     72              LINE OF INPUT BUFFER
PROMAD  EQU     *
OFFSET  RMB     2
TADR    EQU     *               TAPE ADDRESS
ADR     RMB     2
ADDL    RMB     2
ADDH    RMB     2
BUFPTR  RMB     2
RECTYP  RMB     1               RECORD TYPE
COUNT   RMB     1               BYTE COUNT
CKSM    RMB     1               CALCULATED CKSM
SAVESP  RMB     2
SAVEX   RMB     2
ECHO    RMB     1
TCOUNT  RMB     1
*       USER REGISTERS (NOT USED BY THIS SYSTEM)
CREG    RMB     1
BREG    RMB     1
AREG    RMB     1
XREG    RMB     2
PREG    RMB     2
SREG    RMB     2
*       INTERRUPT VECTORS
USWI    RMB     2               USER SWI VECTOR
ACIAI   RMB     2               INDIRECT POINTER TO ACIA
IRQVEC  RMB     2               IRQ VECTOR
SWIVEC  RMB     2               SWI VECTOR
NMIVEC  RMB     2               NMI VECTOR
```

```
*
*                **************
*                * MAIN PROGRAM *
*                **************
*
*
*       RESET ROUTINE
*
        ORG     $E600
RESET   LOS     #BOS            SET S.P.
        LDX     #NMI            SET UP
        STX     NMIVEC          INTERRUPT
        STX     IRQVEC          VECTORS
        LDX     #SWI30          SET UP
        STX     USWI            SWI
        LDX     #SWIHAN         ADDRESSES
        STX     SWIVEC
        LDX     #ACIAA          SET UP POINTER
        STX     ACIAI           TO ACIA
        LDAA    #3              RESET
        STAA    ACIAS           THE
        LDAA    #1              ACIA
        STAA    ACIAC
        JSR     CONFIG          CONFIGURE PIAS
        JMP     NMI2            GO TO NMI ROUTINE
ACIAA   FDB     ACIAC
*
*       NMI ROUTINE
*       PROVIDES LOAD,GO,CONFIGURE AND PROTO FUNCTIONS
*
NMI     INS                     CLEAN INTERRUPT
        INS                     REGS FROM STACK
        INS
        INS
        INS
        INS
        INS
NMI2    JSR     CRLF            CR,LF
        LDAA    #'!             PROMPT
        SWI
        FCB     PRINTA          PDP-11
A05     LDX     #BUF            INITIALISE POINTER
A10     SWI                     WAIT
        FCB     INPUTA          FOR CHARACTER
        CMPA    #*134           \ = ESCAPE (IE RUBOUT)
        BNE     A12             NO
        JSR     CRLF            CR,LF
        BRA     NMI2            GO AGAIN
A12     STAA    0,X             STORE IT
        INX                     INCREMENT POINTER
        CPX     #BUF+50         INPUT LINE TOO LONG ?
        BNE     A15             NO
        JMP     ERROR           YES, ERROR
A15     CMPA    #$0D            IS IT <CR> ?
        BEQ     A18             YES - DECODE
        SWI                             ECHO INPUT
        FCB     PRINTA
        BRA     A10
A18     JSR     CRLF                    CR,LF
*
*       DECODE INSTRUCTION
*
        LDAA    BUF                     FETCH FIRST CHARACTER
        LDX     #CTABLE                 TABLE ADDRESS
A20     CMPA    0,X                     ENTRY = A ?
        BNE     A25                     NO
        LDX     1,X                     YES
        JMP     0,X                     JUMP TO ROUTINE
A25     INX                             INCREMENT POINTER
        INX
        INX
        CPX     #CEND                   END OF TABLE ?
        BNE     A20                     NO
        BRA     NMI2                    YES
*
*       TABLE OF COMMANDS
*
CTABLE  FCC     'C'                     CONFIGURE PIA'S
        FDB     CFSR
        FCC     'G'                     GO
        FDB     GO
        FCC     'J'                     JUMP TO PROTO
        FDB     PROTO
        FCC     'L'                     LOAD
        FDB     LOAD
        FCC     'Z'                     ZERO LOW RAM
        FDB     ZERO
        FCB     $D                      CR
        FDB     A05
CEND    EQU     *
*
*       CONFIGURE
*
CFSR    JSR     CONFIG
        BRA     NMI2
*
*       GO COMMAND
*
GO      LDX     #BUF+1
        LDAB    #6
        SWI
        FCB     CONHB                   CONVERT ADDR TO BINARY
        STAA    SAVEX
        STAB    SAVEX+1
        LDX     SAVEX                   TRANSFER A,B TO X
        JMP     0,X                     JUMP TO NEW ADDRESS
*
*       LOAD COMMAND
*
LOAD    LDAA    #$11                    PROMPT PDP-11
```

```
        SWI                         WITH DC1 ($11)
        FCB     PRINTA
L1Ø     SWI
        FCB     INPUTA              WAIT FOR CHARACTER
        CMPA    #'S                 IS IT AN "S"
        BNE     L1Ø                 NO ,GO AGAIN
L2Ø     SWI                         YES
        FCB     INPUTA              GET RECORD TYPE
        CMPA    #'Ø                 IS IT A HEADER?
        BEQ     L1Ø                 YES ,IGNORE RECORD
        STAA    RECTYP              SAVE RECORD TYPE
        CLR     CKSM                CLEAR CHECKSUM
        JSR     NEXT2D              GET BYTE COUNT
        DECA                        SUBTRACT THREE
        DECA                        BYTES FOR ADRESS
        DECA                        AND CHECKSUM
        STAA    COUNT               SAVE BYTE COUNT
        JSR     NEXT2D              GET MSB OF ADDR
        STAA    TADR                STORE IT
        JSR     NEXT2D              GET LSB OF ADDR
        STAA    TADR+1              STORE IT
        LDAA    RECTYP              GET RECORD TYPE
        CMPA    #'1                 IS IT A "1"?
        BNE     L3Ø                 NO ,CHECK FOR "9"
*       COMMENCE LOADING
L4Ø     JSR     NEXT2D              READ 2 HEX DIGITS AND DECODE
        LDX     TADR
        STAA    Ø,X                 PUT BYTE IN MEMORY
        CMPA    Ø,X                 CHECK BYTE OK ?
        BNE     ERROR               NO ,GO TO ERROR
        INX                         INCR POINTER
        STX     TADR                SAVE POINTER
        DEC     COUNT               DECR COUNT
        BGT     L4Ø                 >Ø ,GO AGAIN
        BRA     CHK                 CHECK CHECKSUM
L3Ø     CMPA    #'9                 EOF RECORD ?
        BNE     ERROR               NO ,ERROR
CHK     LDAA    CKSM                FETCH CHECKSUM
        PSHA                        SAVE ON STACK
        JSR     NEXT2D              GET CHKSM FROM TAPE
        PULB                        PULL CALC CHKSM INTO B
        COMB                        1'S COMPLEMENT B
        CBA                         COMPARE TWO CHKSM'S
        BNE     ERROR               NOT = ,ERROR
        LDAA    RECTYP              FETCH RECORD TYPE
        CMPA    #'9                 IS IT "9" ?
        BNE     L1Ø                 NO ,READ NEXT RECORD
        LDX     #MEOF               SEND END TO PDP-11
        SWI
        FCB     PMSG                WITH    A DC3 ($13)
        JMP     NMI2                GO TO START
*       ERROR HANDLING
ERROR   LDX     #MERR               TURN OFF PDP-11
        SWI                         AND SIGNAL ERROR
        FCB     PMSG                WITH BEL ($07)
        JMP     NMI2            GO TO START
*
*       S/R TO READ 2 DIGITS AND COMBINE THEM IN A
*
NEXT2D  SWI
        FCB     INPUTA          GET 1ST CHARACTER
        TAB                     SHIFT IT TO B
        SWI
        FCB     INPUTA          GET 2ND CHARACTER
        PSHA
        PSHB                    PUSH BOTH ON STACK
        TSX                     S.P. TO X REG
        LDAB    #2              MAX NO OF DIGITS=2
        SWI
        FCB     CONHB           CONVERT TO BINARY
        BCC     ERROR           GO ERROR IF C CLEAR
        TBA                     SWAP LSB TO A
        ADDB    CKSM            ADD BYTE TO CHECKSUM
        STAB    CKSM            SAVE CHECKSUM
        INS                     CLEAN UP
        INS                     THE STACK
        RTS
*
*       SUBROUTINE CONFIG
*       TO CONFIGURE THE PIAS FOR THE
*       COLUMN  CONTROL PERIPHERALS
*
CONFIG  BSR     CS36            SET C/S REGS TO $36
        CLRA                    SET ALL PERIPHERAL REGS TO Ø
        LDX     #V4
        LDAB    #6
CO2     STAA    Ø,X
        INX
        INX
        DECB
        BNE     CO2
        LDAA    #$3Ø            SELECT DDR FOR PIAS
        LDX     #V4CSR
        LDAB    #6
CO3     STAA    Ø,X
        INX
        INX
        DECB
        BNE     CO3
        CLRA                    SET PIA 2 FOR INPUT
        STAA    DMSB
        STAA    DLSB
        COMA                    SET PIAS 1,3 AS OUTPUTS
        LDX     #V4
        LDAB    #4
CO4     STAA    Ø,X
        INX
        INX
        DECB
        BNE     CO4
```

```
         BSR      CS36            SET ALL C/3 TO $36
         RTS
CS36     LDAA     #$36            SET ALL C/S REGS TO $36
         LDX      #V4CSR
         LDAB     #6
CO1      STAA     0,X
         INX
         INX
         DECB
         BNE      CO1
         RTS
*
*        CR,LF SUBROUTINE
*
CRLF     LDAA     #$D
         SWI
         FCB      PRINTA
         LDAA     #$A
         SWI
         FCB      PRINTA
         LDAB     #20             WAIT FOR 40MS FOR LF
CRLF2    LDAA     #$FF
CRLF3    DECA
         BNE      CRLF3
         DECB
         BNE      CRLF2
         RTS
*
*        ZERO LOW RAM
*
ZERO     LDX      #0              CLEAR POINTER
         CLRB                     CLEAR BYTE
Z1       STAB     0,X             STORE BYTE
         LDAA     0,X             READ BYTE
         INX                      INCR POINTER
         CBA                      COMPARE READ & WRITE BYTES
         BEQ      Z1              CONTINUE
         JMP      NMI2            FINISH
*
*        MESSAGES
*
MEOF     FCB      $13             END OF FILE
         FCC      /EOF/
         FCB      $0D,$0A,$04
MERR     FCB      $13,$7          ERROR
         FCC      /L ERR/
         FCB      $0D,$0A,$04
         END
```

IV.2 CC68

CC68 is a suite of programs used to control the microcomputer peripherals, and to provide single loop controllers as described in Chapter 5. A listing follows.

```
*
*   ****************************************
*   *  DEFINITIONS AND DATA BASE FOR CC68  *
*   ****************************************
*
*       DEFINITIONS
*       -----------
*
*       D.A.M. PIA
DCSR1   EQU     $FBC9
DCSR2   EQU     $FBCB
DMSB    EQU     $FBC8
DLSB    EQU     $FBCA
*       D/A PIAS
V1      EQU     $FBC6           D/A CH1 R.D. LEVEL
V2      EQU     $FBC4           D/A CH2 REB LEVEL
V3      EQU     $FBC2           D/A CH3 TOPS PRODUCT
V4      EQU     $FBCØ           D/A CH4 STEAM FLOW
V1CSR   EQU     V1+1
V2CSR   EQU     V2+1
V3CSR   EQU     V3+1
V4CSR   EQU     V4+1
*       ACIA REGISTERS
ACIAS   EQU     $FBCE
ACIAD   EQU     $FBCF
*       RSRSR INDICES
SUBXAB  EQU     $ØB
P2HEX   EQU     $ØF
P4HEX   EQU     $1Ø
PRINTA  EQU     $11
PMSG    EQU     $12
VALAN   EQU     $13
INPUTA  EQU     $14
CONHB   EQU     $15
*       INTERRUPT VECTORS
IRQVEC  EQU     $FFF8
*
*       PROTO RAM DEFINITIONS
*
BUF     EQU     $FF9Ø           LINE OF INPUT
ADR     EQU     $FFDA
BUFPTR  EQU     $FFEØ
IOFLAG  EQU     $FFEA
*
*
*
*       DATA BASE
*       ---------
        ORG     $Ø
SYSTAT  FCB     Ø               SYSTEM STATUS
DAMERR  FCB     Ø               D.A.M. ERRORS
RSTAT   FCB     Ø               REFRACTOMETER STATUS
TSAMPL  FCB     1               SAMPLE INTERVAL
DAM     RMB     32              D.A.M. DATA AREA
REFRAC  RMB     8               REFRAC DATA AREA
```

```
*
*       SETPOINTS
*
        ORG     $30
CLVL    FDB     $800            CONDENSER LEVEL SETPOINT
RLVL    FDB     $6D3            BOILER LEVEL SETPOINT
DIST    FDB     $800            DISTILLATE SETPOINT
BOTT    FDB     $800            BOTTOMS SETPOINT
*
*       VALVE PARAMETERS
*
        ORG     $38
VOLD1   FCB     0               OLD VALVE POSITION
VOLD2   FCB     0
VOLD3   FCB     0
VOLD4   FCB     0
VNEW1   FCB     0               NEW VALVE POSITION
VNEW2   FCB     0
VNEW3   FCB     0
VNEW4   FCB     0
VMAX1   FCB     5               MAX VALVE TRAVEL RATE
VMAX2   FCB     5
VMAX3   FCB     5
VMAX4   FCB     5
*
*       MULTIPLICATION BUFFERS
*
MULT1   RMB     2
MULT2   RMB     2
PROD    RMB     4
FF      RMB     1
*
*       LEVEL LIMITS
*
        ORG     $50
RDU     FDB     $FFF            R.D.
RDL     FDB     $0
RBU     FDB     $FFF            REBOILER
RBL     FDB     $0
*
*       WATCH DOG TIMERS
*
TIM1    FCB     0               TIMER FUNCTIONS
TIM2    FCB     0               REFFRACTOMETER TIMER
TIM3    FCB     0
TIM4    FCB     0
TIM5    FCB     0
*
*       WORKING AREA
*
SIGN    RMB     1
XREG    RMB     2
TEMP1   RMB     2               WORKING STORAGE
TEMP2   RMB     2
CPTR    RMB     2               CONTROLLER POINTER
CNO     RMB     1               CONTROLLER NUMBER
```

```
*
*       COLUMN PRESSURE DROP LIMITS
*
PU      FDB     $FFF            UPPER LIMIT
PL      FDB     $Ø              LOWER LIMIT
        RMB     7
*
*       REFRACTOMETER PARAMTERS
*
        ORG     $8Ø
RESEQ   FCB     Ø,1,2,3,Ø,1,2,3,Ø,1,2,3
SEQPTR  FCB     Ø
TRMIN   FCB     6               REF SAMPLE =TRMIN*BASIC TIME
*
*       PI CONTROLLER PARAMETERS
*
        ORG     $9Ø
C1A     FDB     -128            FIRST COEFF
C1B     FDB     112             SECOND COEFF
CPVPTR  FCB     Ø               PV PTR IN DAM DATA
C1PVØ   FDB     Ø               CURRENT SP-PV
C1PV1   FDB     Ø               LAST SP-PV
LVP1    FDB     Ø               LAST VALVE POSN
C2      FDB     -128,112        SECOND CONTROLLER
        FCB     1
        FDB     Ø,Ø,Ø
C3      FDB     Ø,Ø,Ø,Ø,Ø       THIRD CONTROLLER
        FCB     Ø
C4      FDB     Ø,Ø,Ø,Ø,Ø       FOURTH CONTROLLER
        FCB     Ø
DIVN    FDB     Ø               CONTROLLER DIVISION FACTOR
```

```
*
*                   ******************
*                   *                *
*                   *  MAIN PROGRAM  *
*                   *                *
*                   ******************
*
*
*       SET UP PIA'S
CONFIG  EQU     $E728           ROUTINE IN OPSYS EPROM
        ORG     $EØØØ
MAIN1   JSR     CONFIG
*
*       RESTART ENTRY POINT
*
MAIN2   LDAA    #$37            ENABLE INTS THRU PIA 3 CA1
        STAA    V4CSR
        LDAA    #$3C
        STAA    DCSR2
        LDX     #REFRAC         CLEAR REFRAC DATA AREA
MA1     CLR     Ø,X             CLEAR REF DATA AREA
        INX
        CPX     #REFRAC+9
        BNE     MA1
        LDX     #TIM1           CLEAR TIMERS
MA2     CLR     Ø,X
        INX
        CPX     #TIM5+1
        BNE     MA2
        CLR     SYSTAT          CLEAR SYSTEM STATUS
        LDX     #TIMER          SET UP TIMER INT'S
        STX     IRQVEC
        JSR     USERP           PROMPT USER FOR I/O
        CLI
*
*       MAIN OPERATING LOOP STARTS HERE
*
MA3     TST     IOFLAG          LINE OF INPUT READY ?
        BEQ     MA4
        JSR     USERC           USER COMMAND
MA4     LDAA    TIM1            TIME TO CONTROL ?
        CMPA    TSAMPL
        BCS     MA3             NO
*       TIME TO SAMPLE AND CONTROL
        INC     SYSTAT          SYSTEM STATUS TO 1
        CLR     TIM1            CLEAR COUNTER
        JSR     READAD          GO TO DATA AQUISITION
        INC     SYSTAT          SYSTEM STATUS TO 2
        JSR     CHECK           CHECK ALARMS
        INC     SYSTAT          SYSTEM STATUS TO 3
        JSR     REFCTL          GO TO REFRAC CONTROLLER
        INC     SYSTAT          SYSTEM STATUS TO 4
        JSR     CONTRL          GO TO CONTROL ROUTINE
        INC     SYSTAT          SYSTEM STATUS TO 5
        JSR     VALVES          IMPLEMENT CONTROLS
        INC     SYSTAT          SYSTEM STATUS TO 6
        JSR     $200
*       ALLOWS EXPANSION OF THE BASIC
*       PROGRAM IN RAM BY DOWN LOADING FROM THE
*       PDP-11, A PROGRAM TO EXECUTE FROM $200.
*       THIS PROGRAM MUST LOOK LIKE A SUBROUTINE
*       AND USE THE STANDARD DATA BASE.
        CLR     SYSTAT          SYSTEM STATUS BACK TO 0
        BRA     MA3
```

```
*       ************************
*       *  INTERRUPT HANDLER  **
*       ************************
*
*       TIMER INTERRUPT ROUTINE
*       OPERATING ON IRQ LINE
*       VIA CA1 PORT ON PIA 3
*       AND TO ACCEPT USER COMMANDS
*       FROM A TERMINAL UNDER
*       INTERRUPT.
*
IRQINT  EQU     $FBC1
NMINT   EQU     $E637
*
TIMER   LDAB    IRQINT          GET IRQ FLAG FROM PIA C/S
        LDAA    IRQINT-1        CLEAR INT FLAG,STOP FURTHER INTS
        TSTB
        BMI     T1              WAS IT IRQ ?
        LDAA    ACIAS           READ ACIA STATUS
        BMI     T4              CHARACTER AVAILABLE
        JMP     NMINT           NO - NMI INTERRUPT
*
*       WATCH DOG TIMER ROUTINES
*
T1      LDX     #TIM1           POINTER TO COUNTER
T2      INC     0,X             INCREMENT COUNTER
        INX                     INCREMENT POINTER
        CPX     #TIM5+1         LAST COUNTER ?
        BNE     T2              NO
        RTI                     YES
*
*       INPUT CHARACTERS
*
T4      LDAA    ACIAD           READ CHARACTER
        ANDA    #$7F            STRIP PARITY
        CMPA    #@134           \ = ESCAPE ?
        BNE     T6              NO
T5      JSR     CRLF
        JSR     USERP
        RTI                     ABORT
T6      LDX     BUFPTR
        STAA    0,X             SAVE NEW CHARACTER
        INX                     INC POINTER
        STX     BUFPTR          SAVE POINTER
        CPX     #BUF+50         LINE TOO LONG ?
        BEQ     T5              YES - ABORT
        CMPA    #$D             CR ?
        BEQ     T8              YES
T7      LDAB    ACIAS           READ ACIA STATUS
        BITB    #2              READY TO SEND ?
        BEQ     T7              NO
        STAA    ACIAD           YES
        RTI
T8      INC     IOFLAG          SIGNAL END-OF-MESSAGE
        RTI
```

```
*       ************************
*       *  OPERATOR INTERFACE  *
*       ************************
*
*
USERC   JSR     CRLF
        LDAA    BUF             DECODE INPUT
        LDX     #CTABLE
I02Ø    CMPA    Ø,X             COMPARE COMM WITH TABLE
        BNE     I025            NO
        LDX     1,X             YES - GET ROUTINE ADDRESS
        JMP     Ø,X             GO THERE
I025    INX
        INX
        INX                     INCREMENT POINTER
        CPX     #CEND           END OF TABLE ?
        BNE     I02Ø            NO - CONTINUE
*
*       ERROR HANDLING
*
ERROR   EQU     *
USERP   LDAA    #'?             PROMPT USER
        SWI
        FCB     PRINTA
I03Ø    LDX     #BUF
        STX     BUFPTR          SET UP POINTER
        CLR     IOFLAG          CLEAR FLAG
        LDAA    #$81            ENABLE INTS ON ACIA
        STAA    ACIAS
        RTS
*
*       COMMAND TABLE
*
CTABLE  FCC     'C'
        FDB     PCON
        FCC     'D'
        FDB     PDAM
        FCC     'I'
        FDB     SETMEM
        FCC     'O'
        FDB     OUTMEM
        FCC     'P'
        FDB     PSYS
        FCC     'R'
        FDB     PREF
        FCC     'S'
        FDB     PSET
        FCC     'V'
        FDB     PVAL
        FCB     $15
        FDB     I03Ø
CEND    EQU     *
```

```
*
*       PRINT DATA IN RESPONSE TO USER REQUESTS
*       ENTRY POINTS ARE:PVAL,PSYS,PSET,PREF,PDAM,PCON
*
PVAL    LDX     #VOLD1          PRINT VALVES
        BRA     P1
PSYS    LDX     #SYSTAT         PRINT STATUS
P1      LDAB    #4              BYTE COUNT
P2      BSR     PSNGL           PRINT BYTE
        DECB
        BNE     P2              FINISHED ?
        BRA     PEXIT           YES
PSET    LDX     #CLVL           PRINT SETPOINTS
        BRA     P3
PREF    LDX     #REFRAC         PRINT COMPOSITIONS
P3      LDAB    #4              DOUBLE BYTE COUNT
P4      BSR     PDBLE           PRINT DOUBLE BYTE
        DECB
        BNE     P4              FINISHED ?
PEXIT   BSR     CRLF            YES
        JMP     USERP
PDAM    LDX     #DAM            PRINT ACQUIRED DATA
        LDAB    #-1             COUNTER
P5      LDAA    #8              COUNTER
P6      BSR     PDBLE           PRINT DOUBLE BYTE
        DECA
        BNE     P6              FINISHED LINE ?
        BSR     CRLF            YES
        INCB                    FINISHED ?
        BEQ     P5              NO
        JMP     USERP           YES
PCON    LDX     #C1A            PRINT CONTROLLERS
        LDAB    #4              COUNTER
P7      BSR     PDBLE           PRINT K1
        BSR     PDBLE           PRINT K2
        BSR     PSNGL           PRINT PV INDEX
        BSR     CRLF
        LDAA    #6              INC POINTER
P8      INX
        DECA
        BNE     P8
        DECB
        BNE     P7              FINISHED ?
        BSR     PSNGL           PRINT DIVN
        BSR     CRLF
        JMP     USERP
*
*       PRINTING ROUTINES
*
PSNGL   SWI                     PRINT SINGLE BYTE
        FCB     P2HEX
        BRA     BLK
PDBLE   SWI                     PRINT DOUBLE BYTE
        FCB     P4HEX
BLK     PSHA                    SAVE A
```

```
        LDAA    #$2Ø
        SWI
        FCB     PRINTA
        PULA                    RESTORE A
        RTS
CRLF    LDAA    #$D             CR
        SWI
        FCB     PRINTA
        LDAA    #$A             LF
        SWI
        FCB     PRINTA
        PSHB                    SAVE B
        LDAB    #2Ø             WAIT 4ØMS FOR TERMINAL TO LF
CRLF2   LDAA    #$FF
CRLF3   DECA
        BNE     CRLF3
        DECB
        BNE     CRLF2
        PULB                    RESTORE B
        RTS
*
*       ROUTINE TO OUTPUT MEMORY LOCATIONS
*
*       FORMAT: O <ADDRESS>(,<COUNT>)
*       COUNT IN THE RANGE 1 TO 15 ONLY
*       OUTPUT: <ADDR> <BYTE> <BYTE> ...
*
OUTMEM  LDX     #BUF+2          POINTER TO INPUT STRING
        LDAB    #4              MAX NO OF DIGITS
        SWI
        FCB     CONHB           CONVERT ADDRESS TO BINARY
        STAA    ADR             SAVE ADDRESS
        STAB    ADR+1
        LDAB    #1              SET UP COUNTER
        LDAA    Ø,X             NEXT CHARACTER IN STRING
        CMPA    #$D             IS IT CR ?
        BEQ     OU1Ø            YES
        INX                     NO - GET COUNT FROM INPUT
        SWI
        FCB     CONHB           CONVERT COUNT
OU1Ø    LDX     #ADR            GET DUMP ADDRESS
        BSR     PDBLE           PRINT ADDRESS
        LDX     ADR             POINT TO OUTPUT BYTES
OU2Ø    BSR     PSNGL           PRINT 1 BYTE
        DECB                    DEC COUNT
        BNE     OU2Ø            NOT FINISHED
        BSR     CRLF            CR,LF
        JMP     USERP           RETURN
*
*       ROUTINE TO SET MEMORY
*
*       FORMAT: I <ADDR>,<BYTE>(<DEL>)<BYTE>...
*
SETMEM  LDX     #BUF+2          POINT TO INPUT STRING
        LDAB    #4              MAX NO OF DIGITS
        SWI
        FCB     CONHB           CONVERT ASCII TO BINARY
        STAA    ADR             SAVE ADDRESS
        STAB    ADR+1
        INX                     INCREMENT TO FIRST BYTE
SET1Ø   LDAB    #2              2 DIGITS PER BYTE
        SWI
        FCB     CONHB           CONVERT BYTE
        BCC     SET2Ø           FINISHED ?
        STX     BUFPTR          SAVE POINTER
        LDX     ADR             FETCH ADDRESS
        STAB    Ø,X             SAVE NEW BYTE
        INX                     INCR POINTER
        STX     ADR             SAVE ADDRESS
        LDX     BUFPTR          GET POINTER TO STRING
        LDAA    Ø,X             GET NEXT CHARACTER
        CMPA    #$D             IS IT CR ?
        BEQ     SET2Ø           YES - FINISH
        SWI
        FCB     VALAN           IS IT HEXADECIMAL ?
        BVC     SET1Ø           NO - NEXT BYTE
        INX                     YES - DEC POINTER,NEXT BYTE
        BRA     SET1Ø
SET2Ø   JMP     USERP
        END
        ORG     $E2ØØ
*
*      ********************************
*      *  DATA ACQUISITION SUBROUTINE  *
*      ********************************
*
*       CONTROLS THE OPERATION OF
*       THE DATA ACQUISITON MODULE
*       AND MONITORS ITS OPERATION
*
READAD  CLR     DAMERR          CLEAR ERROR COUNTER
        CLRB                    FIRST CHANNEL POINTER
        LDX     #DAM            SET UP DATA AREA POINTER
        LDAA    DLSB            READ PERIPH REG TO CLEAR ANY EOC
        CLR     TIM3            USE TIM3 AS WATCHDOG
        LDAA    #$36            PULSE THE START LINE
        STAA    DCSR2
        NOP
        NOP
        LDAA    #$3E
        STAA    DCSR2
DAQ1    LDAA    DCSR2           WAIT FOR EOC
        BMI     DAQ1A
        LDAA    TIM3            CHECK WATCHDOG
        CMPA    #$1             OK ?
        BCS     DAQ1            YES
        LDAA    #$FF            ERROR - MISSED EOC'S
        STAA    DAMERR          INDICATE ERROR
        BRA     DAQ5
DAQ1A   LDAA    DMSB            READ MSB
        PSHA                    SAVE MSB FOR LATER
        LSRA
        LSRA
        LSRA
        LSRA
        CBA                     IS IT CORRECT ?
        BEQ     DAQ2            YES
        INC     DAMERR          NO
        PULA                    RESTORE STACK
        BRA     DAQ3
DAQ2    PULA                    RETRIEVE MSB FROM STACK
        STAA    Ø,X             STORE IT
        LDAA    DLSB            GET LSB
        STAA    1,X             STORE IT
DAQ3    INCB                    INCREMENT CHANNEL POINTER
        INX                     INCREMENT DATA POINTER
        INX
        CMPB    #$1Ø            LAST CHANNEL ?
        BCS     DAQ1            NO
DAQ5    RTS
```

```
*
*    ****************************
*    *  ALARM CHECKING ROUTINE  *
*    ****************************
*
CHECK   LDAA    DAMERR          D.A.M OK ?
        BEQ     CH1             YES
        LDAA    #'Ø             ALARM NUMBER
        BSR     ERRL
*       CHECK ON REFLUX DRUM
CH1     LDAA    DAM             GET MSB
        ANDA    #$ØF            STRIP CH NO
        LDAB    DAM+1           GET LSB
        LDX     RDU             UPPER LIMIT
        BSR     CMP16           COMPARE
        BLS     CH2             INSIDE
        LDAA    #'1             ERROR 1
        BSR     ERRL
        BRA     CH3
CH2     LDX     RDL             LOWER LIMIT
        BSR     CMP16           COMPARE
        BCC     CH3             INSIDE
        LDAA    #'1             ERROR 1
        BSR     ERRL
*       CHECK ON REBOILER LEVEL
CH3     LDAA    DAM+2           GET MSB
        ANDA    #$ØF            STRIP CH NO
        LDAB    DAM+3           GET LSB
        LDX     RBU             UPPER LIMIT
        BSR     CMP16           COMPARE
        BLS     CH4             INSIDE
        LDAA    #'2             ERROR 2
        BSR     ERRL
        BRA     CH5
CH4     LDX     RBL             LOWER LIMIT
        BSR     CMP16           COMPARE
        BCC     CH5             INSIDE
        LDAA    #'2             ERROR 2
        BSR     ERRL
*       CHECK COLUMN PRESSURE DROP
CH5     LDAA    DAM+26          GET MSB
        ANDA    #$ØF            STRIP CH NO
        LDAB    DAM+27          LSB
        LDX     PU              UPPER LIMIT
        BSR     CMP16           COMPARE
        BLS     CH6             INSIDE
        LDAA    #'3             ERROR 3
        BSR     ERRL
        BRA     CH7
CH6     LDX     PL              LOWER LIMIT
        BSR     CMP16           COMPARE
        BCC     CH7             INSIDE
        LDAA    #'3             ERROR 3
        BSR     ERRL
CH7     RTS
```

```
*
*       ALARM CONDITION
*       0 = DAM ERROR
*       1 = REFLUX ACCUMULATOR LEVEL
*       2 = REBOILER LEVEL
*       3 = COLUMN PRESSURE DROP
*
ERRL    LDX     #CHKMSG         PRINT ALARM MESSAGE
        SWI
        FCB     PMSG
        SWI
        FCB     PRINTA          ALARM CODE
        SWI
        FCB     PMSG
        RTS
CHKMSG  FCB     $D,$A,$07
        FCC     / ? ALARM /
        FCB     $4
        FCC     / ?/
        FCB     $D,$A,$7,$4
*
*       CMP16: PERFORMS A 16 BIT COMPARISION
*       BETWEEN A,B AND X REGISTERS. THE CC ARE
*       SET ACCORDINGLY. A,B,X ARE UNALTERED.
*       A=MSB,B=LSB.
*
CMP16   PSHA                    SAVE REGISTERS
        PSHB
        SWI
        FCB     SUBXAB          CALCULATE A,B-X
        PULB
        PULA                    RESTORE REGISTERS
        RTS
```

```
*
*     ************************************
*     *  REFRACTOMETER CONTROL ROUTINE  *
*     ************************************
*
REFCTL  LDAA    TIM2            GET REF TIMER
        CMPA    TRMIN           COMP WITH TRMIN
        BCS     RE1Ø            NOT LONG ENOUGH
        LDX     #REFRAC         CLEAR ALL NEW DATA BITS
        LDAB    #4              (BIT 15) ON ALL CHANNELS
REØ     LDAA    Ø,X             FETCH MSB
        ANDA    #$ØF            STRIP NDB
        STAA    Ø,X             REPLACE BYTE
        INX
        INX                     NEXT CHANNEL
        DECB                    FINISHED ?
        BNE     REØ             NO
        LDX     #REFRAC         ADDRESS O REFRAC DATA AREA
        LDAB    RSTAT           GET REF CHANNEL
        BEQ     RE2             CHANNEL Ø
        ASLB                    DOUBLE CHANNEL NO FOR OFFSET
RE1     INX                     INCREMENT POINTER
        DECB                    FOR CHANNELS
        BNE     RE1             OTHER THAN Ø
RE2     LDAA    DAM+28          GET MSB FROM D.A.M.
        ANDA    #$ØF            CLEAR D.A.M. CH NO.
        ORAA    #$8Ø            SET NEW DATA BIT (15)
        STAA    Ø,X             STORE REF DATA MSB
        LDAA    DAM+29          GET LSB FROM D.A.M.
        STAA    1,X             AND STORE IT
        LDX     #RESEQ          GO TO SEQ TABLE
        LDAB    SEQPTR          GET TABLE POINTER
        BEQ     RE4
RE3     INX
        DECB                    UPDATE X TO SEQPTR
        BNE     RE3
RE4     LDAA    Ø,X
        STAA    RSTAT           STORE NEXT CH
        LDAB    SEQPTR          INCR SEQ PTR
        INCB
        CMPB    #$ØC            WRAP AROUND ?
        BCS     RE5
        CLRB
RE5     STAB    SEQPTR          STORE NEW SEQ PTR
        ASRA                    SET BIT Ø FOR REF
        BCC     RE6
        LDAB    #$3E
        BRA     RE7
RE6     LDAB    #$36
RE7     STAB    V1CSR
        ASRA                    SET BIT 1 FOR REF
        BCC     RE8
        LDAB    #$3E
        BRA     RE9
RE8     LDAB    #$36
RE9     STAB    V2CSR
        CLR     TIM2            CLEAR REF TIMER
RE1Ø    RTS
        END
        ORG     $E4ØØ
*
*
*     *********************
*     *  CONTROL ROUTINE  *
*     *********************
*
*       ALGORITHM IS
*       DV=(C?A*(SP-PVØ)+C?B*(SP-PV1))/2**DIVN
*       VALVE=LAST VALVE+DV
*       VNEW=VALVE/256+128 (8 BITS)
*
CONTRL  LDX     #C1A            INITL POINTER
        CLR     CNO             INITL COUNTER
CTØ     STX     CPTR            SAVE POINTER
        LDAA    5,X             SHIFT C?PVØ TO C?PV1
        LDAB    6,X
        STAA    7,X
        STAB    8,X
        LDAA    4,X             ADDR OF PV IN DAM BLOCK
        LDX     #DAM            GET PV ADDR (ADJUST IF NEC.)
        ASLA                    DOUBLE A
        JSR     ADJX            ADJUST X POINTER
        LDAB    1,X             GET PV FROM DAM BLOCK
        LDAA    Ø,X
        ANDA    #$ØF            CLEAR CH NO.
        LDX     CPTR            GET CONTROLLER POINTER
        STAA    5,X
        STAB    6,X             STORE AT C?PVØ
        LDX     #CLVL           GET SP ADDR
        LDAA    CNO
        ASLA                    DOUBLE OFFSET
        JSR     ADJX            ADJUST X POINTER
        LDAA    Ø,X             GET SP
        LDAB    1,X
        LDX     CPTR            GET CONTROLLER POINTER
        SUBB    6,X
        SBCA    5,X             SP-PV
        STAB    6,X
        STAA    5,X             STORE IN C?PVØ
*
*       NOW THE CONTROLLER CALCULATIONS
*
        LDX     Ø,X             C?A
        STX     MULT1
        LDX     CPTR
        LDX     5,X             C?PVØ
        STX     MULT2
        JSR     MULT16          MULTIPLY
        LDX     PROD+2
        STX     TEMP1           C?A*C?PVØ
        LDX     CPTR
        LDX     2,X             C?B
        STX     MULT1
        LDX     CPTR
        LDX     7,X             C?PV1
        STX     MULT2
        JSR     MULT16          MULTIPLY
        LDAA    PROD+2
        LDAB    PROD+3
        ADDB    TEMP1+1         C?A*C?PVØ + C?B*C?PV1
        ADCA    TEMP1
        BSR     OVERFL          CHECK FOR OVERFLOW
        LDX     DIVN            DIVIDE OUTPUT BY 2**DIVN
        BEQ     CT4B
CT4A    ASRA
        RORB
        DEX
        BNE     CT4A
CT4B    LDX     CPTR
        ADDB    1Ø,X            ADD ON LAST VALVE POSITION
        ADCA    9,X
        BSR     OVERFL          CHECK FOR OVERFLOW
        STAB    1Ø,X
        STAA    9,X             SAVE VALVE POSITION
        LDX     #8              NOW /256  = 8 RIGHT SHIFTS
CT5     ASRA
        RORB
        DEX
        BNE     CT5
        ADDB    #$8Ø            ADD ON 128
        ADCA    #$Ø
*       NORMALISE TO Ø-255 RANGE
        BMI     CT6             < Ø
        BEQ     CT7             Ø<= VALUE <= 255
        LDAB    #$FF            SET TO MAX
        BRA     CT7
CT6     CLRB                    SET TO MIN
CT7     LDX     #VNEW1
        LDAA    CNO
        BSR     ADJX            ADJUST X POINTER
        STAB    Ø,X             SAVE VNEW
*
*       ADJUST POINTER FOR NEXT CONTROLLER
*
        LDX     CPTR
        LDAB    #$8
CT1Ø    INX
        DECB
        BNE     CT1Ø
        INC     CNO             INCREMENT COUNTER
        LDAB    CNO
        CMPB    #$3             LAST CONTROLLER?
        BHI     CT11
        JMP     CTØ             NO - GO AGAIN
CT11    RTS
*
*       ROUTINE TO INCREMENT THE X REGISTER
*       BY THE CONTENTS OF A (UNSIGNED)
*
```

```
ADJX    BEQ     AD2
AD1     INX
        DECA
        BNE     AD1
AD2     RTS
*
*       OVERFLOW SUBROUTINE
*       DETECTS +- OVERFLOWS FOR 16 BIT INTEGER
*       ARITHMETIC ON THE ACCUMULATORS WITH
*       A AS THE MSB.
*
OVERFL  BVS     OV2             OVERFLOW ?
        RTS                     NO
OV2     BCS     OV1             NEGATIVE OVERFLOW ?
        LDAB    #$FF            NO - SET TO MAX
        LDAA    #$7F
        RTS
OV1     CLRB            YES - SET TO MIN
        LDAA    #$8Ø
        RTS


*    **************************************************
*    *  16 BIT SIGNED INTEGER MULTIPLICATION ROUTINE  *
*    **************************************************
*
*       THIS ROUTINE MULTIPLIES TWO 16 BIT
*       2'S COMPLEMENT NUMBERS USING BOOTH'S ALGORITHM.
*       IT PRODUCES A 16 BIT RESULT WITH THE V BIT SET
*       IF OVERFLOW OCURRED.
*       THE MULTIPLIER = MULT1 =MULT1,MULT1+1
*       THE MULTIPLICAND = MULT2 =MULT2,MULT2+1
*       THE PRODUCT = PROD =PROD+2,PROD+3
*       THE TEST BYTE FOR MULT1(LSB-1) = FF
*
*       THE MULTIPLICAND WILL BE UNCHANGED,THE MULTIPLIER
*       WILL BE DESTROYED. A 16 BIT RESULT IN PROD+2,PROD+3
*       RESULTS,BUT PROD,PROD+1 NOW CONTAIN RUBBISH.
*
MULT16  LOX     #5              CLEAR THE WORKING REGS.
        CLRA
LP1     STAA    PROD-1,X
        DEX
        BNE     LP1
        LDX     #16             INIT'L SHIFT COUNTER TC 16.
LP2     LDAA    MULT1+1         GET Y(LSBIT).
        ANDA    #1
        TAB                     SAVE Y(LSBIT) IN B.
        EORA    FF              DOES Y(LSBIT)=Y(LSB-1)?
        BEQ     SHIFT           YES: GO TO SHIFT ROUTINE.
        TSTB                    NO: DOES Y(LSBIT)=Ø?
        BEQ     ADD             YES: GO TO ADD ROUTINE.
        LDAA    PROD+1          NO: SUBTRACT MULTIPLICAND
        LDAB    PROD            PRODUCT WITH THE MSBYTES
        SUBA    MULT2+1        LINED UP,
        SBCB    MULT2
        STAA    PROD+1
        STAB    PROD
        BRA     SHIFT          THEN GO TO SHIFT ROUTINE.
ADD     LDAA    PROD+1         ADD THE MULTPLICAND TO THE
        LDAB    PROD           PRODUCT WITH THE MSBYTES
        ADDA    MULT2+1        LINED UP.
        ADCB    MULT2
        STAA    PROD+1
        STAB    PROD
SHIFT   CLR     FF             CLEAR THE TEST BYTE.
        ROR     MULT1          SHIFT THE MULTIPLIER RIGHT
        ROR     MULT1+1        ONE BIT WITH THE LSBIT
        ROL     FF             INTO THE LSBIT OF FF.
        ASR     PROD           SHIFT THE PRODUCT RIGHT ONE
        ROR     PROD+1         BIT,THE  MSB REMAINING THE SAME.
        ROR     PROD+2
        ROR     PROD+3
        DEX                    DECREMENT THE SHIFT COUNT.
        BNE     LP2            IF <> Ø CONTINUE.
        LDX     PROD           CHECK FOR OVERFLOW
        LDAA    PROD
        BPL     POS
        CPX     #$FFFF
        BNE     NEGOFL
        LDAA    PROD+2
        BMI     RETURN
NEGOFL  LDX     #$8ØØØ
        STX     PROD+2
        SEV
        BRA     RETURN
POS     CPX     #$Ø
        BNE     POSOFL
        LDAA    PROD+2
        BPL     RETURN
POSOFL  LDX     #$7FFF
        STX     PROD+2
        SEV
RETURN  RTS                    RETURN
```

```
*     ****************************
*     *  CONTROL OUTPUT ROUTINE  *
*     ****************************
*
*        INCORPORATING CONTINUOUS SLEWING
*        NOTE: THE CONTROLLERS MUST ENSURE THAT
*              THE CONTROLLER OUTPUTS ARE IN THE
*              RANGE Ø-255(1Ø).
*
*                THE ROUTINE MAY BE CALLED AS S/R
*        BY THE MAINLINE PROGRAM PROVIDED THE
*        IRQ INTERRUPT MASK IS OFF. IF THE MASK IS
*        ON THEN THE ROUTINE WILL EXIT BACK TO
*        THE MONITOR.   THIS ALLOWS THIS
*        ROUTINE TO BE CALLED WHILE THE MAINLINE
*        IS STALLED AND MICRO-PDP DIALOG IS UNDERWAY.
*        NOTE: THE ADDRESS OF NMI2 IS $E637
NMI2     EQU      $E637
*
VALVES   LDX      #V4                ENSURE VOLD=OUTPUT(CURRENT)
         LDAA     6,X
         STAA     VOLD1
         LDAA     4,X
         STAA     VOLD2
         LDAA     2,X
         STAA     VOLD3
         LDAA     Ø,X
         STAA     VOLD4
01       CLRB                        INDICATOR FOR >1 STEPS
         LDX      #VOLD1             POINTER TO VALVE DATA
02       CLR      SIGN               CLEAR SIGN
         LDAA     4,X                VNEW
         SUBA     Ø,X                VNEW-VOLD
         BEQ      07                 NO CONTROL ACTION
         BCC      03                 VNEW>VOLD
         NEGA                        NEGATE VNEW-VOLD
         INC      SIGN               AND NOTE IT
03       CMPA     8,X                COMPARE CHANGE WITH MAX
         BLS      05                 VNEW-VOLD<=VRATE
         INCB                        INCREMENT TO INDICATE MAX STEP
         LDAA     Ø,X
         TST      SIGN               CHECK SIGN
         BNE      04
         ADDA     8,X                ADD MAX STEP
         BCC      06                 CHECK FOR OVERFLOW
         LDAA     #$FF               SET TO MAX
         BRA      06
04       SUBA     8,X                SUBTRACT MAX STEP
         BCC      06                 CHECK FOR NEG OVFLW
         LDAA     #$ØØ               SET TO MIN
         BRA      06
05       LDAA     4,X                USE CALC STEP
06       STAA     Ø,X                SAVE NEW VOLD
07       INX                         INCREMENT POINTER
         CPX      #VOLD1+4           LAST TIME AROUND
```

```
        BNE     02              NO
        LDX     #V4             OUTPUT SETTINGS TO VALVES
        LDAA    VOLD1
        STAA    6,X
        LDAA    VOLD2
        STAA    4,X
        LDAA    VOLD3
        STAA    2,X
        LDAA    VOLD4
        STAA    Ø,X
        TSTB                    CHECK FOR MAX STEP ?
        BEQ     09              NO
        LDX     #$4ØØØ          PAUSE 1/8 SEC
08      DEX
        BNE     08
        BRA     01              AND GO AROUND AGAIN
*       CHECK HERE FOR S/R OR EXTERNAL CALL
09      TPA                     GET C.C.
        ANDA    #$1Ø            IS IRQ MASK ON ?
        BEQ     01Ø             NO - RTS
        INS
        INS                     CLEAN UP STACK
        JMP     NMI2            JUMP TO MONITOR
01Ø     RTS
        END
```

IV.3 FEEDFORWARD EXTENSIONS TO CC68

The software described in the CC68 suite of routines was modified to implement the feedforward/feedback controller described in Chapter 8. This involved additions to the database, modification of the main program, and the addition of routines to implement the controller and adapter. A listing of these changes and additions follows.

```
*
*    ***************************************
*    *   FEEDFORWARD CONTROLLER DATA BASE  *
*    ***************************************
*
*
*       STATUS FLAGS (Ø=OFF , <>Ø=ON)
*
        ORG     $FD
ADSTAT  FCB     $Ø                  ADAPT STATUS , ON=ADAPT
SESTAT  FCB     $Ø                  TRIMMING CONTROLLER STATUS
FFSTAT  FCB     $Ø                  FEEDFORWARD CONTROLLER STATUS
*
*       DATA BASE EXTENSION FOR THE FEEDFORWARD CONTROLLER
*
        ORG     $1ØØ
XD      FDB     $ØØE6,$6666         (Ø.9)
XB      FDB     $7CCC,$CCCD         (Ø.Ø5)
XF      FDB     $7FCC,$CCCD         (Ø.4)
TF      FDB     $Ø5AØ,$ØØØØ         (2Ø.)
F       FDB     $Ø1EØ,$Ø            (1.75=#1Ø)
Q       FDB     Ø,Ø
RM      FDB     Ø,Ø
NM      FDB     Ø,Ø
FM      FDB     Ø,Ø
LM      FDB     Ø,Ø
LV      FDB     Ø,Ø
VM      FDB     Ø,Ø
QS      FDB     Ø,Ø
R       FDB     Ø,Ø
Y9      FDB     Ø,Ø
DALR    FDB     Ø
DAQS    FDB     Ø
PVN1    FDB     Ø,Ø
PVN4    FDB     Ø,Ø
*
*       FF COMPENSATOR PARAMETERS
*
ZNA     FDB     $Ø,$Ø               COMPENSATED LV
YNP1A   FDB     $Ø,$Ø
ZNB     FDB     $Ø,$Ø               COMPENSATED QS
YNP1B   FDB     $Ø,$Ø
KA1     FDB     $Ø,$Ø
KA2     FDB     $Ø,$Ø
KB1     FDB     $Ø,$Ø
KB2     FDB     $Ø,$Ø
*
*       FLOW CALIBRATIONS
*
LVM     FDB     $Ø9BF,$8ØØØ                   (383.)
LVC     FDB     $8993,$8ØØØ                   (-295.)
QSM     FDB     $8CA,$Ø                       (2Ø2.)
QSC     FDB     $8888,$Ø                      (-136.)
```

```
*
*       GILLILAND CORRELATION COEFFS
*
HNYA    FDB     $2DA,$E148      (3.42)
HNYB    FDB     $8283,$D7ØA     (-2.Ø6)
STAGES  FDB     $3DØ,$Ø         (6.5)
*
*       TRIMMING CONTROLLERS
*
KS1     FDB     $Ø,$Ø
KS8     FDB     $Ø,Ø
*
*       REFRACTOMETER CALIBRATIONS
*
DØ      RMB     12              XD POLY COEFFS
RXD     RMB     4
WØ      RMB     12              XW POLY COEFFS
RXW     RMB     4
*
*       ADAPTION DATA
*
REFLUX  RMB     4               FLOW (MOL/MIN)
RMAD    RMB     4               MIN REFLUX RATIO
NMAD    RMB     4               MIN STAGES
Y9AD    RMB     4               TEMP STORE
RAD     RMB     4               REFLUX RATIO
*
*       DECOUPLER COEFFS
*
KLV     FDB     $Ø,$Ø           (Ø.Ø)
KQS     FDB     $Ø,$Ø           (Ø.Ø)
*
```

```
*
*             **********************************
*             *                                *
*             *          MAIN PROGRAM          *
*             *  FEEDFORWARD/FEEDBACK VERSION  *
*             *                                *
*             **********************************
*
*
*       SET UP PIA'S
CONFIG  EQU     $E728           ROUTINE IN OPSYS EPROM
        ORG     $2400
MAIN1   JSR     CONFIG
*
*       RESTART ENTRY POINT
*
MAIN2   LDAA    #$37            ENABLE INTS THRU PIA 3 CA1
        STAA    V4CSR
        LDAA    #$3C
        STAA    DCSR2
        LDX     #REFRAC         CLEAR REFRAC DATA AREA
MA1     CLR     0,X             CLEAR REF DATA AREA
        INX
        CPX     #REFRAC+9
        BNE     MA1
        LDX     #TIM1           CLEAR TIMERS
MA2     CLR     0,X
        INX
        CPX     #TIM5+1
        BNE     MA2
        CLR     SYSTAT          CLEAR SYSTEM STATUS
        LDX     #TIMER          SET UP TIMER INT'S
        STX     IRQVEC
        JSR     USERP           PROMPT USER FOR I/O
        CLI
*
*       MAIN OPERATING LOOP STARTS HERE
*
MA3     TST     IOFLAG          LINE OF INPUT READY ?
        BEQ     MA4
        JSR     USERC           USER COMMAND
MA4     LDAA    TIM1            TIME TO CONTROL ?
        CMPA    TSAMPL
        BCS     MA3             NO
*       TIME TO SAMPLE AND CONTROL
        INC     SYSTAT          SYSTEM STATUS TO 1
        CLR     TIM1            CLEAR COUNTER
        JSR     READAD          GO TO DATA AQUISITION
        INC     SYSTAT          SYSTEM STATUS TO 2
        JSR     CHECK           CHECK ALARMS
        INC     SYSTAT          SYSTEM STATUS TO 3
        JSR     REFCTL          GO TO REFRAC CONTROLLER
        INC     SYSTAT          SYSTEM STATUS TO 4
        JSR     LCTRL           GO TO CONTROL ROUTINE
        INC     SYSTAT          SYSTEM STATUS TO 5
```

```
JSR     HNYFF           FEED FORWARD CONTROLLER
JSR     PIFB            FEEDBACK CONTROLLER
JSR     ADAPT           FF ADAPTION
INC     SYSTAT          SYSTEM STATUS TO 6
JSR     $2ØØ            MISCELLANEA
INC     SYSTAT          SYSTEM STATUS TO 7
JSR     VALVES          OUTPUT
CLR     SYSTAT          SYSTEM STATUS BACK TO Ø
BRA     MA3
```

```
*
*     ***********************************
*     *  FEEDFORWARD CONTROLLER PART I  *
*     ***********************************
*
        ORG     $28ØØ
HNYFF   LDAA    FFSTAT          FF ON ?
        BNE     HN1             YES
        LDAA    #$7F            NO - BIAS
        STAA    DALR+1          FB CONTROLLERS
        STAA    DAQS+1
        RTS
*
*       ENTER APU
*
HN1     LDX     #ERR
        STX     ERRADR
        JSR     APUDRV
[
*
*       COMPUTE Q
*
        IPSHF   85.
        IPSHF   23.
        PUSHF   XF
        FMUL;FSUB
        PUSHF   TF
        FSUB
        IPSHF   .ØØ2
        FMUL
        IPSHF   1.
        FADD
        POPF    Q
*
*       COMPUTE RM
*       FIRST FIND EQULIB LINE INTERCEPT (X)
*
        PUSHF   XF
        PUSHF   Q
        IPSHF   1.
        FSUB;FDIV
        IPSHF   .498
        FADD
        PUSHF   Q
        PTOF
        IPSHF   1.
        FSUB;FDIV
        IPSHF   .566
        FSUB;FDIV
*
*       SECOND , FIND Y* FOR X INTERCEPT
*
        PTOF
        IPSHF   .566
        FMUL
        IPSHF   .498
        FADD
*
*       THIRD FIND RM
*
        POPF    Y9
        PUSHF   Y9
        XCHF;FSUB
        PUSHF   XD
        PUSHF   Y9
        FSUB;XCHF;FDIV
        POPF    RM
*
*       COMPUTE NM
*
        PUSHF   XD
        PUSHF   XB
        FMUL;CHSF;PTOF
        PUSHF   XD
        FADD;XCHF
        PUSHF   XB
        FADD;FDIV;LOG
        IPSHF   1.6
        FMUL
        POPF    NM
*
*       COMPUTE R FROM GILLILAND CORRELATION
*
        PUSHF   RM
        IPSHF   1.
        FADD
        PUSHF   HNYB
        FMUL
        PUSHF   NM
        PUSHF   STAGES
        FDIV;EXP
        PUSHF   HNYA
        FSUB;FDIV
        IPSHF   1.
        FSUB
        POPF    R
*
*       COMPUTE F (GMOLE/MIN)
*
        IPSHF   220.
        PUSHF   XF
        FMUL
        IPSHF   992.
        XCHF;FSUB
        IPSHF   14.
        PUSHF   XF
        FMUL
        IPSHF   18.
        FADD;FDIV
        PUSHF   F
        FMUL
        POPF    FM
```

```
*
*       COMPUTE LR (GMOLE/MIN)
*
        PUSHF   XF
        PUSHF   XB
        FSUB
        PUSHF   XD
        PUSHF   XB
        FSUB;FDIV
        PUSHF   FM
        FMUL
        PUSHF   R
        FMUL
        POPF    LM
*
*       COMPUTE LR (L/MIN)
*
        IPSHF   14.
        PUSHF   XD
        FMUL
        IPSHF   18.
        FADD
        PUSHF   LM
        FMUL
        IPSHF   957.
        IPSHF   170.
        PUSHF   XD
        FMUL;FSUB;FDIV
        POPF    LV
*
*  COMPUTE VM (GMOLE/MIN)
*  USING THE NON-EQUIMOLAL
*  OVERFLOW ANALYSIS
*
*  CALCULATE D
        PUSHF   XF
        PUSHF   XB
        FSUB
        PUSHF   XD
        PUSHF   XB
        FSUB;FDIV
        PUSHF   FM
        FMUL
        PTOF
        POPF    RM              SAVE DISTILLATE FLOW IN RM
*  CALCULATE THE INTERNAL REFLUX FLOW
        PUSHF   R
        FMUL
        IPSHF   1.11            (R(INT)=1.11R(EXT))
        FMUL
*  CALCULATE LIQUID FLOW ABOVE THE FEED PLATE
        IPSHF   7.55
        PUSHF   XD
        FSUB;FMUL
        IPSHF   7,55
        PUSHF   XF
        FSUB;FDIV
*  CALCULATE LIQUID FLOW BELOW THE FEED TRAY
        PUSHF   FM
        PUSHF   Q
        FMUL;FADD
*  CALCULATE LIQUID FLOW INTO REBOILER
        IPSHF   7.55
        PUSHF   XF
        FSUB;FMUL
        IPSHF   7.35
        FDIV
*  COMPUTE VAPOUR FLOW FROM REBOILER
        PUSHF   FM
        FSUB
        PUSHF   RM              DISTILLATE FLOW STORED IN RM
        FADD
        POPF    VM
*
*       COMPUTE QS (KG/MIN)
*
        PUSHF   VM
        IPSHF   55.2
        FDIV
        POPF    QS
        UNTHRD
[
*
*       JUMP TO SECOND PART OF FF CONTROLLER
*
FFCOMP  EQU     $2AØØ
        JMP     FFCOMP
*
*       APU ERROR HANDLER
*
[
ERR     UNTHRD
[
        LDX     #ERRM
        SWI
        FCB     $12
        RTS
*
*       ERROR MESSAGE
*
ERRM    FCC     /APU ERROR IN FF CONTROLLER PT 1/
        FCB     $7,$D,$A,$4
        END
```

```
*
*    ***********************************
*    *  FEEDFORWARD CONTROLLER PART II  *
*    ***********************************
*
*
*       FEEDFORWARD COMPENSATOR
*
        ORG     $2AØØ
*
*       INTO THE APU
*
FFCOMP  LDX     #ERR
        STX     ERRADR
        JSR     APUDRV
[
*
*       FIRST LR LOOP
*
        PUSHF   LV
        PTOF
        PUSHF   YNP1A
        FSUB
        PUSHF   KA1
        FMUL;FADD
        POPF    ZNA
        PUSHF   YNP1A
        PTOF
        PUSHF   LV
        FSUB;CHSF
        PUSHF   KA2
        FMUL;FADD
        POPF    YNP1A
*
*       NOW QS LOOP
*
        PUSHF   QS
        IPSHF   Ø.Ø7            (ADD .Ø7 KG/MIN FOR LOSSES)
        FADD;PTOF
        POPF    QS
        PTOF
        PUSHF   YNP1B
        FSUB
        PUSHF   KB1
        FMUL;FADD
        POPF    ZNB
        PUSHF   YNP1B
        PTOF
        PUSHF   QS
        FSUB;CHSF
        PUSHF   KB2
        FMUL;FADD
        POPF    YNP1B
```

```
*
*       COMPUTE D/A OUTPUT BYTES,REFLUX FIRST
*
        PUSHF   ZNA
        PUSHF   LVM
        FMUL
        PUSHF   LVC
        FADD
        FIXS              CONVERT TO INTEGER AND STORE
        POPS    DALR
*
*       NOW STEAM FLOW
*
        PUSHF   ZNB
        PUSHF   QSM
        FMUL
        PUSHF   QSC
        FADD
        FIXS              CONVERT TO INTEGER AND STORE
        POPS    DAQS
        UNTHRD
[
*
*       CHECK COMPUTED VALVE POSITIONS IN THE RANGE Ø-FF
*
        LDAB    DALR+1            FIRST LR
        LDAA    DALR
        BMI     A1                < Ø
        BEQ     A2                Ø<= # <= FF
        LDAB    #$FF              > FF
        BRA     A2
A1      CLRB                      < Ø
A2      STAB    DALR+1
        CLRA
        STAA    DALR              CLEAR UPPER BYTE
        LDAB    DAQS+1            NOW QS
        LDAA    DAQS
        BMI     A3
        BEQ     A4                Ø<= # <= FF
        LDAB    #$FF
        BRA     A4
A3      CLRB                      <Ø
A4      STAB    DAQS+1
        CLRA
        STAA    DAQS              CLEAR UPPER BYTE
*
*       CONVERT REFRACTOMETER DATA TO M.F.
*
*       GET DATA FROM DATABASE
        LDAA    $8                MSB XD
        ANDA    #$ØF              STRIP BITS
        STAA    RXD
        LDAA    $9                LSB
        STAA    RXD+1
        LDAA    $A                MSB XW
        ANDA    #$ØF            STRIP BITS
        STAA    RXW
        LDAA    $B      LSB
        STAA    RXW+1
*       CONVERT DATA
        JSR     APUDRV
[
        PUSHS   RXD
        FLTS
        PTOF
        POPF    RXD
        PUSHF   D2
        FMUL
        PUSHF   D1
        FADD
        PUSHF   DØ
        FADD
        POPF    RXD
        PUSHS   RXW
        FLTS
        PTOF
        POPF    RXW
        PUSHF   W2
        FMUL
        PUSHF   W1
        FADD
        PUSHF   RXW
        FMUL
        PUSHF   WØ
        FADD
        POPF    RXW
        UNTHRD
[
*
*       INTEGRAL TRIMMING CONTROLLERS
*       ACTING ON THE TEMPERATURE
*       CONTROL LOOP SETPOINTS
*
        TST     $FE             CONTROLLERS ON ?
        BNE     SE1             YES
        RTS                     NO
*       USE APU TO COMPUTE CONTROLLERS
SE1     JSR     APUDRV
[
        PUSHF   XD              T1 LOOP
        PUSHF   RXD
        FSUB                    ERROR
        PUSHF   KS1             GAIN
        FMUL;FIXS
        PUSHS   SP1             ADD TO SP
        SADD
        POPS    SP1             AND STORE
        PUSHF   X8              T8 LOOP
        PUSHF   RXW
        FSUB                    ERROR
        PUSHF   KS8             GAIN
        FMUL;FIXS
        PUSHS   SP4             ADD TO SP
        SADD
        POPS    SP4
        UNTHRD
[
        RTS
*
*       ERROR HANDLER
*
[
ERR     UNTHRD
[
        LDX     #ERRM
        SWI
        FCB     $12
        RTS
*
*       ERROR MESSAGE
*
ERRM    FCC     /APU ERROR IN FEED FORWARD CALCULATION/
        FCB     $7,$D,$A,$7,$4
        RTS
        END
```

```
*
*   *************************************************
*   *   FEEDBACK SECTION OF FEEDFORWARD CONTROLLER  *
*   *************************************************
*
        ORG     $2CØØ
*
*       COMPUTE DV FOR LV LOOP
*
PIFB    LDX     #Ø
        CPX     C1A             CONTROLLER OFF (K=Ø) ?
        BEQ     CTL2            YES
        LDX     C1A+5           SAVE LAST ERROR
        STX     C1A+7
        LDX     #DAM            SET UP POINTER TO DATA BASE
        LDAA    C1A+4
        ASLA
        JSR     ADJX
        LDAA    Ø,X
        ANDA    #$ØF            REMOVE CH ID
        STAA    TEMP1
        LDAA    1,X
        STAA    TEMP1+1
        LDAA    SP1             GET SP
        LDAB    SP1+1
        SUBB    TEMP1+1
        SBCA    TEMP1
        STAA    C1A+5           SAVE ERROR
        STAB    C1A+6
        LDX     #ERRM
        STX     ERRADR
        JSR     APUDRV
*
*       FIRST LV LOOP (USE F.P. TO PREVENT OVERFLOW)
*
[
        PUSHS   C1A
        FLTS
        PUSHS   C1A+5
        FLTS;FMUL
        PUSHS   C1A+2
        FLTS
        PUSHS   C1A+7
        FLTS;FMUL;FADD
        POPF    MULT1           SAVE DV
        UNTHRD
[
        BRA     CTL3
CTL2    LDX     #$8Ø
        STX     MULT1           SET DV=Ø
        LDX     #$Ø
        STX     MULT1+2
```

```
*
*       NOW CALCULATE DV FOR QS LOOP
*
CTL3    LDX     #Ø              
        CPX     C4                      CONTROLLER OFF ?
        BEQ     CTL4
        LDX     C4+5                    SAVE LAST ERROR
        STX     C4+7
        LDX     #DAM                    GET PV
        LDAA    C4+4
        ASLA
        JSR     ADJX
        LDAA    Ø,X
        ANDA    #$ØF                    REMOVE CH ID
        STAA    TEMP1
        LDAA    1,X
        STAA    TEMP1+1
        LDAA    SP4                     GET SP
        LDAB    SP4+1
        SUBB    TEMP1+1
        SBCA    TEMP1
        STAA    C4+5                    SAVE ERROR
        STAB    C4+6
        LDX     #ERRM
        STX     ERRADR
        JSR     APUDRV
[
        PUSHS   C4
        FLTS
        PUSHS   C4+5
        FLTS;FMUL
        PUSHS   C4+2
        FLTS
        PUSHS   C4+7
        FLTS;FMUL;FADD
        POPF    PROD                    SAVE DV
        UNTHRD
[
        BRA     CTL5
CTL4    LDX     #$8Ø                    SET DV=Ø CONTROLLER OFF
        STX     PROD
        LDX     #$Ø
        STX     PROD+2
*
*       NOW DECOUPLE OUTPUTS
*
CTL5    LDX     #ERRM
        STX     ERRADR
        JSR     APUDRV
[
        PUSHF   PROD
        PUSHF   KLV
        FMUL
        PUSHF   MULT1
        FADD
        PUSHF   VPN1
        FADD
        PTOF
        POPF    VPN1
        IPSHF   256.
        FDIV;FIXS
        POPS    TEMP1
*       NOW QS LOOP
        PUSHF   MULT1
        PUSHF   KQS
        FMUL
        PUSHF   PROD
        FADD
        PUSHF   VPN4
        FADD
        PTOF
        POPF    VPN4
        IPSHF   256.
        FDIV;FIXS
        POPS    TEMP2
        UNTHRD
[
*
*       NOW COMBINE FF AND FB CONTROLLERS
*
*       LV LOOP FIRST
        LDAA    TEMP1           GET FB OUTPUT
        LDAB    TEMP1+1
        ADDB    DALR+1          ADD ON FF OUTPUT
        ADCA    DALR
        JSR     OVERFL          CHECK FOR OVERFLOW
        TSTA                    RESULT Ø-$FF ?
        BMI     CTL6            <Ø
        BEQ     CTL7            OK
        LDAB    #$FF            >$FF
        BRA     CTL7
CTL6    CLRB
CTL7    STAB    VNEW1           SAVE FOR OUTPUT LATER
*       QS LOOP
        LDAA    TEMP2           FB OUTPUT
        LDAB    TEMP2+1
        ADDB    DAQS+1          ADD ON FF OUTPUT
        ADCA    DAQS
        JSR     OVERFL          CHECK FOR OVERFLOW
        TSTA                    IN RANGE Ø-$FF ?
        BMI     CTL8            <Ø
        BEQ     CTL9            OK
        LDAB    #$FF            .>$FF
        BRA     CTL9
CTL8    CLRB
CTL9    STAB    VNEW4           SAVE FOR OUTPUT LATER
*
*       COMPLETE
*
        RTS
```

```
*
*       SUBROUTINE TO INCR X BY CONTENTS OF A (UNSIGNED)
*
ADJX    TSTA
        BEQ     AD2
AD1     INX
        DECA
        BNE     AD1
AD2     RTS
*
*       OVERFLOW SUBROUTINE
*       DETECTS OVERFLOWS ON 16 BIT INTEGER ARITHMETIC
*       USING A,B ACCUMULATORS. (A=MSB)
*
OVERFL  BVS     OV2             OVEFLOW?
        RTS                     NO
OV2     BCS     OV1             NEGATIVE?
        LDAB    #$FF            NO SET TO MAX
        LDAA    #$7F
        RTS
OV1     CLRB                    YES SET TO MIN
        LDAA    #$80
        RTS
*
*       ERROR HANDLER
*
[
ERRM    UNTHRD
[
        LDX     #ERRMS
        SWI
        FCB     PMSG
        RTS
*
*       ERROR MESSAGE
*
ERRMS   FCC     /APU ERROR IN PIFB ROUTINE/
        FCB     $7,$D,$A,$4
        END
```

```
*
*    *************************************************
*    *  ADAPTION ROUTINE FOR FEEDFORWARD CONTROLLER  *
*    *************************************************
*
*       MODIFIES STAGES USING THE GILLILAND CORRELATION
*       IF THE BYTE AT $FD <> 0
*
        ORG     $3000
ADAPT   TSTA    $FD             ADAPT ?
        BNE     AD1             YES
        RTS                     NO
*
*       CALCULATE CURRENT REFLUX FLOW
*
AD1     LDAA    VNEW1
        STAA    REFLUX+1
        CLRA
        STAA    REFLUX
*
*       INTO APU
*
        LDX     #ERR
        STX     ERRADR
        JSR     APUDRV
[
*
*       COMPUTE REFLUX IN MOL/MIN
*
        PUSHS   REFLUX
        FLTS
        PUSHF   LVC
        FSUB
        PUSHF   LVM
        FDIV
        PUSHF   RXD             DENSITY
        IPSHF   -170.
        FMUL
        IPSHF   957.
        FADD;FMUL
        PUSHF   RXD             M.WT.
        IPSHF   14.
        FMUL
        PUSHF   18.
        FADD;FDIV
        POPF    REFLUX          SAVE
*
*       COMPUTE RM
*
        PUSHF   XF
        PUSHF   Q
        IPSHF   1.
        FSUB;FDIV
        IPSHF   .498
        FADD
        PUSHF   Q
        PTOF
        IPSHF   1.
        FSUB;FDIV
        IPSHF   .566
        FSUB;FDIV
        PTOF
        IPSHF   .566
        FMUL
        IPSHF   .498
        FADD
        PTOF
        POPF    Y9AD
        XCHF;FSUB
        PUSHF   RXD
        PUSHF   Y9AD
        FSUB;XCHF;FDIV
        POPF    RMAD
*
*       COMPUTE NM
*
        PUSHF   RXD
        PUSHF   RXW
        FMUL;CHSF;PTOF
        PUSHF   RXD
        FADD;XCHF
        PUSHF   RXW
        FADD;FDIV;LOG
        IPSHF   1.6
        FMUL
        POPF    NMAD
*
*       COMPUTE R
*
        PUSHF   REFLUX
        PUSHF   RXD
        PUSHF   RXW
        FSUB;FMUL
        PUSHF   FM
        FDIV
        PUSHF   XF
        PUSHF   RXW
        FSUB;FDIV
        POPF    RAD
*
*       COMPUTE STAGES
*
        PUSHF   RMAD
        IPSHF   1.
        FADD
        PUSHF   RAD
        IPSHF   1.
        FADD;FDIV
        PUSHF   HNYB
        FMUL
        PUSHF   HNYA
        FADD;LN
        PUSHF   NMAD
        XCHF;FDIV
        POPF    STAGES
*
*       EXIT FROM APU
*
        UNTHRD
[
        CLR     $FD                     CLEAR FLAG
        RTS
*
*       ERROR HANDLER
*
[
ERR     UNTHRD
[
        LDX     #ERRM
        SWI
        FCB     $12
        RTS
*
*       ERROR MESSAGE
*
ERRM    FCC     /APU ERROR IN ADAPT/
        FCB     $7,$D,$A,$4
        END
```

APPENDIX V

CROSS-ASSEMBLER AND PRE-ASSEMBLER

A cross-assembler, XASMBL, was written to produce executable code for the Motorola M6800 microprocessor. The documentation manual for this program is given at the end of this appendix. A pre-assembler, PREASS, was written to translate the mneumonics and macros associated with the Am9511 arithmetic processing unit (APU) described in Chapter 4, into standard assembly language form for XASMBL.

The macros defined in PREASS were divided into three groups:

(i) Those involving APU commands (single byte);

(ii) those involving direct mode APU input/output (single byte command, two byte address);

(iii) those involving immediate mode APU input (single byte command, data bytes).

For case (i) the macro was replaced by an assembler form-constant-byte directive with the appropriate byte, e.g.

```
FMUL
```

was replaced by

```
FCB  18
```

for a floating point multiply operation. Several APU commands could be given on a single line using a semicolon dilimiter. In case (ii), the command was also replaced by a form-constant-byte directive, but was followed by a form-double-byte directive containing the address of the source bytes or destination bytes for the command, e.g.

```
PUSHF  NUM
```

was replaced by

```
FCB    95
FDB    #NUM
```

for a floating point load into the APU where #NUM is the address of the variable NUM . The FDB directive was replaced by the actual address of NUM by XASMBL . In case (iii), the command was again replaced by a form-constant-byte directive, followed by either two or four bytes of data defined in one or two form-double-byte directives. The data for the immediate mode operations was placed within the threaded code. The only operations using immediate mode were the push on to the APU stack operations: IPSHS, IPSHD, IPSHF. PREASS allowed the data associated with immediate mode operations to be represented in any standard FORTRAN type form, and translated this form to the appropriate bytes for the APU, e.g.

```
                    IPSHF  2.0
```

was replaced by

```
                    FCB     92    (the command code)
                    FDB    640,0  (APU representation in floating
                                   point of 2.0)
```

In this example the floating point value 2.0 would be pushed on to the APU stack.

PREASS processed code between two delimiters which were chosen to be left square brackets [($5B) . All other code outside the delimiters was left untouched. All translations of mneumonics, addresses, and constants were decimal based, and eventually converted to hexadecimal base by XASMBL. PREASS also supplied definitions of the starting address of APUDRV, the APU interpreter, and the error handler pointer address ERRADR as

```
        APUDRV    EQU    $2000   (The  $  indicates a base 16 number)
        ERRADR    EQU    $FFE7
```

These definitions were made at the beginning of the source program.

Operation of PREASS was straightforward; two files were used, one was named as the input file and contained the threaded code sections to be translated, the second was named as the output file containing the translated code. Any errors detected by PREASS were marked in the output file as an assembler comment preceded by an asterisk. The total number of errors was displayed on the console to the user.

A sample program showing the format of the pre-assembler source code is shown in figure V-1, along with the output file from PREASS showing the translated sections. Table V-1 is a summary of the available commands to APUDRV.

TABLE V-1

SUMMARY OF APUDRV COMMANDS

HEXADECIMAL CODE

	00	10	20	30	40	50	60	70
0	NOP	FADD	-	-	BEQ	UNTHRD	-	-
1	SQRT	FSUB	-	-	BNE	LOOP	-	-
2	SIN	FMUL	-	-	BLT	PRINT	-	-
3	COS	FDIV	-	-	BGE	ENTER	-	-
4	TAN	-	-	CHSD	BLE	-	-	CHSS
5	ASIN	CHSF	-	-	BGT	-	-	-
6	ACOS	-	-	DMUU	BCS	-	-	SMUU
7	ATAN	PTOF	-	PTOD	BCC	POPS	-	PTOS
8	LOG	RUPF	-	RUPD	BRA	POPD	-	RUPS
9	LN	XCHF	-	XCHD	-	POPF	-	XCHS
A	EXP	PUPI	-	-	BEN	IPSHS	-	-
B	PWR	-	-	-	BEZ	IPSHD	-	-
C	-	FLTD	DADD	-	BEA	IPSHF	SADD	-
D	-	FLTS	DSUB	-	BEU	PUSHS	SSUB	-
E	-	FIXD	DMUL	-	BEO	PUSHD	SMUL	-
F	-	FIXS	DDIV	-	BER	PUSHF	SDIV	-

```
*
*       DEMONSTRATION SUBROUTINE USING
*       THE AM9511 ARITHMETIC PROCESSOR UNIT
*       AND THE PRE-ASSEMBLER, PREASS.
*
        ORG     $0
ENTRY   LDX     #ERR                SET UP POINTER
        STX     ERRADR              TO ERROR HANDLER.
        JSR     APUDRV              ENTER THREADED CODE.
[                                   SIGNAL PREASS TO START PROCESSING.
        PUSHF   MULT1               PUSH F.P. VARIABLE MULT1 INTO APU.
        PUSHF   MULT2               PUSHF F.P. VARIABLE MULT2 INTO APU.
        FMUL                        MULTIPLY S*T.
        POPF    RESULT              STORE ANSWER IN RESULT.
        UNTHRD                      EXIT FROM THREADED CODE.
[                                   SIGNAL PREASS TO STOP PROCESSING.
        RTS                         RETURN TO CALLING PROGRAM.
*
*       ERROR HANDLER.
*
[                                   SIGNAL PREASS TO START PROCESSING.
ERR     UNTHRD                      EXIT FROM THREADED CODE.
[                                   SIGNAL PREASS TO STOP PROCESSING.
*       <PRINT ERROR MESSAGE>
        RTS                         RETURN TO CALLING PROGRAM.
*
*       DATA
*
MULT1   FDB     $0280,$0000                 2.0 (FLOATING POINT)
MULT2   FDB     $02C0,$0000                 3.0 (FLOATING POINT)
RESULT  RMB     2
        END
```

(a) SOURCE PROGRAM

```
APUDRV  EQU     $6000
ERRADR  EQU     $FFE7
*
*       DEMONSTRATION SUBROUTINE USING
*       THE AM9511 ARITHMETIC PROCESSOR UNIT
*       AND THE PRE-ASSEMBLER, PREASS.
*
        ORG     $0
ENTRY   LDX     #ERR                SET UP POINTER
        STX     ERRADR              TO ERROR HANDLER.
        JSR     APUDRV              ENTER THREADED CODE.
*[                                  SIGNAL PREASS TO START PROCESSING.
*       PUSHF   MULT1               PUSH F.P. VARIABLE MULT1 INTO APU.
        FCB     95
        FDB     #MULT1
*       PUSHF   MULT2               PUSHF F.P. VARIABLE MULT2 INTO APU.
        FCB     95
        FDB     #MULT2
*       FMUL                        MULTIPLY S*T.
        FCB     18
*       POPF    RESULT              STORE ANSWER IN RESULT.
        FCB     89
        FDB     #RESULT
*       UNTHRD                      EXIT FROM THREADED CODE.
        FCB     80
*[                                  SIGNAL PREASS TO STOP PROCESSING.
        RTS                         RETURN TO CALLING PROGRAM.
*
*       ERROR HANDLER.
*
*[                                  SIGNAL PREASS TO START PROCESSING.
*ERR    UNTHRD                      EXIT FROM THREADED CODE.
ERR     FCB     80
*[                                  SIGNAL PREASS TO STOP PROCESSING.
*       <PRINT ERROR MESSAGE>
        RTS                         RETURN TO CALLING PROGRAM.
*
*       DATA
*
MULT1   FDB     $0280,$0000                 2.0 (FLOATING POINT)
MULT2   FDB     $02C0,$0000                 3.0 (FLOATING POINT)
RESULT  RMB     2
        END
```

(b) PREASS OUTPUT

FIGURE V-1 PREASS EXAMPLE

XASMBL

A CROSS-ASSEMBLER FOR THE MOTOROLA M6800
TO RUN ON A PDP11 UNDER RT-11.

INTRODUCTION

XASMBL IS A CROSS-ASSEMBLER FOR THE M68ØØ MICROPROCESSOR, WRITTEN IN FORTRAN FOR EXECUTION ON A PDP11 COMPUTER UNDER THE RT-11 OPERATING SYSTEM. IT IS A SUBSET OF THE MOTOROLA AND AMI ASSEMBLERS, THE MAIN DIFFERENCES BEING THAT MACROS, CONDITIONAL ASSEMBLY AND RELOCATABLE CODE ARE NOT SUPPORTED. THESE RESTRICTIONS ARE EASILY OVERCOME USING THE RT-11 UTILITIES AND EDITOR, BUT ALLOW THIS CROSS ASSEMBLER TO RUN IN AS LITTLE AS 12K OF MEMORY.

PROGRAM CONSTRUCTION

```
OVERLAID (FOR 12K SYSTEMS)        NON-OVERLAID

.R LINK                             .R LINK
*XASMBL<XAS1/F/C<CR>                *XASMBL<XAS1,XAS2,XAS3,XAS4/F<CR>
*XAS2/O:1/C                         *
*XAS3/O:1/C
*XAS4/O:2
*
```

XASMBL.SAV IS READY TO BE USED.

PROGRAM USE

WHEN XASMBL IS EXECUTED, IT ASKS FOR THE SOURCE, LOAD AND LIST FILE NAMES IN TURN, FOLLOWING THE RT-11 SYSTEM CONVENTIONS, FOR EXAMPLE:-

```
.R XASMBL

68ØØ MICROPROCESSOR CROSS ASSEMBLER, VERSION 1.3A

SOURCE FILE? RK:CONTRL.MIC<CR>

LOAD FILE? PP:<CR>

LIST FILE? CONTRL.LST<CR>

ERRORS DETECTED  Ø
```

SAMPLE PROGRAM

THE LIST OUTPUT FROM XASMBL FOR THE SUPPLIED SAMPLE PROGRAM FOLLOWS. THIS SAMPLE PROGRAM CAN BE USED TO CHECK THE OPERATION OF THE ASSEMBLER PROGRAM.

```
TEST FOR MUL8                 XASMBL-S6800 ASSEMBLER PAGE  1
                             NAM     TEST FOR MUL8
      0100                   ORG     $100
0100       96                LDAA    $10
           10
0102       D6                LDAB    $20
           20
0104       BD                JSR     MUL8    CALL MULTIPLY S/R
           01
           0C
0107       3F                SWI
0108       81                FCB     $81
0109       7E                JMP     $F000      RETURN TO PROTO
           F0
           00
               *
               *       MUL8 MULTIPLIES THE UNSIGNED 8-BIT INTEGERS
               * IN THE A AND B REGISTERS, TO GIVE THE 16-BIT
               * RESULT IN A,B.
               *
010C       36  MUL8      PSHA
010D       B6            LDAA    EIGHT
           01
           21
0110       36            PSHA
0111       30            TSX
0112       4F            CLRA
0113       56            RORB
0114       24  MT        BCC     MF
           02
0116       AB            ADDA    1,X
           01
0118       46  MF        RORA
0119       56            RORB
011A       6A            DEC     0,X
           00
011C       26            BNE     MT
           F6
011E       31            INS
011F       31            INS
0120       39            RTS
0121       08  EIGHT     FCB     8
                         END
ERRORS DETECTED  0




TEST FOR MUL8                 XASMBL-S6800 ASSEMBLER PAGE  2
                           SYMBOL TABLE
MUL8    010C   EIGHT   0121   MT      0114   MF      0118
```

SOURCE PROGRAM FORMAT

THE SOURCE PROGRAM MUST RESIDE ON A MASS STORAGE DEVICE, AND CONSIST OF A SEQUENCE OF LINES IN THE FORM:-

<LABEL> <OPERATOR> <OPERAND> <COMMENT>

THE LABEL AND COMMENT FIELDS ARE OPTIONAL AND THE OPERAND FIELD IS INCLUDED AS REQUIRED. THE FIELDS ARE SEPARATED BY ONE OR MORE BLANKS OR TABS.

BLANK LINES ARE ILLEGAL (SEE THE ASSEMBLER DIRECTIVE SPC). COMMENT LINES CAN BE INDICATED BY AN ASTERISK (*) IN COLUMN ONE. THE LABEL FIELD, IF PRESENT, MUST START IN COLUMN ONE, AND THE FIRST DELIMITING BLANK IS NOT OPTIONAL. THE MAXIMUM LINE LENGTH IS 8Ø CHARACTERS.

OUTPUT FILE FORMAT

TWO FILES ARE CREATED BY XASMBL. THE ASSEMBLED LOAD FILE IS FORMATTED IN RECORDS TO THE MOTOROLA AND THE AMI HEX TAPE FORMAT. IT CAN BE STORED ON A MASS STORAGE DEVICE FOR SUBSEQUENT PUNCHING TO PAPER TAPE OR PUNCHED IMMEDIATELY, TO BE LOADED INTO THE MICROPROCESSOR.

THE SECOND FILE IS A LISTING IN THE FORMAT OF THE FOLLOWING:-

```
ØØFF   CE  LADR    LDX     #STRING    COMMENT
       Ø1
```

ØØFF IS THE ADDRESS OF THE ASSEMBLED INSTRUCTION.
CE (AND BELOW) IS THE GENERATED PROGRAM CODE.
LADR IS THE LABEL.
THE OPERATION CODE MNEUMONIC IS LDX. THIS POSITION MAY ALTERNATIVELY SHOW AN ASSEMBLER DIRECTIVE.
#STRING IS THE OPERAND.

ALL LISTED VALUES, OPERATION CODES AND INSTRUCTIONS ARE LISTED IN THE HEXADECIMAL NUMBER BASE.

SYMBOLS

XASMBL MAINTAINS A SINGLE SYMBOL TABLE CONTAINING THE PERMANENT AND USER DEFINED SYMBOLS. PERMANENT SYMBOLS ARE THE INSTRUCTION MNEUMONICS AND ASSEMBLER DIRECTIVES, AND NEED NOT BE DEFINED. USER SYMBOLS ARE THOSE USED AS LABELS OR DEFINED BY THE EQU DIRECTIVE. THEY MUST BE CONSTRUCTED AS FOLLOWS:-

- THE FIRST CHARACTER MUST BE ALPHABETIC
- UP TO FIVE ALPHANUMERIC CHARACTERS MAY FOLLOW
- EACH SYMBOL MUST BE UNIQUE
- SPACES, TABS AND OTHER SPECIAL CHARACTERS ARE ILLEGAL WITHIN THE SYMBOL.

LOCAL SYMBOLS ARE NOT SUPPORTED.

ASSEMBLY LOCATION COUNTER

THE ASTERISK (*) REPRESENTS THE ASSEMBLY LOCATION COUNTER (IE THE ADDRESS OF THE CURRENT INSTRUCTION) WHEN USED IN THE OPERAND FIELD. FOR EXAMPLE, THE STATEMENT

```
        ORG    * + $100
```

INCREMENTS THE ASSEMBLY LOCATION COUNTER BY 256, LEAVING AN UNCHANGED 256 BYTE BLOCK.

NUMBERS

NUMBERS CAN BE REPRESENTED IN ANY ONE OF FOUR BASES - BINARY, OCTAL, DECIMAL OR HEXADECIMAL. NUMBERS HAVE THE GENERAL FORM

```
        <QUALIFIER><NUMBER>
```

WHERE <QUALIFIER> CAN BE

```
        % FOR BINARY, FOR EXAMPLE, %10 = 2(BASE TEN)
        @ FOR OCTAL, FOR EXAMPLE, @10 = 8(BASE TEN)
        $ FOR HEXADECIMAL, FOR EXAMPLE, $10 = 16(BASE TEN)
```

ANY NUMBER WITHOUT QUALIFIER IS ASSUMED TO BE DECIMAL.

ADDRESSING MODES

(1) INHERENT AND ACCUMULATOR MODES

IN THESE MODES THE OPERAND IS IMPLIED BY THE OPERATION CODE, FOR EXAMPLE:-

```
        TSTA
        INX
```

NOTE THAT FOR ALL OPERATION CODES INVOLVING AN ACCUMULATOR, THERE MUST NOT BE A BLANK BETWEEN THE OPERATION CODE AND THE ACCUMULATOR INDICATOR, FOR EXAMPLE:-

```
        TST A        IS ILLEGAL
        TSTA         IS VALID
```

(2) IMMEDIATE MODE

FOR THIS MODE, THE OPERAND IS ASSEMBLED AS PART OF THE INSTRUCTION ITSELF IN ONE OF THE FORMS:-

```
                #<VARIABLE>
             OR #<NUMBER>
             OR #'<SINGLE CHARACTER>
```

FOR EXAMPLE:-

```
  LDAA #LOCN   THE ADDRESS OF LOCN IS PUT INTO ACCUMULATOR A
  LDAA #$16    THE HEXADECIMAL NUMBER 16 IS LOADED INTO A
  LDAB #'N     THE ASCII CODE FOR THE CHARACTER N IS PUT INTO B
```

(3) INDEXED MODE

IN THIS MODE, THE EFFECTIVE OPERAND ADDRESS IS THE SUM OF AN OFFSET PLUS THE CONTENTS OF THE INDEX REGISTER, FOR EXAMPLE:-

```
          LDAA  17,X
```

THE ACCUMULATOR WILL BE LOADED WITH THE DATA AT THE ADDRESS GIVEN BY THE CONTENT OF THE X REGISTER PLUS 17.

(4) RELATIVE MODE

FOR BRANCH INSTRUCTIONS,THE TARGET ADDRESS OF THE BRANCH IS THE ADDRESS OF THE NEXT INSTRUCTION PLUS THE OPERAND, WHICH MUST BE IN THE RANGE -128 TO +127. FOR EXAMPLE:-

```
    BGT   ERROR   IF THE CONDITION CODES ARE SET, EXECUTION
                     WILL CONTINUE FROM THE LABEL ERROR.
```

(5) DIRECT AND EXTENDED MODE

THE FORM OF THESE TWO MODES OF ADDRESSING IS THE SAME, FOR EXAMPLE:-

```
          LDAA STORE
          LDAB  DATA
```

THE DIRECT MODE IS USED IF THE SYMBOL HAS BEEN DEFINED IN THE FIRST 256 WORDS OF CORE, AND USES ONLY TWO BYTES. OTHERWISE, EXTENDED ADDRESSING IS USED, REQUIRING THREE BYTES.

ASSEMBLER DIRECTIVES

END - END OF SOURCE PROGRAM. END MUST BE PRESENT AS THE LAST LINE OF THE SOURCE PROGRAM.

EQU - EQUATE SYMBOL. THIS DIRECTIVE ASSIGNS TO A SYMBOL A VALUE, WHICH MAY BE A NUMBER OR AN EXPRESSION IN TERMS OF PREVIOUSLY DEFINED SYMBOLS, FOR EXAMPLE:-

```
           DATA EQU $FC        DATA WILL BE USED AS HAVING THE VALUE -4
           ADDR EQU DATA-8     ADDR WILL BE TAKEN AS -12
```

FCB - FORM CONSTANT BYTE. FCB MAY HAVE A NUMBER OF ARGUMENTS SEPARATED BY COMMAS, WITH ONE BYTE BEING ALLOCATED TO HOLD THE VALUE OF EACH OPERAND. SUCCESSIVE COMMAS GENERATE ZERO BYTES.

```
        FCB     ,1,2    THREE BYTES CONTAINING Ø,1,2 ARE FORMED
```

FCC - FORM CONSTANT CHARACTERS. THE FCC DIRECTIVE TRANSLATES CHARACTERS INTO THEIR 7-BIT ASCII EQUIVALENTS AND ALLOCATES STORAGE. TWO FORMATS MAY BE USED:-

(1) <COUNT>,<TEXT> WHERE <COUNT> IS THE NUMBER OF CHARACTERS TO BE TRANSLATED IN <TEXT>.

(2) <DELIMITER><TEXT><DELIMITER> WHERE <DELIMITER> MAY BE ANY NON-BLANK CHARACTER, NOT OCCURRING IN <TEXT>, BUT SERVING TO DEFINE ITS EXTENT. IF <DELIMITER> IS A NUMBER, THEN <TEXT> MUST NOT BEGIN WITH A NUMBER. FOR EXAMPLE:-

```
       MSSGE   FCC    11,INPUT ERROR
       MSSGE   FCC    !INPUT ERROR!    THESE TWO LINES ARE EQUIVALENT
```

FDB - FORM DOUBLE BYTES. THIS DIRECTIVE IS SIMILAR TO THE FCB DIRECTIVE EXCEPT THAT EACH OPERAND IS ASSIGNED TWO BYTES, FOR EXAMPLE:-

```
       ADDRS  FDB   ,15,$EF,$AFF
```

WILL GENERATE ØØ ØØ ØØ ØF ØØ EF ØA FF.

NAM - NAME. THIS DIRECTIVE OUTPUTS A HEADER RECORD TO THE OBJECT FILE AND HEADS EACH PAGE OF THE LISTING WITH THE TEXT WHICH IS THE OPERAND. ONLY THE FIRST 3Ø CHARACTERS ARE SIGNIFICANT. FOR EXAMPLE:-

```
       NAM   DEMONSTRATION PROGRAM
```

OPT - OPTION DEFINITION. THE ASSEMBLER OUTPUT CAN BE CONTROLLED WITH THE ARGUMENTS OF OPT. NO LABEL MAY BE USED, AND THE OPTIONS ARE WRITTEN IN THE OPERAND FIELD SEPARATED BY COMMAS:-

LONG FORM	SHORT FORM	
NOOTAP	NOO	SUPPRESS LOAD FILE
NOLIST	NOL	SUPPRESS LISTING
NOSYMB	NOS	SUPPRESS SYMBOL TABLE LISTING

FOR EXAMPLE:-

```
    OPT   NOO,NOLIST,NOS      TO OUTPUT ERROR COUNT ONLY
```

ORG - ORIGIN DEFINITION. THE PROGRAMMER CAN DEFINE AND REDEFINE THE ABSOLUTE ADDRESS WHERE THE NEXT AND FOLLOWING ASSEMBLED INSTRUCTIONS ARE TO START. THE OPERAND MAY BE A NUMBER OR AN EXPRESSION, FOR EXAMPLE:-

```
ORG   $100
ORG    *+$100
```

THE FIRST LINE SETS THE ASSEMBLY LOCATION COUNTER TO 256, WHILE THE SECOND SETS IT 256 BYTES FURTHER ON.

PAGE - PAGE ADVANCE. THE LISTING IS ADVANCED TO THE TOP OF THE NEXT PAGE, AND PRINTS THE HEADING SPECIFIED IN THE NAM DIRECTIVE. THE PAGE DIRECTIVE ITSELF IS NOT LISTED.

RMB - RESERVE MEMORY BYTES. THIS DIRECTIVE RESERVES A BLOCK OF MEMORY, WITH SIZE GIVEN BY THE OPERAND, FOR EXAMPLE

```
RMB   64
```

RESERVES 64 BYTES. THE ACTION OF THE DIRECTIVE IS MERELY TO INCREASE THE LOCATION COUNTER AND THE CONTENTS OF THE MEMORY BLOCK ARE NOT CHANGED.

SPC - SPACE LINES. THIS DIRECTIVE PROVIDES THE NUMBER OF BLANK LINES IN THE LISTING AS DEFINED BY THE OPERAND. THE SPC LINE IS NOT LISTED.

APPENDIX VI

STEADY STATE BINARY COLUMN MODEL PROGRAMS

VI.1 SYSHDL

SYSHDL was written to handle the system property data for the binary column model program SSGW. A listing of the program, and a sample output of the system data for methanol/water (using the data of Appendix VII) follows.

```
FORTRAN IV      V02.1-1     T   06-N   -79 00:42:20                  PAGE 001
      C
      C                  ************
      C                  *  SYSHDL  *
      C                  ************
      C
      C        SYSTEM PROPERTY FILE HANDLER
      C
      C
      C
      C        PURPOSE: TO INPUT SYSTEM PROPERTY DATA
      C        AND CREATE DISK FILE FOR OTHER PROGRAMS.
      C
      C
      C        GRANT WILSON
      C        APRIL 1976
      C        PART OF COLSYM PACKAGE
      C
0001           DIMENSION EQS(41)
0002           DIMENSION LBL(25)
0003           DX=0.025
      C
0004           WRITE(7,100)
0005       100 FORMAT('0 REQUESTING SYSTEM PROPERTIES'/'$ DO YOU HAVE A DISK FILE
             1 AVAILABLE? ')
0006           READ(5,300) IQRE
0007       300 FORMAT(A2)
0008           IF(IQRE.NE.'YE') GO TO 10
0010           WRITE(7,105)
0011       105 FORMAT(/' INPUT FILE NAME THUS: AA:FILNAM.DAT '/)
0012           CALL ASSIGN(10,'AA:FILNAM.DAT',-1,'RDO')
0013           READ(10,330)LBL
0014           READ(10,310)CP1,CP2,CP3,CP4
0015           READ(10,310)EL0,EL1,EL2,EL3,EL4,EL5
0016           READ(10,310)EV0,EV1,EV2,Ev3,EV4,EV5
0017           READ(10,310)AMW1,AMW2
0018           READ(10,310)RH1,RH2,RH3,RH4,RH5,RH6
0019           READ(10,310) TS0,TS1,TS2,TS3,TS4,TS5
0020           READ(10,310) DHV
0021           READ(10,350) (EGS(J),J=1,41)
0022       310 FORMAT(6E12.5)
0023       350 FORMAT(9F7.4)
0024           GO TO 30
      C
      C        COMMENCE DATA INPUT FROM TERMINAL
      C
0025        10 CONTINUE
0026           WRITE(7,295)
0027       295 FORMAT('0 INPUT TITLE LINE'/)
0028           READ(5,330)LBL
0029       330 FORMAT(25A2)
0030           WRITE(7,210)
      C
0031           WRITE(7,110)
0032       110 FORMAT('0INPUT COEFFICIENTS FOR HEATCP-CP1,CP2,CP3,CP4'/)
0033           READ(5,320)CP1,CP2,CP3,CP4
0034       320 FORMAT(6F11.0)
0035           WRITE(7,115)
0036       115 FORMAT('0 INPUT COEFFICIENTS FOR EL-EL0,EL1,EL2,EL3,EL4,EL5'/)
```

```
FORTRAN IV      V02.1-1     T    06-N   -79 00:42:20                 PAGE 002
0037          READ(5,320)EL0,EL1,EL2,EL3,EL4,EL5
0038          WRITE(7,120)
0039      120 FORMAT('0 INPUT COEFFICIENTS FOR EV-EV0,EV1,EV2,EV3,EV4,EV5'/)
0040          READ(5,320)EV0,EV1,EV2,EV3,EV4,EV5
0041          WRITE(7,125)
0042      125 FORMAT('0 INPUT COEFFICIENTS FOR MW-AMW1,AMW2'/)
0043          READ(5,320)AMW1,AMW2
0044          WRITE(7,130)
0045      130 FORMAT('0 INPUT COEFFICIENTS FOR DENSITIES-RH1-RH6'/)
0046          READ(5,320)RH1,RH2,RH3,RH4,RH5,RH6
0047          WRITE(7,131)
0048      131 FORMAT(/' INPUT COEFFICIENTS FOR T=F(X),TS0-TS5'/)
0049          READ(5,320) TS0,TS1,TS2,TS3,TS4,TS5
0050          WRITE(7,137)
0051      137 FORMAT(/' INPUT HEAT OF VAPOURISATION FOR REBOILER STEAM'/)
0052          READ(5,320) DHV
0053          WRITE(7,134)
0054      134 FORMAT(/'$DO YOU WISH TO USE CST REL VOL (YES/NO)? ')
0055          READ(5,300) IQRE
0056          IF(IQRE.NE.'YE') GO TO 11
0058          WRITE(7,136)
0059      136 FORMAT(/' INPUT CST REL VOL '/)
0060          READ(5,320) ALFA
0061          DO 12 J=1,41
0062          XT=DX*(J-1)
0063          EQS(J)=ALFA*XT/(1.+(ALFA-1.)*XT)
0064       12 CONTINUE
0065          GO TO 20
0066       11 CONTINUE
0067          WRITE(7,132)
0068      132 FORMAT('0INPUT EQULIBRIUM DATA TABLE'/
             1' (41 Y VALUES FOR AN X SPACING OF 0.025)'/
             2' 10 POINTS TO A LINE IN 10F6.0'/)
0069          READ(5,355) (EQS(J),J=1,41)
0070      355 FORMAT(10F6.0)
        C
0071       20 CONTINUE
        C
        C     CREATE DISK FILE?
        C
0072          WRITE(7,135)
0073      135 FORMAT(/'$DO YOU WISH TO CREATE A DISK FILE FOR FUTURE USE? ')
0074          READ(5,300) IQRE
0075          IF(IQRE.NE.'YE') GO TO 30
0077          WRITE(7,140)
0078      140 FORMAT(/' ENTER DISK FILE NAME THUS AA:FILNAM.DAT '/)
0079          CALL ASSIGN(11,'AA:FILNAM.DAT',-1,'NEW')
0080          WRITE(11,330)LBL
0081          WRITE(11,310)CP1,CP2,CP3,CP4
0082          WRITE(11,310)EL0,EL1,EL2,EL3,EL4,EL5
0083          WRITE(11,310)EV0,EV1,EV2,EV3,EV4,EV5
0084          WRITE(11,310)AMW1,AMW2
0085          WRITE(11,310)RH1,RH2,RH3,RH4,RH5,RH6
0086          WRITE(11,310) TS0,TS1,TS2,TS3,TS4,TS5
0087          WRITE(11,310) DHV
0088          WRITE(11,350) (EQS(J),J=1,41)
0089          WRITE(7,145)
0090      145 FORMAT('0*** DISK FILE CREATED ***')
```

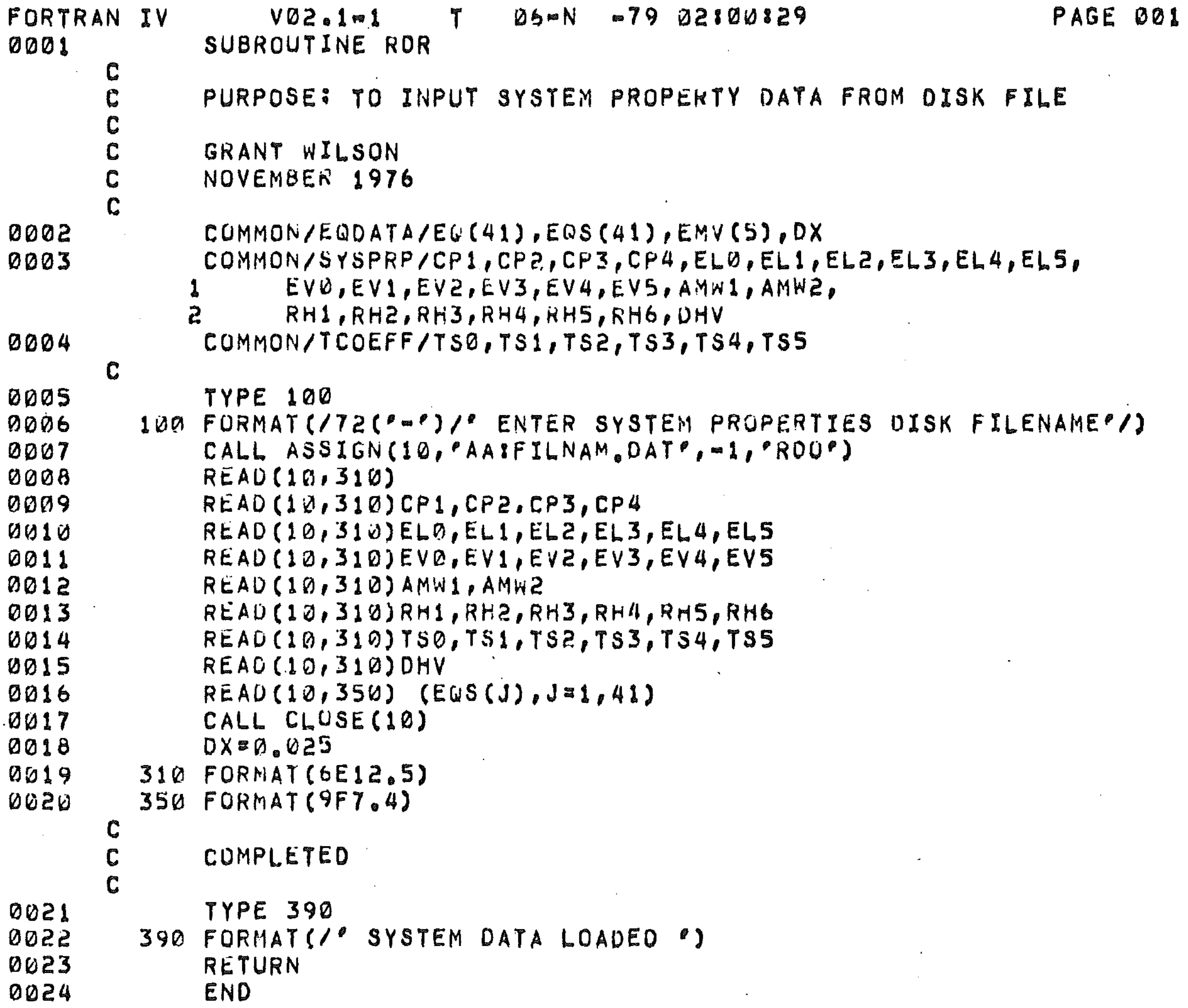
```
FORTRAN IV      V02.1-1     T    06-N   -79 00:42:20                  PAGE 003
0091          ENDFILE 11
        C
0092       30 CONTINUE
        C
        C     WRITE OUT SYSTEM PROPERTIES
        C
0093          WRITE(7,150)
0094      150 FORMAT(/'$DO YOU WISH TO LIST DATA ? ')
0095          READ(5,300) IQRE
0096          IF(IQRE.NE.'YE') STOP
0098          WRITE(7,152)
0099      152 FORMAT(/' ASSIGN OUTPUT FILE THUS AA:FILNAM.DAT/C'/)
0100          CALL ASSIGN(6,'AA:FILNAM.DAT/C',-1,'NEW')
0101          WRITE (6,200)
0102          WRITE(6,205)LBL
0103      205 FORMAT('0',25A2)
0104          WRITE(6,210)
0105          WRITE(6,220)
0106          WRITE(6,230)CP1,CP2,CP3,CP4
0107          WRITE(6,240)EL0,EL1,EL2,EL3,EL4,EL5
0108          WRITE(6,250)EV0,EV1,EV2,EV3,EV4,EV5
0109          WRITE(6,260)AMW1,AMW2
0110          WRITE(6,270)RH1,RH2,RH3,RH4,RH5,RH6
0111          WRITE(6,275) TS0,TS1,TS2,TS3,TS4,TS5
0112          WRITE(6,277) DHV
0113          WRITE(6,280)
0114          WRITE(6,290) (EQS(J),J=1,41)
0115      200 FORMAT(///,22X,'SYSTEM PROPERTIES'/22X,17('='))
0116      210 FORMAT('0HEATCP=(CP1T+CP2)X+CP3T+CP4',17X,'KJ/MOL K'/
             1'0',4X,'EL=((((EL5X+EL4)X+EL3)X+EL2)X+EL1)X+EL0 KJ/MOLE'/
             2'0',4X,'EV=((((EV5Y+EV4)Y+EV3)Y+EV2)Y+EV1)Y+EV0 KJ/MOLE'/
             3'0',4X,'MW=AMW1X+AMW2',27X,'MOLECULAR WEIGHT'/
             4'0',3X,'RHO=RH1+RH2T+(RH3+RH4T)X',17X,'GM/CC(SUBCOOLED)'/
             5'0',3X,'RHO=RH5+RH6X',29X,'GM/CC(SATURATED)'/
             6'0',5X,'T=((((TS5X+TS4)X+TS3)X+TS2)X+TS1)X+TS0 OC'/
             7'0'3X,'DHV=CONSTANT',29X,'KJ/KG')
0117      220 FORMAT('0',3(2X,'VARIABLE',5X,'VALUE',4X))
0118      230 FORMAT('0',4X,'CP1',5X,E12.5,4X,'CP2',5X,E12.5,4X,'CP3',5X,E12.5
             15X,'CP4',5X,E12.5)
0119      240 FORMAT('0',4X,'EL0',5X,E12.5,4X,'EL1',5X,E12.5,4X,'EL2',5X,E12.5
             15X,'EL3',5X,E12.5,4X,'EL4',5X,E12.5,4X,'EL5',5X,E12.5)
0120      250 FORMAT('0',4X,'EV0',5X,E12.5,4X,'EV1',5X,E12.5,4X,'EV2',5X,E12.5
             15X,'EV3',5X,E12.5,4X,'EV4',5X,E12.5,4X,'EV5',5X,E12.5)
0121      260 FORMAT('0',4X,'AMW1',4X,E12.5,4X,'AMW2',4X,E12.5)
0122      270 FORMAT('0',4X,'RH1',5X,E12.6,4X,'RH2',5X,E12.5,4X,'RH3',5X,E12.5
             15X,'RH4',5X,E12.5,4X,'RH5',5X,E12.5,4X,'RH6',5X,E12.5)
0123      275 FORMAT('0',4X,'TS0',5X,E12.5,4X,'TS1',5X,E12.5,4X,'TS2',5X,E12.5
             11X,4X,'TS3',5X,E12.5,4X,'TS4',5X,E12.5,4X,'TS5',5X,E12.5)
0124      277 FORMAT('0',4X,'DHV',5X,E12.5)
0125      280 FORMAT('0EQULIBRIUM Y VALUES FOR A SPACING OF 0.025 IN X ')
0126      290 FORMAT(3X,F6.4,5F10.4)
        C
        C     COMPLETE
        C
0127          CALL EXIT
0128          END
```

```
                                 SYSTEM PROPERTIES
                                 =================
METHANOL/WATER BINARY SYSTEM DATA
HEATCP=(CP1T+CP2)X+CP3T+CP4                             KJ/MOL K
    EL=((((EL5X+EL4)X+EL3)X+EL2)X+EL1)X+ELØ KJ/MOLE
    EV=((((EV5Y+EV4)Y+EV3)Y+EV2)Y+EV1)Y+EVØ KJ/MOLE
    MW=AMW1X+AMW2                                        MOLECULAR WEIGHT
   RHO=RH1+RH2T+(RH3+RH4T)X                              GM/CC(SUBCOOLED)
   RHO=RH5+RH6X                                          GM/CC(SATURATED)
     T=((((TS5X+TS4)X+TS3)X+TS2)X+TS1)X+TSØ OC
   DHV=CONSTANT                                          KJ/KG
  VARIABLE     VALUE         VARIABLE      VALUE         VARIABLE      VALUE
    CP1        Ø.67ØØØE-Ø7     CP2        Ø.126ØØE-Ø1      CP3        Ø.134ØØE-Ø4
    CP4        Ø.7467ØE-Ø1
    ELØ        Ø.7524ØE+Ø1     EL1       -Ø.1363ØE+Ø2      EL2        Ø.3393ØE+Ø2
    EL3       -Ø.3897ØE+Ø2     EL4        Ø.1647ØE+Ø2      EL5        Ø.ØØØØØE+ØØ
    EVØ        Ø.4821ØE+Ø2     EV1       -Ø.9Ø84ØE+Ø1      EV2        Ø.149ØØE+Ø1
    EV3        Ø.ØØØØØE+ØØ     EV4        Ø.ØØØØØE+ØØ      EV5        Ø.ØØØØØE+ØØ
    AMW1       Ø.14Ø24E+Ø2     AMW2       Ø.18Ø16E+Ø2
    RH1       Ø.1ØØ65ØE+Ø1     RH2       -Ø.36162E-Ø3      RH3       -Ø.19574E+ØØ
    RH4       -Ø.6Ø352E-Ø3     RH5        Ø.97ØØØE+ØØ      RH6       -Ø.2223ØE+ØØ
    TSØ        Ø.9993ØE+Ø2     TS1       -Ø.1627ØE+Ø3      TS2        Ø.5Ø19ØE+Ø3
    TS3       -Ø.8742ØE+Ø3     TS4        Ø.7398ØE+Ø3      TS5       -Ø.24ØØØE+Ø3
    DHV        Ø.225ØØE+Ø4
EQULIBRIUM Y VALUES FOR A SPACING OF Ø.Ø25 IN X
  Ø.ØØØØ     Ø.155Ø     Ø.267Ø     Ø.345Ø     Ø.418Ø     Ø.47ØØ
  Ø.517Ø     Ø.548Ø     Ø.579Ø     Ø.6Ø6Ø     Ø.627Ø     Ø.647Ø
  Ø.665Ø     Ø.682Ø     Ø.697Ø     Ø.712Ø     Ø.729Ø     Ø.738Ø
  Ø.75ØØ     Ø.764Ø     Ø.779Ø     Ø.788Ø     Ø.8ØØØ     Ø.813Ø
  Ø.825Ø     Ø.831Ø     Ø.846Ø     Ø.858Ø     Ø.87ØØ     Ø.882Ø
  Ø.893Ø     Ø.9Ø4Ø     Ø.915Ø     Ø.926Ø     Ø.936Ø     Ø.947Ø
  Ø.958Ø     Ø.967Ø     Ø.979Ø     Ø.99ØØ     1.ØØØØ
```

VI.2 SSGW

The model equations described in Chapter 6 were programmed and solved in an interactive program SSGW. A listing of this program and a sample output from the program follows.

```
FORTRAN IV      V02.1-1     T    06-N  -79 01:57:22                  PAGE 001
      C                                **********
      C                                *  SSGW  *
      C                                **********
      C
      C        A BINARY STEADY STATE DISTILLATION COLUMN PROGRAM
      C
      C
      C        GRANT WILSON
      C
      C
      C
0001           COMMON/DRMST /RMSTL,RMSTD
0002           COMMON/ELVDRV/EL(20),EV(20),DLEDT(20),DHDT(20),ELF,SUBCAL
0003           COMMON/EQDATA /EQ(41),EQS(41),EMV(5),DX
0004           COMMON/FLAG  /RFLAG,IFLAG
0005           COMMON/FLOCOM/L(20),V(20),X(20),Y(20),F,D,W,XF
0006           COMMON/HLBL  /LABEL(20),MSSG(8)
0007           COMMON/IDATA /TT,TB,TF,TR,FV,DV,RV,WV,QV1,QV2,PSI,R
0008           COMMON/RBRNCH/IOK,IYES,INO,IHT,LP(14)
0009           COMMON/NDATA/NP2,NP1,NFP,NF,NF2,ICOND,NEQN
0010           COMMON/REBOIL/QS1,QS2
0011           COMMON/SPRTN /TNSM,RM
0012           COMMON/SYSPRP/CP1,CP2,CP3,CP4,EL0,EL1,EL2,EL3,EL4,EL5,
              1     EV0,EV1,EV2,EV3,EV4,EV5,AMW1,AMW2,
              2     RH1,RH2,RH3,RH4,RH5,RH6,DHV
0013           COMMON/TCOEFF/ TS0,TS1,TS2,TS3,TS4,TS5
0014           COMMON/TRIDM / A(20),B(20),C(20),RHS(20)
0015           LOGICAL SUBCAL
0016           LOGICAL RFLAG,IFLAG
0017           REAL L
0018           DATA IOK,IYES,INO,IHT/'OK','YE','NO','HT'/
0019           CALL ASSIGN(6,'TT:')        !OUTPUT FILE
      C
0020           TYPE 500
0021       500 FORMAT(/'1',33X,'SSGW'/34X,4('*')/
              1/,14X,' A STEADY-STATE BINARY DISTILLATION PROGRAM'/
              2 15X,42('-'))
      C
      C        CALL RDR FOR SYSTEM PROPERTIES
      C
0022           CALL RDR
      C
0023        20 CONTINUE
      C
      C        USER OPTIONS
      C
0024           CALL READR1
      C
      C
      C        COMPUTE COLUMN VARIABLES
      C
0025           CALL SSDRIV
      C
0026           GO TO 20
      C
0027           END
```



```
FORTRAN IV      V02.1-1    T   06-N   -79 02:00:29                PAGE 001
0001          SUBROUTINE RDR
        C
        C     PURPOSE: TO INPUT SYSTEM PROPERTY DATA FROM DISK FILE
        C
        C     GRANT WILSON
        C     NOVEMBER 1976
        C
0002          COMMON/EQDATA/EQ(41),EQS(41),EMV(5),DX
0003          COMMON/SYSPRP/CP1,CP2,CP3,CP4,EL0,EL1,EL2,EL3,EL4,EL5,
             1     EV0,EV1,EV2,EV3,EV4,EV5,AMW1,AMW2,
             2     RH1,RH2,RH3,RH4,RH5,RH6,DHV
0004          COMMON/TCOEFF/TS0,TS1,TS2,TS3,TS4,TS5
        C
0005          TYPE 100
0006      100 FORMAT(/72('-')/' ENTER SYSTEM PROPERTIES DISK FILENAME'/)
0007          CALL ASSIGN(10,'AA:FILNAM.DAT',-1,'RDO')
0008          READ(10,310)
0009          READ(10,310)CP1,CP2,CP3,CP4
0010          READ(10,310)EL0,EL1,EL2,EL3,EL4,EL5
0011          READ(10,310)EV0,EV1,EV2,EV3,EV4,EV5
0012          READ(10,310)AMW1,AMW2
0013          READ(10,310)RH1,RH2,RH3,RH4,RH5,RH6
0014          READ(10,310)TS0,TS1,TS2,TS3,TS4,TS5
0015          READ(10,310)DHV
0016          READ(10,350) (EQS(J),J=1,41)
0017          CALL CLOSE(10)
0018          DX=0.025
0019      310 FORMAT(6E12.5)
0020      350 FORMAT(9F7.4)
        C
        C     COMPLETED
        C
0021          TYPE 390
0022      390 FORMAT(/' SYSTEM DATA LOADED ')
0023          RETURN
0024          END
```

```
FORTRAN IV      V02.1-1    T    06-N   -79 02:03:32                PAGE 001
0001            SUBROUTINE READR1
          C
          C     PURPOSE:TO INPUT INTERACTIVELY DATA FOR STEADY STATE SOLUTION
          C
          C     FOR USE BY INTERACTIVE TERMINAL
          C
          C     GRANT WILSON
          C     NOVEMBER 1975
          C
          C     PART OF COLSYM PACKAGE
          C
          C
0002            COMMON/DRMST/RMSTL,RMSTD
0003            COMMON/EQDATA/EQ(41),EQS(41),EMV(5),DX
0004            COMMON/FLOCOM/L(20),V(20),X(20),Y(20),F,D,W,XF
0005            COMMON/IDATA/TT,TB,TF,TR,FV,DV,RV,WV,QV1,QV2,PSI,R
0006            COMMON/NDATA/NP2,NP1,NFP,NF,NF2,ICOND,NEQN
0007            COMMON/REBOIL/QS1,QS2
0008            COMMON/RBRNCH/ IOK,IYES,INO,IHT,LP(14)
0009            COMMON/SYSPRP/CP1,CP2,CP3,CP4,EL0,EL1,EL2,EL3,EL4,EL5,
               1 EV0,EV1,EV2,EV3,EV4,EV5,AMW1,AMW2,
               2 RH1,RH2,RH3,RH4,RH5,RH6,DHV
0010            REAL L
          C
0011            TYPE 94
0012         94 FORMAT(/,1X,72('-'))
          C
0013         10 CONTINUE
          C
          C     ACCEPT INPUT FROM TERMINAL
          C
0014            TYPE 350
0015        350 FORMAT(/'$NEXT COMMAND ? ')
0016            READ(5,104) IQR
0017        104 FORMAT(A2)
0018            IF(IQR.EQ.'AL') GO TO 51
0020            IF(IQR.EQ.'PL') GO TO 51
0022            IF(IQR.EQ.'TE') GO TO 52
0024            IF(IQR.EQ.'FL') GO TO 53
0026            IF(IQR.EQ.'XF') GO TO 54
0028            IF(IQR.EQ.'RF') GO TO 55
0030            IF(IQR.EQ.'CL') GO TO 56
0032            IF(IQR.EQ.'RL') GO TO 57
0034            IF(IQR.EQ.'VE') GO TO 58
0036            IF(IQR.EQ.'LI') GO TO 59
0038            IF(IQR.EQ.'HL') GO TO 50
0040            IF(IQR.EQ.'OF') GO TO 61
0042            IF(IQR.EQ.'EX') CALL EXIT
0044            IF(IQR.EQ.'LP') GO TO 66      !LIST ON PRINTER
0046            IF(IQR.EQ.'LO') GO TO 1
0048            IF(IQR.EQ.'SA') GO TO 60
0050            IF(IQR.EQ.'RU') GO TO 70
          C
0052         50 TYPE 370
0053        370 FORMAT(/' OPTIONS ALLOWED ARE:'/
               1' AL - ENTER ALL DATA           PL - ENTER PLATE DATA'/
               2' TE - ENTER COLUMN TEMPS       FL - ENTER COLUMN FLOWS'/
               3' XF - ENTER FEED COMPOSITION   RF - REFLUX RATIO'/
```

```
FORTRAN IV      V02.1-1     T    06-N   -79 02:03:32           PAGE 002
           4' CL - CONVERGENCE LIMIT        RL - REBOILER LOAD'/
           5' VE - VAPOUR EFFICIENCY        LI - LIST ON SCREEN'/
           6' OF - ASSIGN OUTPUT FILE       EX - EXIT PROGRAM'/
           7' LP - LIST ON OUTPUT FILE      HL - THIS LISTING'//
           8' LOAD - LOAD DATA FROM DISK'/
           9' SAVE - SAVE DATA ON DISK'/
           *' RUN - RUN PROGRAM')
            GO TO 10
      C
          1 CONTINUE
            TYPE 415
        415 FORMAT(/' INPUT DISK FILE NAME'/)
            CALL ASSIGN(1,'AA:FILNAM.EXT',-1,'RDO')
            READ(1,100) NP
            READ(1,100) NF
            READ(1,120)TF,TR,TT,TB
            READ(1,120)FV,DV,WV
            READ(1,120)XF
            READ(1,120)R,PSI
            READ(1,120)RMSTL
            READ(1,120)QV1,QV2
            READ(1,120)(EMV(J),J=1,5)
            CALL CLOSE(1)
        120 FORMAT(6E13.6)
            NP2=NP+2
            NF2=NF+2
            NP1=NP+1
            NFP=NF+1
            ICOND=1
            NEQN=NP2
            TYPE 122
        122 FORMAT(' COLUMN DATA LOADED ')
            GO TO 10
      C
         51 TYPE 300
        300 FORMAT(/'$ENTER NUMBER OF COLUMN PLATES - ')
            READ(5,100) NP
        100 FORMAT(2I5)
            TYPE 302
        302 FORMAT(/'$ENTER FEED PLATE NUMBER - ')
            READ(5,100) NF
            ICOND=1
            NP2=NP+2
            NP1=NP+1
            NFP=NF+1
            NF2=NF+2
            NEQN=NP2
            IF(IQR.NE.'AL') GO TO 10
      C
         52 TYPE 304
        304 FORMAT(/' INPUT FEED,REFLUX,EST. TOPS,EST. BOTTOMS'/
           1'$TEMPERATURES IN OC - ')
            READ(5,110) TF,TR,TT,TB
        110 FORMAT(6F10.0)
            IF(IQR.NE.'AL') GO TO 10
      C
         53 TYPE 306
        306 FORMAT(/' INPUT FEED RATE,AND EST.DISTILLATE AND BOTTOMS'/
            1'$RATES IN L/TIME = ')
0102          READ(5,110) FV,DV,WV
0103          IF(IQR.NE.'AL') GO TO 10
      C
0105       54 TYPE 307
0106      307 FORMAT(/'$INPUT FEED COMPOSITION,CONDITION = ')
0107          READ(5,110) XF
0108          IF(IQR.NE.'AL') GO TO 10
      C
0110       55 TYPE 308
0111      308 FORMAT(/'$INPUT REFLUX RATIO = ')
0112          READ(5,110) R,PSI
0113          IF(IQR.NE.'AL') GO TO 10
      C
0115       56 TYPE 312
0116      312 FORMAT(/'$INPUT CONVERGENCE LIMIT RMSTL = ')
0117          READ(5,110) RMSTL
0118          IF(IQR.NE.'AL') GO TO 10
      C
0120       57 TYPE 314
0121      314 FORMAT(/' INPUT REBOILER COEFFICIENTS,QS=QV1+QV2*T'/
            1'$QS IN KJ/TIME = ')
0122          READ(5,110) QV1,QV2
0123          IF(IQR.NE.'AL') GO TO 10
      C
0125       58 TYPE 318
0126      318 FORMAT(/' INPUT COEFFICIENTS EMV(I),I=1,5 FOR FOURTH'/
            1'$ORDER POLYNOMIAL IN EMV = ')
0127          READ(5,110) (EMV(J),J=1,5)
0128          IF(IQR.NE.'AL') GO TO 10
0130          GO TO 20
      C
0131       61 TYPE 319
0132      319 FORMAT(/' REASSIGN OUTPUT FILE'/)
0133          REWIND 6
0134          CALL CLOSE(6)
0135          CALL ASSIGN(6,'AA:FILNAM.EXT/C',-1,'NEW')
0136          IF(IQR.NE.'AL') GO TO 10
      C
      C       OUTPUT ALL DATA RECEIVED
      C
0138       20 CONTINUE
0139       59 WRITE(7,200)
0140      200 FORMAT('1INPUT DATA FOR COLUMN (CONDENSER TYPE =TOTAL)')
0141          WRITE(7,210)NP,NF
0142      210 FORMAT(/' NUMBER OF PLATES  ',I3/' FEED PLATE NUMBER ',I3)
0143          WRITE(7,220)FV,XF,R,TF,TR
0144      220 FORMAT(/' STEADY STATE DATA:'/' FEED RATE        ',F11.5/
            1' FEED COMP.       ',F11.5/
            2' REFLUX RATIO     ',F11.5/' FEED TEMPERATURE',F11.5/
            3' REFLUX TEMP      ',F11.5)
0145          WRITE(7,230)DV,WV,TT,TB
0146      230 FORMAT(/' STEADY STATE GUESSES:'/' DISTILLATE RATE ',F11.5/
            1' BOTTOMS RATE     ',F11.5/' TOP TEMPERATURE ',F11.5/
            2' BOT TEMPERATURE ',F11.5)
0147          WRITE(7,240)QV1,QV2
0148      240 FORMAT(/' REBOILER DUTY:QS=',F10.4,' +T(NP2)* ',F8.4)
0149          WRITE(7,280) (EMV(J),J=1,5)
```

```
FORTRAN IV      V02.1-1    T   06-N  -79 02:03:32              PAGE 004
  280 FORMAT(/' MURPHREE TRAY EFFICIENCY COEFFICIENTS:'/
     11X,5E10.3)
      GO TO 10
C
C     NEW INPUT FILE TO BE CREATED
C
   60 TYPE 430
  430 FORMAT(/' INPUT NEW FILE NAME'/)
      CALL ASSIGN(1,'AA:FILNAM.EXT',-1,'NEW')
      WRITE(1,100) NP
      WRITE(1,100) NF
      WRITE(1,120)TF,TR,TT,TB
      WRITE(1,120)FV,DV,WV
      WRITE(1,120)XF
      WRITE(1,120)R,PSI
      WRITE(1,120)RMSTL
      WRITE(1,120)QV1,QV2
      WRITE(1,120)(EMV(J),J=1,5)
      CALL CLOSE(1)
      TYPE 124
  124 FORMAT(' NEW FILE CREATED')
      GO TO 10
C
C     RUN STEADY STATE SOLVER
C
C
C     CONVERT VOLUME,MASS FLOWS TO MOLAR FLOWS
C
   70 RHO=RH1+RH2*TF+(RH3+RH4*TF)*XF
      F=FV/(AMW1*XF+AMW2)*1000.0*RHO
      RHO=RH1+RH2*TR+(RH3+RH4*TR)*1.0
      D=DV/(AMW1*1.0+AMW2)*1000.0*RHO
      RHO=RH5+RH6*0.0
      W=WV/(AMW1*0.0+AMW2)*1000.0*RHO
      QS1=QV1*DHV
      QS2=QV2*DHV
      DO 72 J=1,41
   72 EQ(J)=EQS(J)
C
      RETURN
C
C     LIST OUTPUT ON PRINTER
C
   66 CONTINUE
      WRITE(6,200)
      WRITE(6,210)NP,NF
      QS=0.0
      WRITE(6,220)FV,XF,R,TF,TR
      WRITE(6,230)DV,WV,TT,TB
      WRITE(6,240)QV1,QV2
      WRITE(6,280) (EMV(J),J=1,5)
      REWIND 6
      GO TO 10
      END
```

```
FORTRAN IV      V02.1-1     T    06-N   -79 02:10:35                 PAGE 001
0001          FUNCTION HEATCP(COMP,T)
          C
          C   PURPOSE: TO PROVIDE LIQUID HEAT CAPACITY GIVEN COMP,T
          C
          C   GRANT WILSON
          C   APRIL 1976
          C
          C   HEAT CAPACITIES CALCULATED AS FOLLOWS:
          C        HEATCP=MOLE FRACTION AVERAGE OF PURE
          C   LIQUID CP'S WHICH ARE LINEAR
          C   FUNCTIONS OF TEMPERATURE. (KJ/MOLE K)
          C
0002          COMMON/SYSPRP/CP1,CP2,CP3,CP4,EL0,EL1,EL2,EL3,EL4,EL5,
             1EV0,EV1,EV2,EV3,EV4,EV5,AMW1,AMW2,
             2RH1,RH2,RH3,RH4,RH5,RH6,DHV
          C
0003          HEATCP=(CP1*T+CP2)*COMP+CP3*T+CP4
          C
0004          RETURN
0005          END
```

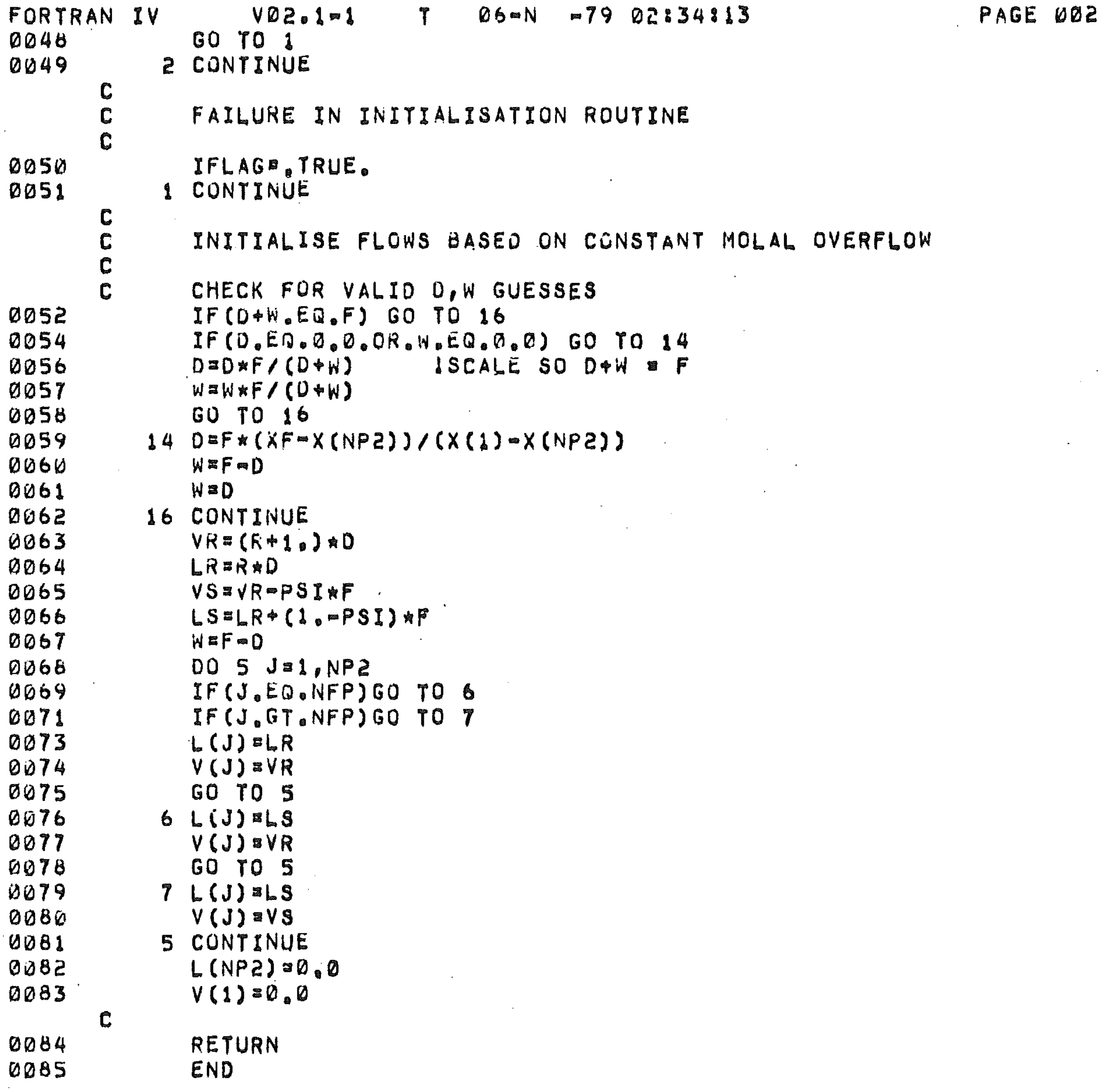
```
FORTRAN IV      VØ2.1-1    T    Ø6-N   -79 Ø2:1Ø:37                PAGE ØØ1
ØØØ1            SUBROUTINE NTHLPY
          C
          C     PURPOSE: TO CALCULATE LIQUID AND VAPOUR ENTHALPIES.
          C
          C     GRANT WILSON
          C     APRIL 1976
          C
          C     ENTHALPY=FIFTH ORDER POLYNOMIAL IN COMPOSITION
          C
          C
ØØØ2            COMMON/ELVDRV/EL(2Ø),EV(2Ø),DLEDT(2Ø),DHDT(2Ø),ELF,SUBCAL
ØØØ3            COMMON/FLOCOM/L(2Ø),V(2Ø),X(2Ø),Y(2Ø),F,D,W,XF
ØØØ4            COMMON/IDATA/TT,TB,TF,TR,FV,DV,RV,WV,QV1,QV2,PSI,R
ØØØ5            COMMON/NDATA/NP2,NP1,NFP,NF,NF2,ICOND,NEQN
ØØØ6            COMMON/SYSPRP/CP1,CP2,CP3,CP4,ELØ,EL1,EL2,EL3,EL4,EL5,
               1EVØ,EV1,EV2,EV3,EV4,EV5,AMW1,AMW2,
               2RH1,RH2,RH3,RH4,RH5,RH6,DHV
ØØØ7            COMMON/TCOEFF/TSØ,TS1,TS2,TS3,TS4,TS5
ØØØ8            LOGICAL SUBCAL
ØØØ9            REAL L
          C
ØØ1Ø            DO 1 J=1,NP2
ØØ11            EL(J)=((((EL5*X(J)+EL4)*X(J)+EL3)*X(J)+EL2)*X(J)+EL1)*X(J)+ELØ
ØØ12            EV(J)=((((EV5*Y(J)+EV4)*Y(J)+EV3)*Y(J)+EV2)*Y(J)+EV1)*Y(J)+EVØ
ØØ13          1 CONTINUE
          C
          C     CHECK FOR SUBCOOLING ON FEED AND REFLUX
          C
ØØ14            TBR=((((TS5*X(1)+TS4)*X(1)+TS3)*X(1)+TS2)*X(1)+TS1)*X(1)+TSØ
ØØ15            IF(TBR.LE.TR)GO TO 1Ø
ØØ17            EL(1)=EL(1)-HEATCP(X(1),(TBR+TR)/2.)*(TBR-TR)
ØØ18         1Ø CONTINUE
          C
ØØ19            ELF=((((EL5*XF+EL4)*XF+EL3)*XF+EL2)*XF+EL1)*XF+ELØ
ØØ2Ø            TBF=((((TS5*XF+TS4)*XF+TS3)*XF+TS2)*XF+TS1)*XF+TSØ
ØØ21            IF(TBF.LE.TF)GO TO 4Ø
ØØ23            ELF=ELF-HEATCP(XF,(TBF+TF)/2.)*(TBF-TF)
ØØ24         4Ø CONTINUE
          C
ØØ25            RETURN
ØØ26            END
```

```
FORTRAN IV      VØ2.1-1    T   Ø6-N   -79 Ø2:1Ø:43                PAGE ØØ1
ØØØ1          FUNCTION EQULIB(X,I)
      C
      C       PURPOSE TO PROVIDE EQUILIBRIUM DATA FOR PROGRAM
      C       USING LINEAR INTERPOLATION IN THE EQ TABLE AND
      C       INCORPORATING MURPHREE STAGE EFFICIENCYS(VAPOUR)
      C
      C       DX=SPACING OF X VALUES CORRESPONDING TO EQ TABLE
      C
      C       EQUILIBRIUM DATA ADJUSTED FOR NON-IDEAL STAGES BY
      C       TRANSFORMING THE EQ TABLE THUS
      C            Y'=X+EMV(Y-X)
      C       WHERE EMV IS A 4TH ORDER POLYNOMIAL IN X
      C
      C          IF I=Ø RETURN Y/X GIVEN X
      C          IF I=1 RETURN Y/X GIVEN Y
      C          IF I=2 RETURN X GIVEN Y
      C          IF I=3 RETURN Y GIVEN X
      C
      C       GRANT WILSON
      C       DECEMBER 1975
      C
ØØØ2          COMMON/EQDATA/EQ(41),EQS(41),EMV(5),DX
      C
ØØØ3          IF(I.EQ.1.OR.I.EQ.2) GO TO 999
      C
      C       FIND POSITION IN THE EQ TABLE
      C
ØØØ5          J=X/DX+1.
ØØØ6          IF(J.GT.4Ø) J=4Ø
ØØØ8          JP1=J+1
      C
      C       USE LINEAR INTERPOLATION BETWEEN J AND JP1
      C
ØØØ9          EQULIB=(X-DX*(J-1))*(EQ(JP1)-EQ(J))/DX+EQ(J)
ØØ1Ø          IF(I.EQ.Ø) EQULIB=EQULIB/X
ØØ12          RETURN
      C
ØØ13      999 CONTINUE
      C
      C       FIND THE X VALUE
      C
ØØ14          DO 1Ø J=2,41
ØØ15          IF(X.LT.EQ(J)) GO TO 16
ØØ17       1Ø CONTINUE
      C
      C       VALUE LIES BETWEEN J-1,J
      C       IE DX(J-2),DX(J-1)
      C       USE LINEAR INTERPOLATION
      C
ØØ18       16 X1=(J-2)*DX
ØØ19          EQULIB=(X-EQ(J-1))*DX/(EQ(J)-EQ(J-1))+X1
ØØ2Ø          IF(I.EQ.1) EQULIB=X/EQULIB
      C
ØØ22          RETURN
ØØ23          END
```

```
FORTRAN IV      VØ2.1-1    T    Ø6-N   -79 Ø2:33:58                 PAGE ØØ1
ØØØ1            SUBROUTINE SSDRIV
        C
        C       TO CONTROL THE SOLUTION OF THE STEADY STATE COLUMN.
        C
        C       GRANT WILSON
        C       AUGUST 1975
        C
        C
        C       SUBROUTINES REQUIRED: "FLOWS","TRID","THOMAS","RMST"
        C                             "INITL","EQULIB","HEATCP"
        C
ØØØ2            COMMON/DRMST/RMSTL,RMSTD
ØØØ3            COMMON/ELVDRV/EL(2Ø),EV(2Ø),DLEDT(2Ø),DHDT(2Ø),ELF,SUBCAL
ØØØ4            COMMON/FLAG/RFLAG,IFLAG
ØØØ5            COMMON/FLOCOM/L(2Ø),V(2Ø),X(2Ø),Y(2Ø),F,D,W,XF
ØØØ6            COMMON/IDATA/TT,TB,TF,TR,FV,DV,RV,WV,QV1,QV2,PSI,R
ØØØ7            COMMON/NDATA/NP2,NP1,NFP,NF,NF2,ICOND,NEQN
ØØØ8            COMMON/REBOIL/QS1,QS2
ØØØ9            COMMON/SPRTN/TNSM,RM
ØØ1Ø            COMMON/SYSPRP/CP1,CP2,CP3,CP4,ELØ,EL1,EL2,EL3,EL4,EL5,
               1 EVØ,EV1,EV2,EV3,EV4,EV5,AMW1,AMW2,
               2 RH1,RH2,RH3,RH4,RH5,RH6,DHV
ØØ11            COMMON/TCOEFF/TSØ,TS1,TS2,TS3,TS4,TS5
ØØ12            COMMON/TRIDM/A(2Ø),B(2Ø),C(2Ø),RHS(2Ø)
ØØ13            REAL L
ØØ14            LOGICAL RFLAG,IFLAG
ØØ15            RFLAG=.FALSE.
ØØ16            IFLAG=.FALSE.
        C
        C       SET UP INITIAL FLOWS AND COMPOSITIONS THEN
        C       CHECK FOR PROGRAM FAILURE IN "INITL"
        C
ØØ17            CALL INITL
ØØ18            IF(.NOT.IFLAG)GO TO 1
        C
        C       FAILURE IN "INITL" SUBROUTINE
        C
ØØ2Ø            WRITE(6,1ØØ)
ØØ21        1ØØ FORMAT(/' ',27('=')/' FAILURE IN SUBROUTINE INITL'/' ',27('='))
ØØ22            WRITE(6,11Ø)
ØØ23        11Ø FORMAT(/' INITIAL CONDITIONS WERE: TRAY COMPOSITION')
ØØ24            WRITE(6,12Ø)(J,X(J),J=1,NP2)
ØØ25        12Ø FORMAT(26X,I3,3X,F7.4)
ØØ26            STOP '? FAILURE IN INITIAL ?'
        C
ØØ27          1 CONTINUE
        C
        C
        C       ITERATE ON THE STEADY STATE SOLUTION
        C
ØØ28            DO 2 K=1,8ØØ
ØØ29            IF(K/5Ø*5Ø.EQ.K)CALL CHKSPR
ØØ31            CALL TRID
ØØ32            CALL THOMAS
ØØ33            CALL RMST
ØØ34            CALL FLOWS
        C       SENSE SWITCH 1 LISTS CONVERGENCE CRITERIA ON CONSOLE
ØØ35            IF(ISSW1(MM).LT.1) TYPE 23Ø,K,RMSTD,RMSTL
```

```
FORTRAN IV      V02.1-1    T   06-N  -79 02:33:58                PAGE 002
0037      230 FORMAT(' ITERATION=',I3,' RMSTO (RMSTL)= ',G12.4,'( ',G12.4,' )':
      C       SENSE SWITCH 2 LISTS XO,XW,D,W ON CURRENT ITERATION
0038          IF(ISSW2(MM).LT.1) TYPE 232,K,X(1),X(NP2),V(1),L(NP2)
0040      232 FORMAT(' ITERATION=',I3,' XO,XW,D,W= ',4G12.4)
      C       CONSOLE INTERRUPT FOR PROGRAM ABORT ?
0041          CALL BRKPT(M)
0042          IF(M.EQ.1) RFLAG=.TRUE.     !ABORT ON CTRL/A FROM TERMINAL
      C
      C       IF RFLAG IS SET SOLUTION HAS CONVERGED.
      C
0044          IF(RFLAG)GO TO 3
0046        2 CONTINUE
      C
      C       HAS NOT CONVERGED INSIDE 800 ITERATIONS.
      C
0047          WRITE(6,150)
0048      150 FORMAT(/' ',33('=')/' EXCEEDED 800 ITERATIONS IN SSDRIV'/' '33('=
         1))
0049          WRITE(6,190)
0050      190 FORMAT(/' LAST RESULT',12X,': TRAY COMPOSITION')
0051          WRITE(6,120) (J,X(J),J=1,NP2)
0052          WRITE(7,160) K,RMSTD,RMSTL
0053          RETURN
      C
0054        3 CONTINUE
0055          CALL RCTRLO
      C
      C
      C       STEADY STATE SOLUTION FOUND
      C
      C       COMPUTE NECESSARY PARAMETERS
      C
0056          DO 10 J=1,NP2
0057       10 Y(J)=EQULIB(X(J),3)
      C
0058          QC=V(2)*(EV(2)-EL(1))
0059          Z=X(NP2)
0060          TC=(((((TS5*Z+TS4)*Z+TS3)*Z+TS2)*Z+TS1)*Z+TS0
0061          QS=QS1+QS2*TC
0062          CALL RMNM
      C
      C       CALCULATE VOLUME FLOWS
      C
0063          RHO=RH1+RH2*TR+(RH3+RH4*TR)*X(1)
0064          AMW=AMW1*X(1)+AMW2
0065          DV=D*AMW/1000.0/RHO
0066          RV=L(1)*AMW/1000.0/RHO
0067          RHO=RH5+RH6*X(NP2)
0068          WV=W*(AMW1*X(NP2)+AMW2)/1000.0/RHO
0069          BALV=FV-DV-WV
0070          BAL=F-D-W
      C
      C       CALCULATE END TEMPERATURES
      C
0071          Z=X(1)
0072          TT=(((((TS5*Z+TS4)*Z+TS3)*Z+TS2)*Z+TS1)*Z+TS0
0073          Z=X(NP2)
0074          TB=(((((TS5*Z+TS4)*Z+TS3)*Z+TS2)*Z+TS1)*Z+TS0
FORTRAN IV       VØ2.1-1     T    Ø6-N  -79 Ø2:33:58                    PAGE ØØ3
      C
      C       CALCULATE INTERNAL REFLUX RATIO
      C
ØØ75          Z=X(1)
ØØ76          TRSATD=(((((TS5*Z+TS4)*Z+TS3)*Z+TS2)*Z+TS1)*Z+TSØ
ØØ77          CP=HEATCP(Z,(TR+TRSATD)/2.)
ØØ78          HVD=(((((EV5*Z+EV4)*Z+EV3)*Z+EV2)*Z+EV1)*Z+EVØ
ØØ79          HLD=(((((EL5*Z+EL4)*Z+EL3)*Z+EL2)*Z+EL1)*Z+ELØ
ØØ8Ø          RINT=R*(1.+CP*(TRSATD-TR)/(HVD-HLD))
      C
      C       SELECT OUTPUT FORMAT USING CONSOLE SWITCH 1
      C       ON = SHORT LISTING , OFF = LONG LISTING
      C
ØØ81          IF(ISSWØ(MM).LT.1) GO TO 9ØØ
ØØ83          WRITE(6,1ØØØ) F,QS,TF,TR,XF,X(1),X(NP2),V(1),L(1),V(2)
ØØ84          WRITE(6,1Ø1Ø) W,L(NP1),V(NP2),R,RINT
ØØ85     1ØØØ FORMAT(/' F =',E13.6,2X,'QS=',E13.6,2X,'TF=',E13.6,2X,
             1'TR=',E13.6/
             2' XF=',E13.6,2X,'XD=',E13.6,2X,'XW=',E13.6,2X/
             3' D =',E13.6,2X,'LR=',E13.6,2X,'VR=',E13.6,2X)
ØØ86     1Ø1Ø FORMAT(' W =',E13.6,2X,'LS=',E13.6,2X,'VS=',E13.6,2X/
             1' RE=',E13.6,2X,'RI=',E13.6,2X)
ØØ87          REWIND 6
ØØ88          RETURN
      C
      C       WRITE OUT RESULTS
      C
ØØ89      9ØØ WRITE(6,13Ø)
ØØ9Ø      13Ø FORMAT(/' STEADY STATE SOLUTION')
ØØ91          WRITE(6,17Ø)
ØØ92      17Ø FORMAT(/' TRAY',4X,'LIQUID',8X,'VAPOUR',8X,'LIQUID',8X,
            1        'VAPOUR'/9X,'COMP.',9X,'COMP.',9X,'RATE',1ØX,'RATE')
ØØ93          WRITE(6,14Ø)(J-1,X(J),Y(J),L(J),V(J),J=1,NP2)
ØØ94      14Ø FORMAT(2X,I2,2X,E13.6,1X,E13.6,1X,E13.6,1X,E13.6)
      C
ØØ95          WRITE(6,16Ø) K,RMSTD,RMSTL
ØØ96      16Ø FORMAT(/' AFTER ',I3,' ITERATIONS,RMSTD= ',G11.3,' (WITH RMSTL= '
             1G11.3,')')
ØØ97          WRITE(6,18Ø) QS,QC
ØØ98      18Ø FORMAT(/' ENERGY TRANSFER,KJ/TIME:',
             1' REBOILER DUTY  ',E13.6/
             226X,'CONDENSER DUTY ',E13.6)
      C
ØØ99          WRITE(6,2ØØ)TNSM,RM,PSI
Ø1ØØ      2ØØ FORMAT('ØMINIMUM NUMBER OF STAGES ',F6.3,
             1       ' MINIMUM REFLUX RATIO     ',F6.3/
             2       ' FEED FRAC VAPOURISED     ',F6.3)
Ø1Ø1          WRITE(6,21Ø) FV,DV,RV,WV,BAL,BALV
Ø1Ø2      21Ø FORMAT(/' VOLUMETRIC FLOWS,L/TIME:',
             1'   FEED',7X,F6.2/28X,'DISTILLATE ',F6.2/
             228X,'REFLUX',5X,F6.2/28X,'BOTTOMS',4X,F6.2/
             3' MOLAR BALANCE DISCREPANCY',F8.4/
             4' VOLUME BALANCE DIFFERENCE',F8.4)
Ø1Ø3          WRITE(6,215) TT,TB
Ø1Ø4      215 FORMAT(/' COLUMN END TEMPERATURES:',
             1' TOPS        ',F6.2,' BOTTOMS    ',F6.2)
Ø1Ø5          WRITE(6,217) R,RINT
Ø1Ø6      217 FORMAT(/' REFLUX RATIOS:  EXTERNAL = ',F6.3,
FORTRAN IV      V02.1-1    T   06-N  -79 02:33:58                PAGE 004
              1'  INTERNAL = ',F6.3)
0107           REWIND 6
0108           RETURN
0109           END
```

```
FORTRAN IV      V02.1-1    T    06-N   -79 02:34:13                  PAGE 001
0001            SUBROUTINE INITL
          C
          C     PURPOSE:TO INITIALISE COMPOSITIONS AND FLOWS FOR
          C             THE STEADY STATE SOLUTION
          C
          C     GRANT WILSON
          C     AUGUST 1975
          C
          C
0002            COMMON/EQDATA/EQ(41),EQS(41),EMV(5),DX
0003            COMMON/FLAG/RFLAG,IFLAG
0004            COMMON/FLOCOM/L(20),V(20),X(20),Y(20),F,D,W,XF
0005            COMMON/IDATA/TT,TB,TF,TR,FV,DV,RV,WV,QV1,QV2,PSI,R
0006            COMMON/NDATA/NP2,NP1,NFP,NF,NF2,ICOND,NEQN
0007            COMMON/TCOEFF/ TS0,TS1,TS2,TS3,TS4,TS5
0008            REAL L
0009            REAL LR,LS
0010            LOGICAL RFLAG,IFLAG
          C
          C     CHECK INITIAL TEMP GUESSES WHERE VALID FOR TT,TB
          C
0011            TOP=TS5+TS4+TS3+TS2+TS1+TS0
0012            BOT=TS0
0013            DT=BOT-TOP
0014            TOP=TOP+0.05*DT
0015            BOT=BOT-0.05*DT
0016            IF(TT.LT.TOP) TT=TOP
0018            IF(TB.GT.BOT) TB=BOT
0020            IF(TB.GT.TT) GO TO 10
0022            TT=TOP
0023            TB=BOT
0024         10 CONTINUE
          C
          C     SET UP LINEAR TEMPERATURE PROFILES
          C
0025            DO 1 J=1,NP2
0026            TI=TT+(J-1)*(TB-TT)/NP1
          C
          C     COMMENCE WEGSTEIN SEARCH FOR X(J),MAX OF 50 ITERATIONS
          C
0027            X0=1.-1./NP1*(J-1)
0028            X1=(TI-TS0-(((TS5*X0+TS4)*X0+TS3)*X0+TS2)*X0*X0)/TS1
0029            Y0=X1
0030            IJ=0
0031         30 CONTINUE
0032            Y1=(TI-TS0-(((TS5*X1+TS4)*X1+TS3)*X1+TS2)*X1*X1)/TS1
0033            DENOM=X1-X0+Y0-Y1
0034            IF(DENOM.EQ.0.0) GO TO 2
0036            X2=(X1*Y0-X0*Y1)/DENOM
0037            IF(ABS(X2-X1).LT.0.0001) GO TO 20
0039            X0=X1
0040            Y0=Y1
0041            X1=X2
0042            IJ=IJ+1
0043            IF(IJ.GT.50) GO TO 2
0045            GO TO 30
0046         20 CONTINUE
0047            X(J)=X2
```



```
FORTRAN IV      VØ2.1-1     T    Ø6-N   -79 Ø2:34:13                  PAGE ØØ2
ØØ48           GO TO 1
ØØ49         2 CONTINUE
       C
       C       FAILURE IN INITIALISATION ROUTINE
       C
ØØ5Ø           IFLAG=.TRUE.
ØØ51         1 CONTINUE
       C
       C       INITIALISE FLOWS BASED ON CONSTANT MOLAL OVERFLOW
       C
       C       CHECK FOR VALID D,W GUESSES
ØØ52           IF(D+W.EQ.F) GO TO 16
ØØ54           IF(D.EQ.Ø.Ø.OR.W.EQ.Ø.Ø) GO TO 14
ØØ56           D=D*F/(D+W)      !SCALE SO D+W = F
ØØ57           W=W*F/(D+W)
ØØ58           GO TO 16
ØØ59        14 D=F*(XF-X(NP2))/(X(1)-X(NP2))
ØØ6Ø           W=F-D
ØØ61           W=D
ØØ62        16 CONTINUE
ØØ63           VR=(R+1.)*D
ØØ64           LR=R*D
ØØ65           VS=VR-PSI*F
ØØ66           LS=LR+(1.-PSI)*F
ØØ67           W=F-D
ØØ68           DO 5 J=1,NP2
ØØ69           IF(J.EQ.NFP)GO TO 6
ØØ71           IF(J.GT.NFP)GO TO 7
ØØ73           L(J)=LR
ØØ74           V(J)=VR
ØØ75           GO TO 5
ØØ76         6 L(J)=LS
ØØ77           V(J)=VR
ØØ78           GO TO 5
ØØ79         7 L(J)=LS
ØØ8Ø           V(J)=VS
ØØ81         5 CONTINUE
ØØ82           L(NP2)=Ø.Ø
ØØ83           V(1)=Ø.Ø
       C
ØØ84           RETURN
ØØ85           END
```

```
FORTRAN IV      V02.1-1     T    06-N   -79 02:34:23                  PAGE 001
      SUCROUTINE TRID
C
C     SUBROUTINE TO SET UP THE TRIDIAGONAL MATRIX FOR THE
C     STEADY STATE SOLUTION
C
C     GRANT WILSON
C     AUGUST 1975
C
C
      COMMON/EQDATA/EQ(41),EQS(41),EMV(5),DX
      COMMON/FLOCOM/L(20),V(20),X(20),Y(20),F,D,W,XF
      COMMON/NDATA/NP2,NP1,NFP,NF,NF2,ICOND,NEQN
      COMMON/TRIDM/A(20),B(20),C(20),RHS(20)
      REAL L
C
C     SET UP MATRIX
C
C     CONDENSER
C
      A(1)=0.0
      B(1)=-(L(1)+D)
      C(1)=V(2)*EQULIB(X(2),0)
      RHS(1)=0.0
C
C     ENRICHING SECTION
C
      DO 200 J=2,NF
      A(J)=L(J-1)
      B(J)=-(V(J)*EQULIB(X(J),0)+L(J))
      C(J)=V(J+1)*EQULIB(X(J+1),0)
  200 RHS(J)=0.0
C
C     FEED PLATE
C
      A(NFP)=L(NF)
      B(NFP)=-(V(NFP)*EQULIB(X(NFP),0)+L(NFP))
      C(NFP)=V(NFP+1)*EQULIB(X(NFP+1),0)
      RHS(NFP)=-F*XF
C
C     STRIPPING SECTION
C
      DO 210 J=NF2,NP1
      A(J)=L(J-1)
      B(J)=-(V(J)*EQULIB(X(J),0)+L(J))
      C(J)=V(J+1)*EQULIB(X(J+1),0)
  210 RHS(J)=0.0
C
C     REBOILER
C
      A(NP2)=L(NP1)
      B(NP2)=-(V(NP2)*EQULIB(X(NP2),0)+W)
      C(NP2)=0.0
      RHS(NP2)=0.0
C
      RETURN
      END
```

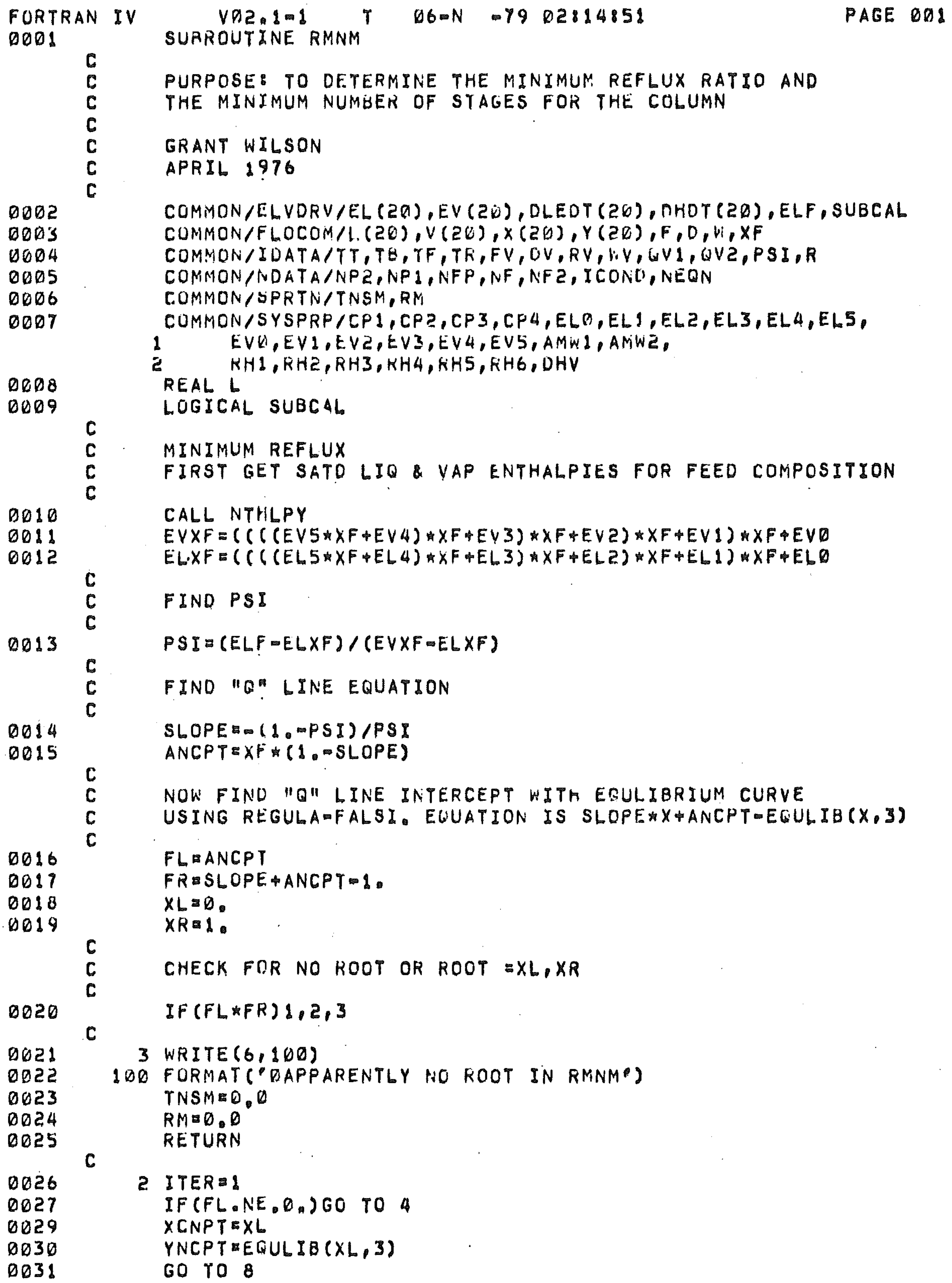
```
FORTRAN IV      V02.1-1     T    06-N   -79 02:34:29              PAGE 001
          SUBROUTINE THOMAS
C
C         PURPOSE: TO SOLVE THE STEADY STATE MATRIX (TRIDIAGONAL)
C
C         GRANT WILSON
C         AUGUST 1975
C
C
          COMMON/NDATA/NP2,NP1,NFP,NF,NF2,ICOND,NEQN
          COMMON/TRIDM/A(20),B(20),C(20),RHS(20)
C
          C(1)=C(1)/B(1)
          DO 200 IJ=2,NP2
          B(IJ)=B(IJ)-A(IJ)*C(IJ-1)
          C(IJ)=C(IJ)/B(IJ)
      200 CONTINUE
C
          A(1)=RHS(1)/B(1)
          DO 201 IJ=2,NP2
          A(IJ)=(RHS(IJ)-A(IJ)*A(IJ-1))/B(IJ)
      201 CONTINUE
C
          RHS(NP2)=A(NP2)
          DO 202 IJ=2,NP2
          I=NP2+1-IJ
          RHS(I)=A(I)-C(I)*RHS(I+1)
      202 CONTINUE
C
          RETURN
          END
```

```
FORTRAN IV      V02.1-1     T    06-N   -79 02:34:33                  PAGE 001
0001          SUBROUTINE RMST
        C
        C     PURPOSE:TO CHECK ON THE STEADY STATE SOLUTION CONVERGENCE
        C
        C     GRANT WILSON
        C     AUGUST 1975
        C
        C
0002          COMMON/DRMST/RMSTL,RMSTD
0003          COMMON/FLAG/RFLAG,IFLAG
0004          COMMON/FLOCOM/L(20),V(20),X(20),Y(20),F,D,W,XF
0005          COMMON/NDATA/NP2,NP1,NFP,NF,NF2,ICOND,NEQN
0006          COMMON/TCOEFF/ TS0,TS1,TS2,TS3,TS4,TS5
0007          COMMON/TRIDM/A(20),B(20),C(20),RHS(20)
0008          REAL L
0009          LOGICAL RFLAG,IFLAG
0010          SE=0.0
        C
        C     FIND BUBBLE POINTS ON PRESENT AND PAST ITERATIONS
        C     SE=SQRT(SE)/NP2/AV.TEMP.*100
        C
0011          DO 1 J=1,NP2
0012          Z=RHS(J)
0013          T1=(((((TS5*Z+TS4)*Z+TS3)*Z+TS2)*Z+TS1)*Z+TS0
0014          U=X(J)
0015          T2=(((((TS5*U+TS4)*U+TS3)*U+TS2)*U+TS1)*U+TS0
0016          X(J)=RHS(J)
0017          SE=SE+(T1-T2)*(T1-T2)
0018          IF(J.EQ.1)TT1=T1
0020        1 IF(J.EQ.NP2)TT2=T1
0022          RMSTD=SQRT(SE)/NP2*200./(TT1+TT2)
        C
        C     TEST FOR CONVERGENCE
        C
0023          IF(RMSTD.LE.RMSTL)RFLAG=.TRUE.
        C
0025          RETURN
0026          END
```

```
FORTRAN IV      V02.1-1    T    06-N   -79 02:34:37              PAGE 001
      SUBROUTINE FLOWS
C
C     PURPOSE: TO RECALCULATE THE COLUMN FLOWS FOR THE STEADY
C              SOLUTION.
C
C     GRANT WILSON
C     OCTOBER 1975
C
C
      COMMON/ELVDRV/EL(20),EV(20),DLEDT(20),DHDT(20),ELF,SUBCAL
      COMMON/EQDATA/EQ(41),EQS(41),EMV(5),DX
      COMMON/FLOCOM/L(20),V(20),X(20),Y(20),F,D,W,XF
      COMMON/IDATA/TT,TB,TF,TR,FV,DV,RV,HV,QV1,QV2,PSI,R
      COMMON/NDATA/NP2,NP1,NFP,NF,NF2,ICOND,NEQN
      COMMON/REBOIL/QS1,QS2
      COMMON/TCOEFF/ TS0,TS1,TS2,TS3,TS4,TS5
      REAL L
      LOGICAL SUBCAL
C
C     CALCULATE NEW Y VALUES
C
      Y(2)=X(1)
      DO 10 J=3,NP2
      Z=X(J)
      Y(J)=EQULIB(Z,3)
   10 CONTINUE
C
C
C     CALCULATE NEW ENTHALPIES
C
      CALL NTHLPY
C
C     CALCULATE QS,D,L(1),V(2),W
C
      Z=X(NP2)
      QS=QS1+(((((TS5*Z+TS4)*Z+TS3)*Z+TS2)*Z+TS1)*Z+TS0)*QS2
      DNUM=F*(ELF-EL(NP2))+QS
      DDEN=(R+1.)*(EV(2)-EL(1))+EL(1)-EL(NP2)
      D=DNUM/DDEN
      L(1)=R*D
      V(2)=(R+1.)*D
      V(1)=D
      W=F-D
      L(NP2)=W
C
C     ENRICHING SECTION
C
      DO 1 J=2,NF
      L(J)=(V(J)*EV(J)-L(J-1)*EL(J-1)-D*EV(J+1))/(EV(J+1)-EL(J))
      V(J+1)=D+L(J)
    1 CONTINUE
C
C     FEED PLATE
C
      L(NFP)=(V(NFP)*EV(NFP)-L(NF)*EL(NF)+W*EV(NF2)-F*ELF)/
     1      (EV(NF2)-EL(NFP))
      V(NF2)=L(NFP)-W
C
```

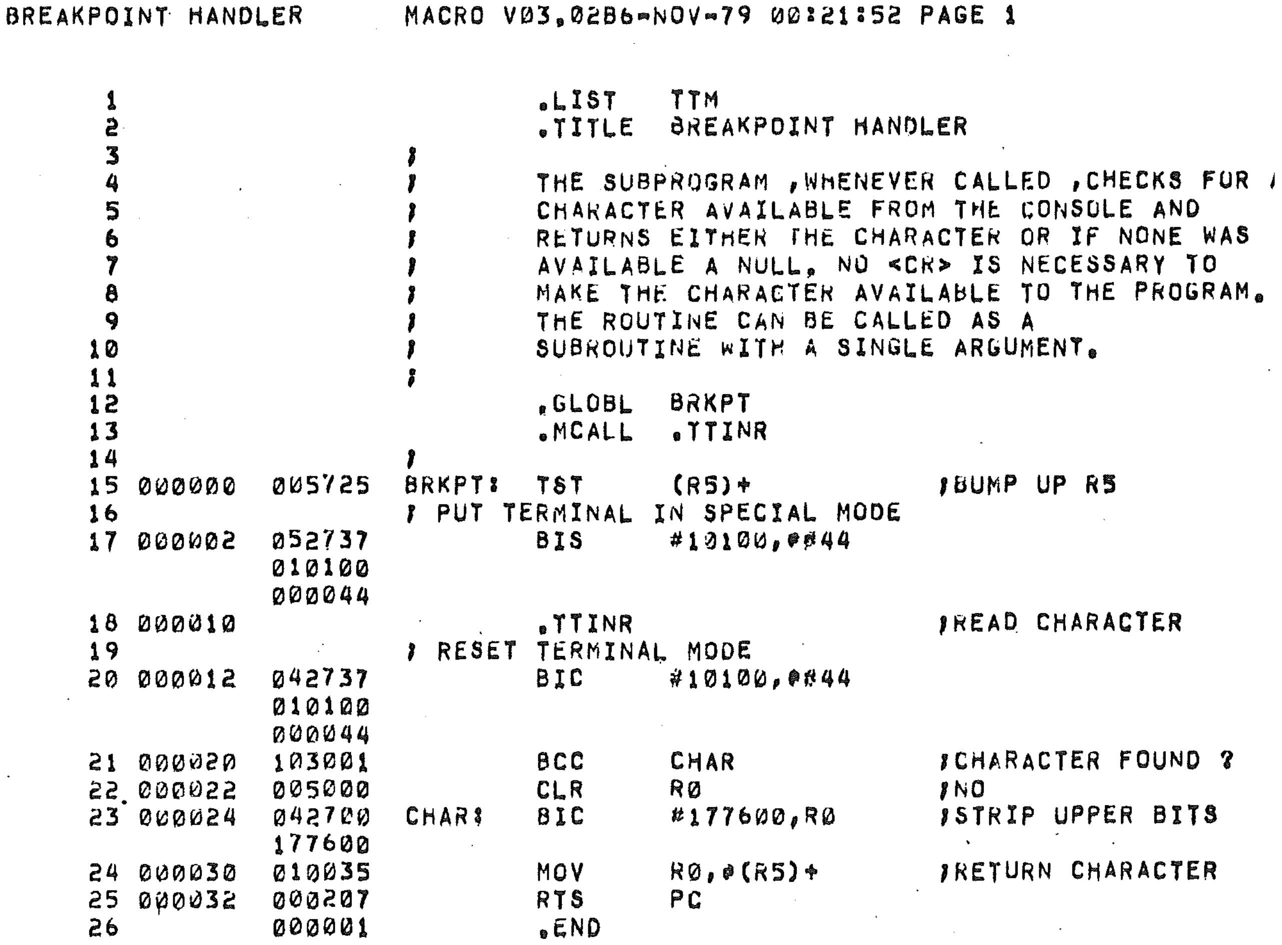
```
FORTRAN IV      V02.1-1    T    06-N   -79 02:34:37              PAGE 002
       C      STRIPPING SECTION
       C
0033          DO 2 J=NF2,NP1
0034          L(J)=(V(J)*EV(J)-L(J-1)*EL(J-1)+W*EV(J+1))/(EV(J+1)-EL(J))
0035          V(J+1)=L(J)-W
0036        2 CONTINUE
       C
       C      RECOMPUTE L/V EQUILIBRIA USING EMV CRITERIA
       C
       C      FIRST FIND AVERAGE FLOWS
       C
0037          ALR=0.
0038          AVR=0.0
0039          ALS=0.0
0040          AVS=0.0
0041          DO 20 J=1,NF
0042       20 ALR=ALR+L(J)
0043          ALR=ALR/NF
0044          DO 22 J=2,NF+1
0045       22 AVR=AVR+V(J)
0046          AVR=AVR/NF
0047          DO 24 J=NFP,NP1
0048       24 ALS=ALS+L(J)
0049          ALS=ALS/(NP1-NFP+1)
0050          DO 26 J=NF+2,NP2
0051       26 AVS=AVS+V(J)
0052          AVS=AVS/(NP1-NFP+1)
       C      NOW MODIFY EQUILIB DATA
0053          DO 30 J=1,41
0054          Z=(J-1)*DX
0055          EM=(((EMV(5)*Z+EMV(4))*Z+EMV(3))*Z+EMV(2))*Z+EMV(1)
0056          XINT=(D*X(1)/AVR-W*X(NP2)/AVS)/(ALS/AVS-ALR/AVR)
0057          IF(Z.GE.X(1)) GO TO 50
0059          IF(Z.GT.XINT) GO TO 48
0061          IF(Z.GT.X(NP2)) GO TO 46
       C      X <= XD
0063          EQ(J)=Z+EM*(EQS(J)-Z)
0064          GO TO 30
       C      XW < X <= XINT
0065       46 EQ(J)=EM*(EQS(J)-ALS/AVS*Z+W/AVS*X(NP2))+ALS/AVS*Z-W/AVS*X(NP2)
0066          GO TO 30
       C      XINT < X < XD
0067       48 EQ(J)=EM*(EQS(J)-ALR/AVR*Z-D/AVR*X(1))+ALR/AVR*Z+D/AVR*X(1)
0068          GO TO 30
       C      X >= XD
0069       50 EQ(J)=EM*(EQS(J)-Z)+Z
       C
0070       30 CONTINUE
       C
       C      COMPLETE
       C
0071          RETURN
0072          END
```

```
FORTRAN IV      V02.1-1    T   06-N   -79 02:14:48                PAGE 001
              SUBROUTINE CHKSPR
      C
      C       PURPOSE:TO CHECK THAT THE CONSTRAINTS OF
      C       MINIMUM REFLUX AND MINIMUM NUMBER
      C       OF STAGES ARE NOT VIOLATED.
      C
      C       GRANT WILSON
      C       APRIL 1976
      C
              COMMON/FLOCOM/L(20),V(20),X(20),Y(20),F,D,W,XF
              COMMON/IDATA/TT,TB,TF,TR,FV,DV,RV,WV,QV1,QV2,PSI,R
              COMMON/NDATA/NP2,NP1,NFP,NF,NF2,ICOND,NEQN
              COMMON/SPRTN/TNSM,RM
              REAL L
      C
      C       CHECK X VALUES
      C
              DO 10 J=1,NP2
              IF(X(J).LE.0.9999.AND.X(J).GE.0.0001) GO TO 10
              WRITE(6,100)J,X(J)
           10 CONTINUE
      C
      C       GET RM,NM
      C
              CALL RMNM
              IF(RM.NE.0.0) GO TO 20
              WRITE(6,110)
              STOP
      C
           20 CONTINUE
      C
      C       CHECK RM,TNSM FOR ACCEPTABLE VALUES
      C
              IF(RM.GE.R/1.02)WRITE(6,120)RM
              IF(TNSM.GE.NP1)WRITE(6,130)TNSM
      C
          100 FORMAT('0#### TRAY ',I2,' COMPOSITION IS ',E12.5,' ####')
          110 FORMAT('0#### ERROR IN RMNM,PROGRAM KILLED ####')
          120 FORMAT('0#### REFLUX RATIO TOO CLOSE TO MINIMUM REFLUX RATIO( ',
             1.3,' ) ####')
          130 FORMAT('0#### TOO FEW STAGES IN COLUMN,MINIMUM NUMBER OF STAGES
             1 ',F6.3,' ####')
      C
              RETURN
              END
```



```
FORTRAN IV      VØ2.1-1     T    Ø6-N   -79 Ø2:14:51             PAGE ØØ1
ØØØ1            SUBROUTINE RMNM
          C
          C     PURPOSE: TO DETERMINE THE MINIMUM REFLUX RATIO AND
          C     THE MINIMUM NUMBER OF STAGES FOR THE COLUMN
          C
          C     GRANT WILSON
          C     APRIL 1976
          C
ØØØ2            COMMON/ELVDRV/EL(2Ø),EV(2Ø),DLEDT(2Ø),DHDT(2Ø),ELF,SUBCAL
ØØØ3            COMMON/FLOCOM/L(2Ø),V(2Ø),X(2Ø),Y(2Ø),F,D,W,XF
ØØØ4            COMMON/IDATA/TT,TB,TF,TR,FV,DV,RV,WV,QV1,QV2,PSI,R
ØØØ5            COMMON/NDATA/NP2,NP1,NFP,NF,NF2,ICOND,NEQN
ØØØ6            COMMON/SPRTN/TNSM,RM
ØØØ7            COMMON/SYSPRP/CP1,CP2,CP3,CP4,ELØ,EL1,EL2,EL3,EL4,EL5,
               1      EVØ,EV1,EV2,EV3,EV4,EV5,AMW1,AMW2,
               2      RH1,RH2,RH3,RH4,RH5,RH6,DHV
ØØØ8            REAL L
ØØØ9            LOGICAL SUBCAL
          C
          C     MINIMUM REFLUX
          C     FIRST GET SATD LIQ & VAP ENTHALPIES FOR FEED COMPOSITION
          C
ØØ1Ø            CALL NTHLPY
ØØ11            EVXF=((((EV5*XF+EV4)*XF+EV3)*XF+EV2)*XF+EV1)*XF+EVØ
ØØ12            ELXF=((((EL5*XF+EL4)*XF+EL3)*XF+EL2)*XF+EL1)*XF+ELØ
          C
          C     FIND PSI
          C
ØØ13            PSI=(ELF-ELXF)/(EVXF-ELXF)
          C
          C     FIND "Q" LINE EQUATION
          C
ØØ14            SLOPE=-(1.-PSI)/PSI
ØØ15            ANCPT=XF*(1.-SLOPE)
          C
          C     NOW FIND "Q" LINE INTERCEPT WITH EQULIBRIUM CURVE
          C     USING REGULA-FALSI. EQUATION IS SLOPE*X+ANCPT-EQULIB(X,3)
          C
ØØ16            FL=ANCPT
ØØ17            FR=SLOPE+ANCPT-1.
ØØ18            XL=Ø.
ØØ19            XR=1.
          C
          C     CHECK FOR NO ROOT OR ROOT =XL,XR
          C
ØØ2Ø            IF(FL*FR)1,2,3
          C
ØØ21          3 WRITE(6,1ØØ)
ØØ22        1ØØ FORMAT('ØAPPARENTLY NO ROOT IN RMNM')
ØØ23            TNSM=Ø.Ø
ØØ24            RM=Ø.Ø
ØØ25            RETURN
          C
ØØ26          2 ITER=1
ØØ27            IF(FL.NE.Ø.)GO TO 4
ØØ29            XCNPT=XL
ØØ3Ø            YNCPT=EQULIB(XL,3)
ØØ31            GO TO 8
```

```
FORTRAN IV      V02.1-1     T    06-N  -79 02:14:51                  PAGE 002
0032        4 XNCPT=XR
0033          YNCPT=EQULIB(XR,3)
0034          GO TO 8
      C
      C       BEGIN ROOT SEARCH
      C
0035        1 CONTINUE
0036          DO 7 ITER=1,20
0037          XNCPT=(XL*FR-XR*FL)/(FR-FL)
0038          FNCPT=SLOPE*XNCPT+ANCPT-EQULIB(XNCPT,3)
0039          IF(ABS(FNCPT).LE.0.001)GO TO 8
0041          IF(FNCPT*FL.LT.0.)GO TO 6
0043          XL=XNCPT
0044          FL=FNCPT
0045          GO TO 7
0046        6 XR=XNCPT
0047          FR=FNCPT
0048        7 CONTINUE
      C
      C       FAILED TO CONVERGE
      C
0049          WRITE(6,110)
0050      110 FORMAT('0 CONVERGENCE FAILURE IN RMNM')
0051          TNSM=0.0
0052          RM=0.0
0053          RETURN
      C
0054        8 CONTINUE
0055          YNCPT=EQULIB(XNCPT,3)
      C
      C       SOLUTION FOUND, NOW CALCULATE RM
      C
0056          RM=(X(1)-YNCPT)/(YNCPT-XNCPT)
      C
      C       MINIMUM NUMBER OF STAGES
      C
0057          N=0
0058          Y1=X(1)
0059          X1=Y1
0060       10 N=N+1
0061          X2=X1
0062          X1=EQULIB(Y1,2)
0063          Y1=X1
0064          IF(X1.GE.X(NP2))GO TO 10
0066          DN=0.0
0067          IF((X2-X1).EQ.0.0) GO TO 989
0069          DN=(X2-X(NP2))/(X2-X1)
0070      989 CONTINUE
0071          TNSM=DN+N-1
      C
0072          RETURN
0073          END
```



```
BREAKPOINT HANDLER      MACRO V03.02B6-NOV-79 00:21:52 PAGE 1

     1                                  .LIST   TTM
     2                                  .TITLE  BREAKPOINT HANDLER
     3                          ;
     4                          ;       THE SUBPROGRAM ,WHENEVER CALLED ,CHECKS FOR
     5                          ;       CHARACTER AVAILABLE FROM THE CONSOLE AND
     6                          ;       RETURNS EITHER THE CHARACTER OR IF NONE WAS
     7                          ;       AVAILABLE A NULL. NO <CR> IS NECESSARY TO
     8                          ;       MAKE THE CHARACTER AVAILABLE TO THE PROGRAM.
     9                          ;       THE ROUTINE CAN BE CALLED AS A
    10                          ;       SUBROUTINE WITH A SINGLE ARGUMENT.
    11                          ;
    12                                  .GLOBL  BRKPT
    13                                  .MCALL  .TTINR
    14                          ;
    15 000000   005725          BRKPT:  TST     (R5)+           ;BUMP UP R5
    16                          ; PUT TERMINAL IN SPECIAL MODE
    17 000002   052737                  BIS     #10100,@#44
                010100
                000044
    18 000010                           .TTINR                  ;READ CHARACTER
    19                          ; RESET TERMINAL MODE
    20 000012   042737                  BIC     #10100,@#44
                010100
                000044
    21 000020   103001                  BCC     CHAR            ;CHARACTER FOUND ?
    22 000022   005000                  CLR     R0              ;NO
    23 000024   042700          CHAR:   BIC     #177600,R0      ;STRIP UPPER BITS
                177600
    24 000030   010035                  MOV     R0,@(R5)+       ;RETURN CHARACTER
    25 000032   000207                  RTS     PC
    26          000001                  .END
```

```
SENSESWITCH HANDLER     MACRO V03.02B6-NOV-79 00:29:24 PAGE 1

      1                                  .LIST   TTM
      2                                  .TITLE  SENSESWITCH HANDLER
      3                         ;
      4                         ;        A FORTRAN CALLABLE FUNCTION
      5                         ;        TO READ THE SIXTEEN CONSOLE SWITCHES
      6                         ;        AS SENSE SWITCHES,AND TO READ THE SWITCH
      7                         ;        REGISTER CONTENTS. RETURNS 1 FOR SWITCH
      8                         ;        ON (UP) AND 0 FOR SWITCH OFF (DOWN).
      9                         ;        ISSW0() = SWITCH 0,ISSW15() = SWITCH 15
     10                         ;        CALL OPTION(I) RETURNS SWITCH
     11                         ;        REGISTER CONTENTS IN "I".
     12                         ;
     13                                  .GLOBL  ISSW0,ISSW1,ISSW2,ISSW3,ISSW4,ISSW5
     14                                  .GLOBL  ISSW6,ISSW7,ISSW8,ISSW9,ISSW10
     15                                  .GLOBL  ISSW11,ISSW12,ISSW13,ISSW14,ISSW15
     16                                  .GLOBL  OPTION
     17                         ;
     18                 177570  SWREG=177570
     19                         ;
     20                         ;        UTILITY MACRO
     21                         ;
     22                                  .MACRO  PUSH    ARG
     23                                  MOV     ARG,-(SP)
     24                                  BR      TEST
     25                                  .ENDM
     26                         ;
     27 000000                  ISSW0:   PUSH    #1
     28 000006                  ISSW1:   PUSH    #2
     29 000014                  ISSW2:   PUSH    #4
     30 000022                  ISSW3:   PUSH    #10
     31 000030                  ISSW4:   PUSH    #20
     32 000036                  ISSW5:   PUSH    #40
     33 000044                  ISSW6:   PUSH    #100
     34 000052                  ISSW7:   PUSH    #200
     35 000060                  ISSW8:   PUSH    #400
     36 000066                  ISSW9:   PUSH    #1000
     37 000074                  ISSW10:  PUSH    #2000
     38 000102                  ISSW11:  PUSH    #4000
     39 000110                  ISSW12:  PUSH    #10000
     40 000116                  ISSW13:  PUSH    #20000
     41 000124                  ISSW14:  PUSH    #40000
     42 000132                  ISSW15:  PUSH    #100000
     43 000140  005000          TEST:    CLR     R0
     44 000142  032637                   BIT     (SP)+,@#SWREG
                177570
     45 000146  001401                   BEQ     RETURN
     46 000150  005200                   INC     R0
     47 000152  000207          RETURN:  RTS     PC
     48                         ;
     49                         ;        CONSOLE SWITCH REGISTER CONTENTS
     50                         ;
     51 000154  005725          OPTION:  TST     (R5)+
     52 000156  013775                   MOV     @#SWREG,@(R5)
                177570
                000000
     53 000164  000207                   RTS     PC
     54          000001                  .END
```

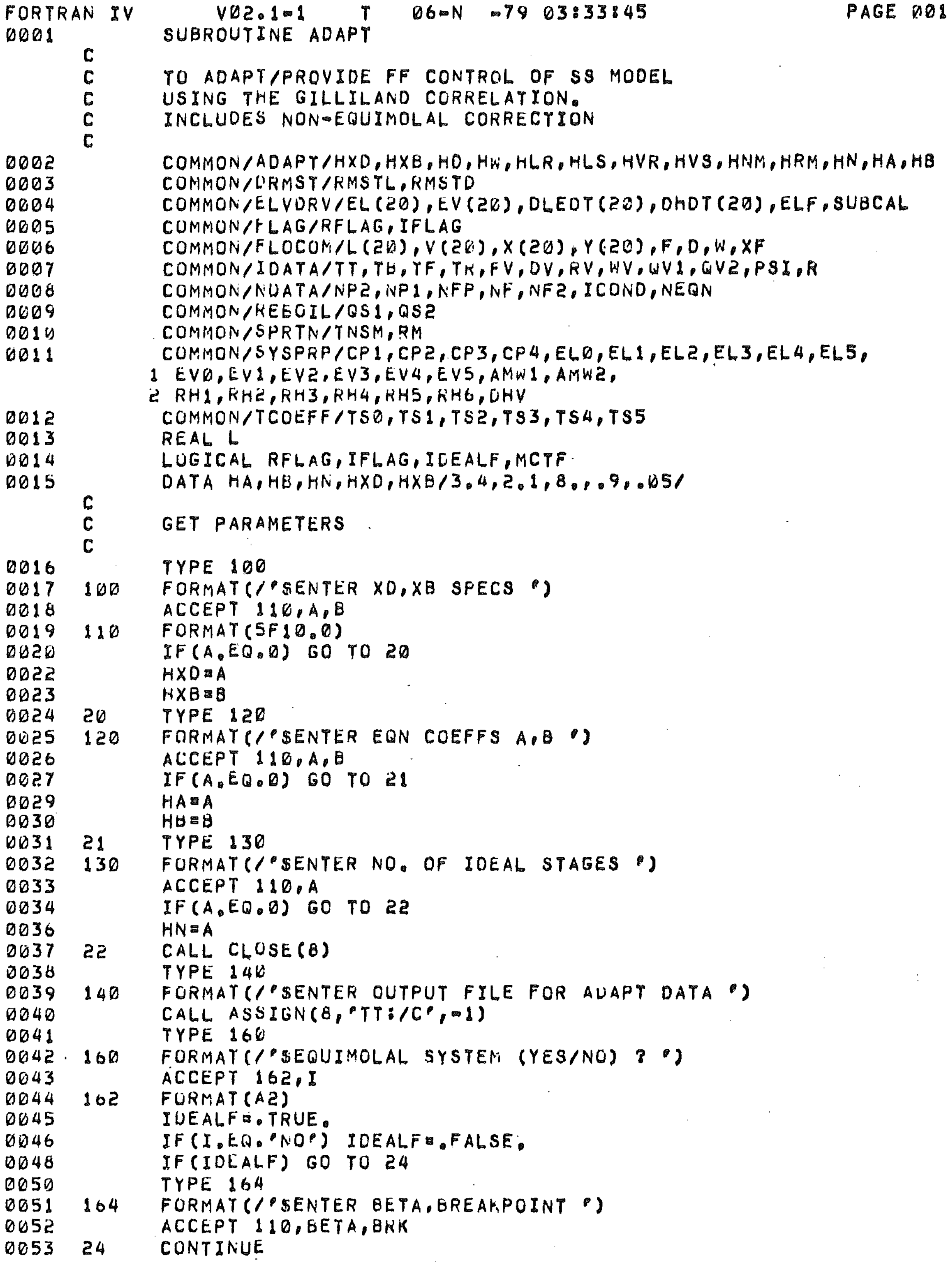
```
INPUT DATA FOR COLUMN (CONDENSER TYPE =TOTAL)
NUMBER OF PLATES      8
FEED PLATE NUMBER     5
STEADY STATE DATA:
FEED RATE             1.75000
FEED COMP.            0.46000
REFLUX RATIO          0.73000
FEED TEMPERATURE     26.00000
REFLUX TEMP          19.60000
STEADY STATE GUESSES:
DISTILLATE RATE       0.00000
BOTTOMS RATE          0.00000
TOP TEMPERATURE      69.00000
BOT TEMPERATURE      96.00000
REBOILER DUTY:QS=     0.9600 +T(NP2)*    0.0000
MURPHREE TRAY EFFICIENCY COEFFICIENTS:
 0.614E+00 0.786E+00-0.103E+01 0.436E+00 0.000E+00
STEADY STATE SOLUTION
TRAY     LIQUID          VAPOUR           LIQUID           VAPOUR
         COMP.           COMP.            RATE             RATE
  0    0.916993E+00  0.955522E+00  0.209687E+02  0.287242E+02
  1    0.826251E+00  0.917000E+00  0.229331E+02  0.496930E+02
  2    0.734174E+00  0.876714E+00  0.225924E+02  0.516574E+02
  3    0.648482E+00  0.836510E+00  0.222925E+02  0.513166E+02
  4    0.565666E+00  0.799666E+00  0.220036E+02  0.510168E+02
  5    0.489762E+00  0.764605E+00  0.929379E+02  0.507276E+02
  6    0.438800E+00  0.737355E+00  0.921769E+02  0.573013E+02
  7    0.350303E+00  0.657600E+00  0.908808E+02  0.565403E+02
  8    0.215311E+00  0.517144E+00  0.892204E+02  0.552442E+02
  9    0.916479E-01  0.297550E+00  0.356366E+02  0.535839E+02
AFTER  54 ITERATIONS,RMSTD=   0.850E-04 (WITH RMSTL=   0.100E-03)
ENERGY TRANSFER,KJ/TIME: REBOILER DUTY   0.216000E+04
                         CONDENSER DUTY  0.198886E+04
MINIMUM NUMBER OF STAGES  4.114 MINIMUM REFLUX RATIO      0.557
FEED FRAC VAPOURISED     -0.101
VOLUMETRIC FLOWS,L/TIME:   FEED          1.75
                           DISTILLATE    1.10
                           REFLUX        0.80
                           BOTTOMS       0.72
MOLAR BALANCE DISCREPANCY  0.0000
VOLUME BALANCE DIFFERENCE -0.0705
COLUMN END TEMPERATURES: TOPS        66.18 BOTTOMS     88.61
REFLUX RATIOS:  EXTERNAL =  0.730  INTERNAL =  0.812
```

SAMPLE SSGW OUTPUT

VI.3 SSGW ADDITIONS

To test the modified feedforward controller and adapter described in Chapter 8, an additional routine was included in SSGW, and a minor modification made to the main program. A listing of the modified main program and the additional routine follows.

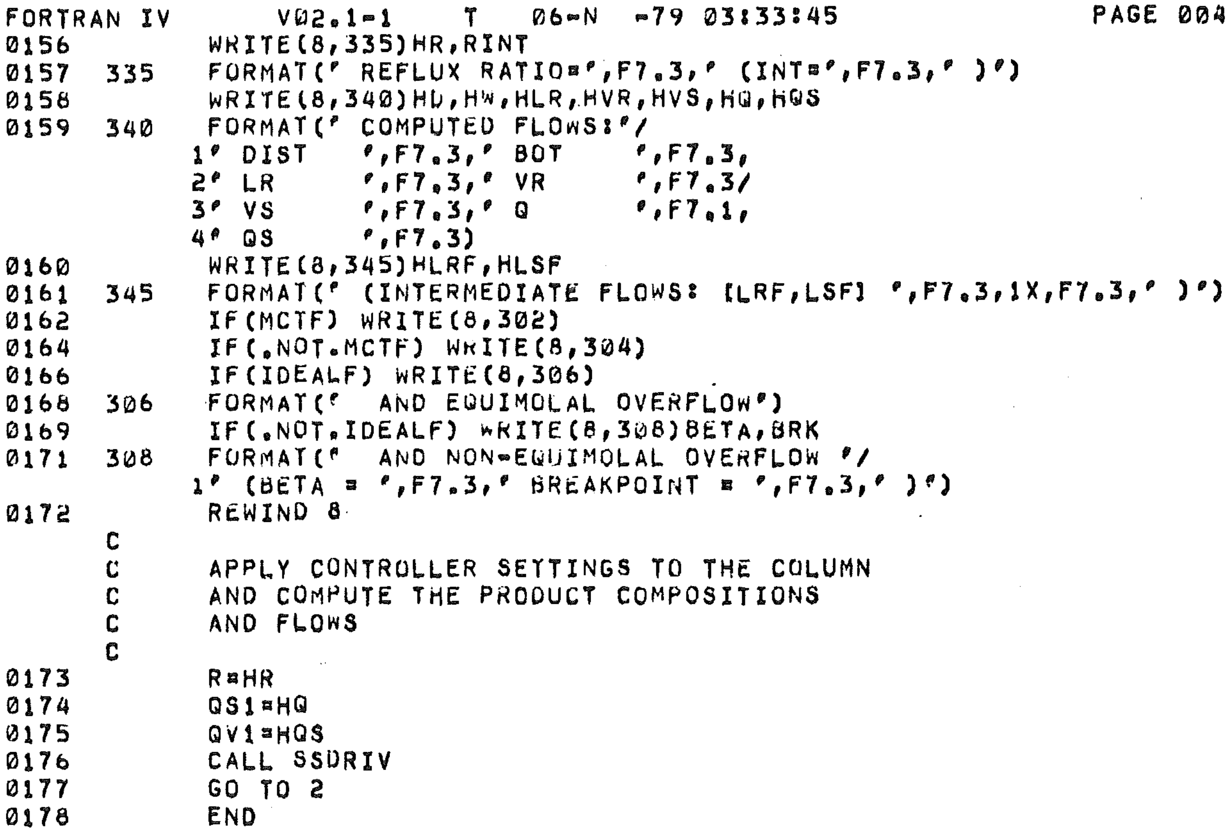
```
FORTRAN IV      V02.1-1    T    06-N  -79 02:49:48                  PAGE 001
      C                              **********
      C                              *  SSGW  *
      C                              **********
      C
      C      A BINARY STEADY STATE DISTILLATION COLUMN PROGRAM
      C
      C      THIS VERSION INCLUDES THE STEADY STATE FEEDFORWARD
      C      CONTROLLER BASED ON THE GILLILAND CORRELATION. THE SAME
      C      ROUTINES USED BY THE STANDARD VERSION OF SSGW ARE REQUIRED.
      C
      C      GRANT WILSON
      C
0001         COMMON/DRMST /RMSTL,RMSTD
0002         COMMON/ELVDRV/EL(20),EV(20),DLEDT(20),DHDT(20),ELF,SUBCAL
0003         COMMON/EQDATA /EQ(41),EQS(41),EMV(5),DX
0004         COMMON/FLAG  /RFLAG,IFLAG
0005         COMMON/FLOCOM/L(20),V(20),X(20),Y(20),F,D,W,XF
0006         COMMON/HLBL  /LABEL(20),MSSG(8)
0007         COMMON/IDATA /TT,TB,TF,TR,FV,DV,RV,WV,QV1,QV2,PSI,R
0008         COMMON/RBRNCH/IOK,IYES,INO,IHT,LP(14)
0009         COMMON/NDATA/NP2,NP1,NFP,NF,NF2,ICOND,NEQN
0010         COMMON/REBOIL/QS1,QS2
0011         COMMON/SPRTN /TNSM,RM
0012         COMMON/SYSPRP/CP1,CP2,CP3,CP4,EL0,EL1,EL2,EL3,EL4,EL5,
            1     EV0,EV1,EV2,EV3,EV4,EV5,AMW1,AMW2,
            2     RH1,RH2,RH3,RH4,RH5,RH6,OHV
0013         COMMON/TCOEFF/ TS0,TS1,TS2,TS3,TS4,TS5
0014         COMMON/TRIDM / A(20),B(20),C(20),RHS(20)
0015         LOGICAL RFLAG,IFLAG,SUBCAL
0016         REAL L
0017         DATA IOK,IYES,INO,IHT/'OK','YE','NO','HT'/
0018         CALL ASSIGN(6,'TT:')      !OUTPUT FILE
0019         TYPE 500
0020     500 FORMAT(/'1',33X,'SSGW'/34X,4('=')/
            1/,14X,' A STEADY-STATE BINARY DISTILLATION PROGRAM'/
            2 15X,42('-'))
      C
      C      CALL RDR FOR SYSTEM PROPERTIES
      C
0021         CALL RDR
      C
      C      USER OPTIONS
      C
0022      20 CONTINUE
0023         CALL READR1
      C
      C
      C      COMPUTE COLUMN VARIABLES
      C
0024         CALL SSDRIV
      C
      C      NOW IMPLEMENT THE FEEDFORWARD CONTROLLER AND ADAPTER
      C
0025         CALL ADAPT
      C
0026         GO TO 20
      C
0027         END
```



```
FORTRAN IV      VØ2.1-1     T    Ø6-N   -79 Ø3:33:45                PAGE ØØ1
ØØØ1            SUBROUTINE ADAPT
          C
          C     TO ADAPT/PROVIDE FF CONTROL OF SS MODEL
          C     USING THE GILLILAND CORRELATION.
          C     INCLUDES NON-EQUIMOLAL CORRECTION
          C
ØØØ2            COMMON/ADAPT/HXD,HXB,HD,HW,HLR,HLS,HVR,HVS,HNM,HRM,HN,HA,HB
ØØØ3            COMMON/DRMST/RMSTL,RMSTD
ØØØ4            COMMON/ELVDRV/EL(2Ø),EV(2Ø),DLEDT(2Ø),DHDT(2Ø),ELF,SUBCAL
ØØØ5            COMMON/FLAG/RFLAG,IFLAG
ØØØ6            COMMON/FLOCOM/L(2Ø),V(2Ø),X(2Ø),Y(2Ø),F,D,W,XF
ØØØ7            COMMON/IDATA/TT,TB,TF,TR,FV,DV,RV,WV,QV1,QV2,PSI,R
ØØØ8            COMMON/NDATA/NP2,NP1,NFP,NF,NF2,ICOND,NEQN
ØØØ9            COMMON/REBOIL/QS1,QS2
ØØ1Ø            COMMON/SPRTN/TNSM,RM
ØØ11            COMMON/SYSPRP/CP1,CP2,CP3,CP4,ELØ,EL1,EL2,EL3,EL4,EL5,
               1 EVØ,EV1,EV2,EV3,EV4,EV5,AMW1,AMW2,
               2 RH1,RH2,RH3,RH4,RH5,RH6,DHV
ØØ12            COMMON/TCOEFF/TSØ,TS1,TS2,TS3,TS4,TS5
ØØ13            REAL L
ØØ14            LOGICAL RFLAG,IFLAG,IDEALF,MCTF
ØØ15            DATA HA,HB,HN,HXD,HXB/3.4,2.1,8.,.9,.Ø5/
          C
          C     GET PARAMETERS
          C
ØØ16            TYPE 1ØØ
ØØ17      1ØØ   FORMAT(/'$ENTER XD,XB SPECS ')
ØØ18            ACCEPT 11Ø,A,B
ØØ19      11Ø   FORMAT(5F1Ø.Ø)
ØØ2Ø            IF(A.EQ.Ø) GO TO 2Ø
ØØ22            HXD=A
ØØ23            HXB=B
ØØ24      2Ø    TYPE 12Ø
ØØ25      12Ø   FORMAT(/'$ENTER EQN COEFFS A,B ')
ØØ26            ACCEPT 11Ø,A,B
ØØ27            IF(A.EQ.Ø) GO TO 21
ØØ29            HA=A
ØØ3Ø            HB=B
ØØ31      21    TYPE 13Ø
ØØ32      13Ø   FORMAT(/'$ENTER NO. OF IDEAL STAGES ')
ØØ33            ACCEPT 11Ø,A
ØØ34            IF(A.EQ.Ø) GO TO 22
ØØ36            HN=A
ØØ37      22    CALL CLOSE(8)
ØØ38            TYPE 14Ø
ØØ39      14Ø   FORMAT(/'$ENTER OUTPUT FILE FOR ADAPT DATA ')
ØØ4Ø            CALL ASSIGN(8,'TT:/C',-1)
ØØ41            TYPE 16Ø
ØØ42      16Ø   FORMAT(/'$EQUIMOLAL SYSTEM (YES/NO) ? ')
ØØ43            ACCEPT 162,I
ØØ44      162   FORMAT(A2)
ØØ45            IDEALF=.TRUE.
ØØ46            IF(I.EQ.'NO') IDEALF=.FALSE.
ØØ48            IF(IDEALF) GO TO 24
ØØ5Ø            TYPE 164
ØØ51      164   FORMAT(/'$ENTER BETA,BREAKPOINT ')
ØØ52            ACCEPT 11Ø,BETA,BRK
ØØ53      24    CONTINUE
```

```
FORTRAN IV      V02.1-1    T   06-N  -79 03:33:45                   PAGE 002
0054            TYPE 166
0055    166     FORMAT(/'$USE CONSTANT REL VOL (YES/NO) ? ')
0056            ACCEPT 162,I
0057            MCTF=.TRUE.
0058            IF(I.EQ.'YE') MCTF=.FALSE.
0060            IF(MCTF) GO TO 26
0062            TYPE 168
0063    168     FORMAT(/'$ENTER CST REL VOL ')
0064            ACCEPT 110,ALFA
0065    26      CONTINUE
        C
        C       GET DIRECTIONS FROM CONSOLE
        C
0066    2       CONTINUE
0067            TYPE 200
0068    200     FORMAT(/'$ADAPT,CONTROL,BOTH,OR EXIT ? ')
0069            ACCEPT 210,I
0070    210     FORMAT(A2)
0071            IF(I.EQ.'EX') RETURN
0073            IF(I.EQ.'CO') GO TO 50
0075            IF(I.NE.'BO'.AND.I.NE.'AD') GO TO 2
        C
        C       ADAPT
        C
0077            IF(.NOT.MCTF) GO TO 30
        C       USE MCCABE-THIELE TO FIND RM,NM
0079            HN2=TNSM/ALOG(HA-HB*(RM+1.)/(R+1.))
0080            GO TO 32
        C       USE FENSKE,UNDERWOOD TO FIND RM,NM
0081    30      ANM=ALOG(X(1)*(1.-X(NP2))/X(NP2)/(1.-X(1)))/ALOG(ALFA)
0082            Q=1.-PSI
0083            A=Q*(ALFA-1)
0084            B=Q-ALFA*(XF+Q-1.)+XF
0085            C=-XF
0086            XP=(-B+SQRT(B*B-4.*A*C))/2./A
0087            YP=ALFA*XP/(1.+XP*(ALFA-1.))
0088            ARM=(X(1)-YP)/(YP-XP)
0089            HN2=ANM/ALOG(HA-HB*(ARM+1.)/(R+1.))
0090    32      CONTINUE
        C       WRITE OUT ADAPTION RESULTS
0091            REWIND 6
0092            REWIND 8
0093            WRITE(8,300) X(1),HXD,X(NP2),HXB,HN2,HN,HA,HB
0094    300     FORMAT(/' ADAPTION: XD=',F6.4,' (RXD=',F6.4,' )',
               1' XB=',F6.4,' (RXB=',F6.4,' )'/
               2'             N=',F6.2,' (LAST N=',F6.2,' )',
               3' (COEFFS A,B: ',F7.3,2X,F7.3,' )')
0095            IF(MCTF) WRITE(8,302)
0097    302     FORMAT('$USING MCCABE-THEILE FOR RM,NM')
0098            IF(.NOT.MCTF) WRITE(8,304)
0100    304     FORMAT('$USING FENSKE,UNDERWOOD FOR RM,NM')
0101            WRITE(8,305)
0102    305     FORMAT(' '/)
0103            HN=HN2
0104            IF(I.EQ.'AD') RETURN
        C
        C       CONTROL
        C
```

```
FORTRAN IV      V02.1-1     T    06-N  -79 03:33:45                 PAGE 003
0106   50      CONTINUE
0107           IF(.NOT.MCTF) GO TO 60
       C       USE MCCABE-THIELE TO FIND RM,NM
       C       SAVE X(1),X(NP2) & USE RMNM TO FIND RN,NM
0109           A=X(1)
0110           B=X(NP2)
0111           X(1)=HXD
0112           X(NP2)=HXB
0113           CALL RMNM
0114           X(1)=A
0115           X(NP2)=B
       C       COMPUTE R(EXT)
0116           HR=HB*(RM+1.)/(HA-EXP(TNSM/HN))-1.
0117           GO TO 62
       C       USE FENSKE,UNDERWOOD TO FIND RM,NM
0118   60      ANM=ALOG(HXD*(1.-HXB)/HXB/(1.-HXD))/ALOG(ALFA)
0119           Q=1.-PSI
0120           A=Q*(ALFA-1)
0121           B=Q-ALFA*(XF+Q-1.)+XF
0122           C=-XF
0123           XP=(-B+SQRT(B*B-4.*A*C))/2./A
0124           YP=ALFA*XP/(1.+XP*(ALFA-1.))
0125           ARM=(HXD-YP)/(YP-XP)
0126           HR=HB*(ARM+1.)/(HA-EXP(ANM/HN))-1.
0127   62      CONTINUE
       C       COMPUTE R(INT)
       C       CALCULATE INTERNAL REFLUX RATIO
       C
0128           Z=HXD
0129           TRSATD=((((TS5*Z+TS4)*Z+TS3)*Z+TS2)*Z+TS1)*Z+TS0
0130           CP=HEATCP(Z,(TR+TRSATD)/2.)
0131           HVD=((((EV5*Z+EV4)*Z+EV3)*Z+EV2)*Z+EV1)*Z+EV0
0132           HLD=((((EL5*Z+EL4)*Z+EL3)*Z+EL2)*Z+EL1)*Z+EL0
0133           RINT=HR*(1.+CP*(TRSATD-TR)/(HVD-HLD))
       C       COMPUTE FLOWS
0134           HD=F*(XF-HXB)/(HXD-HXB)
0135           HW=F-HD
0136           HLR=RINT*HD
0137           HVR=HLR+HD
0138           IF(IDEALF) GO TO 70
       C       NON-EQUIMOLAL SYSTEM
0140           HLRF=HLR*(BETA-HXD)/(BETA-X(NFP))
0141           HLSF=HLRF+(1.-PSI)*F
0142           Z=HXB
0143           IF(HXB.LT.BRK) Z=BRK
0145           HLS=HLSF*(BETA-X(NFP))/(BETA-Z)
0146           GO TO 72
       C       EQUIMOLAL SYSTEM
0147   70      HLRF=HLR
0148           HLSF=HLRF+(1.-PSI)*F
0149           HLS=HLSF
0150   72      HVS=HLS-HW
0151           Z=HXB
0152           HVD=((((EV5*Z+EV4)*Z+EV3)*Z+EV2)*Z+EV1)*Z+EV0
0153           HLD=((((EL5*Z+EL4)*Z+EL3)*Z+EL2)*Z+EL1)*Z+EL0
0154           HQ=HVS*(HVD-HLD)
0155           HQS=HQ/DHV
       C       OUTPUT CONTROLLER RESULTS
```



```
FORTRAN IV      V02.1-1    T   06-N  -79 03:33:45                PAGE 004
0156          WRITE(8,335)HR,RINT
0157    335   FORMAT(' REFLUX RATIO=',F7.3,' (INT=',F7.3,' )')
0158          WRITE(8,340)HD,HW,HLR,HVR,HVS,HQ,HQS
0159    340   FORMAT(' COMPUTED FLOWS:'/
             1' DIST    ',F7.3,' BOT     ',F7.3,
             2' LR      ',F7.3,' VR      ',F7.3/
             3' VS      ',F7.3,' Q       ',F7.1,
             4' QS      ',F7.3)
0160          WRITE(8,345)HLRF,HLSF
0161    345   FORMAT(' (INTERMEDIATE FLOWS: [LRF,LSF] ',F7.3,1X,F7.3,' )')
0162          IF(MCTF) WRITE(8,302)
0164          IF(.NOT.MCTF) WRITE(8,304)
0166          IF(IDEALF) WRITE(8,306)
0168    306   FORMAT('  AND EQUIMOLAL OVERFLOW')
0169          IF(.NOT.IDEALF) WRITE(8,308)BETA,BRK
0171    308   FORMAT('  AND NON-EQUIMOLAL OVERFLOW '/
             1' (BETA = ',F7.3,' BREAKPOINT = ',F7.3,' )')
0172          REWIND 8
        C
        C     APPLY CONTROLLER SETTINGS TO THE COLUMN
        C     AND COMPUTE THE PRODUCT COMPOSITIONS
        C     AND FLOWS
        C
0173          R=HR
0174          QS1=HQ
0175          QV1=HQS
0176          CALL SSDRIV
0177          GO TO 2
0178          END
```

APPENDIX VII

METHANOL-WATER PROPERTIES

(1) Vapour/Liquid Equilibrium (Perry (1963))

x	y	x	y	x	y
0	0.0	.350	.697	.700	.870
.025	.155	.375	.712	.725	.882
.050	.267	.400	.729	.750	.893
.075	.345	.425	.738	.775	.904
.100	.418	.450	.750	.800	.915
.125	.470	.475	.764	.825	.926
.150	.517	.500	.779	.850	.936
.175	.548	.525	.788	.875	.947
.200	.577	.550	.800	.900	.958
.225	.606	.575	.813	.925	.967
.250	.627	.600	.825	.950	.979
.275	.647	.625	.831	.975	.990
.300	.665	.650	.846	1.00	1.00
.325	.682	.675	.858		

x, y are mole fractions.

(2) Boiling Points (Perry (1963))

$$T = 99.93 - 162.7x + 501.9x^2 - 874.2x^3 + 739.8x^4 - 240.0x^5.$$

x is mole fraction methanol

T is °C at 1 atmosphere.

(3) Enthalpy Data (Ansell et al (1951))

Saturated vapour:

$$H_V = 48.21 - 9.08y + 1.49y^2$$

y is mole fraction methanol

H_V is kJ mol^{-1}

Saturated liquid:

$$h_L = 7.524 - 13.63x + 33.93x^2 - 38.97x^3 + 16.47x^4$$

x is mole fraction methanol

h_L is kJ mol^{-1}

(4) Heat Capacity (International Critical Tables (1928))

$$C_p = (.0001T + .00343)x + .0000134T + .07467$$

T is °C

x is mole fraction methanol

C_p is kJ mol^{-1} K^{-1}

(5) Liquid Density (Mikhail and Kimel (1961))

Saturated liquid: $\rho = 0.970 - .222x$

Subcooled liquid: $\rho = 1.00 - .000361T - (.1957 + .000604T)x$

T is °C

x is mole fraction methanol

ρ is kg ℓ^{-1}

The correlations listed were used in the steady state computer model SSGW to predict system properties. The following figures show a comparison between the fitted correlations, and the published data in the literature. There was some variance between the various sources of data as can be seen from these graphs.

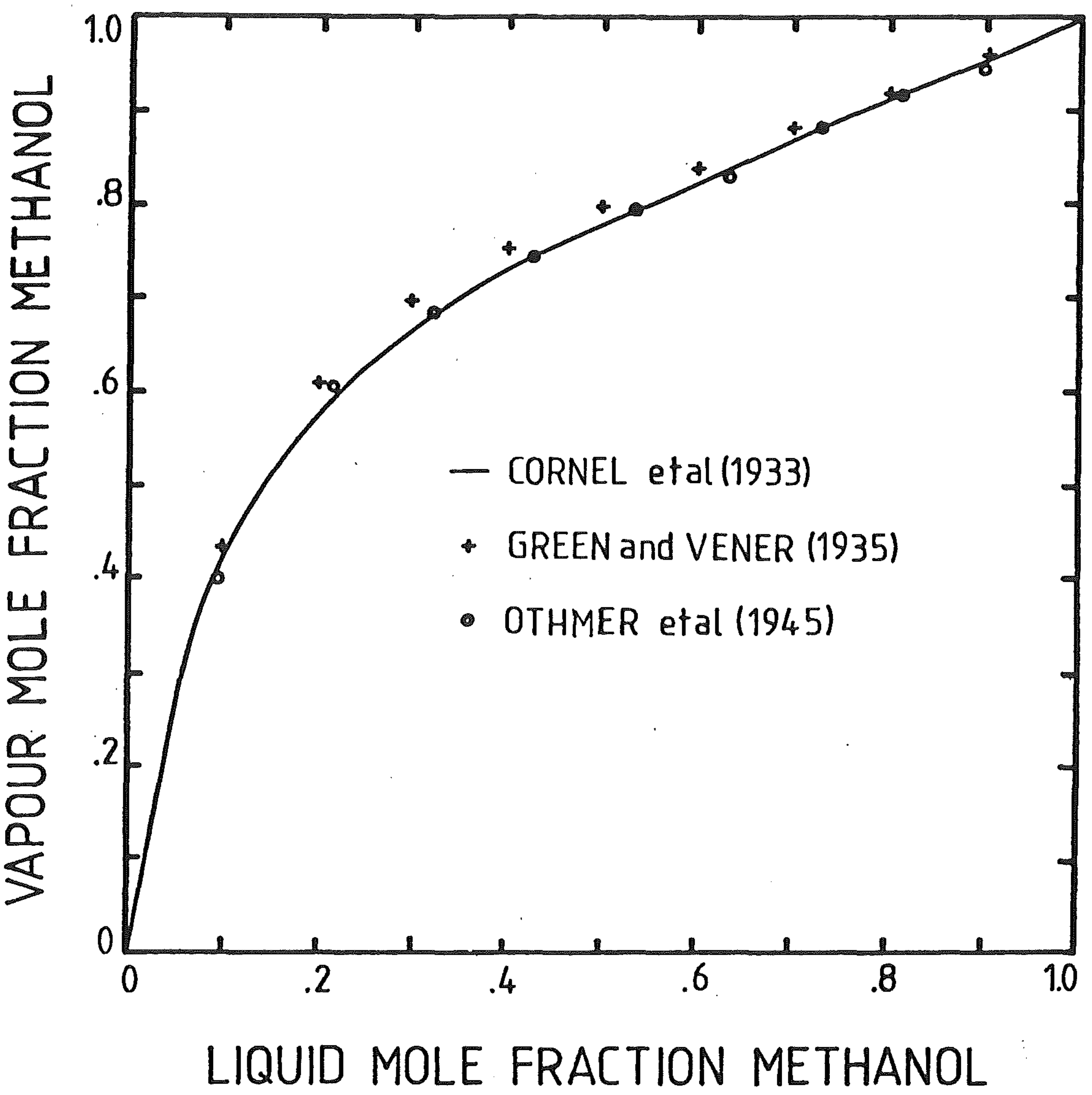


FIGURE VII-1 VAPOUR/LIQUID EQUILIBRIUM FOR METHANOL/WATER

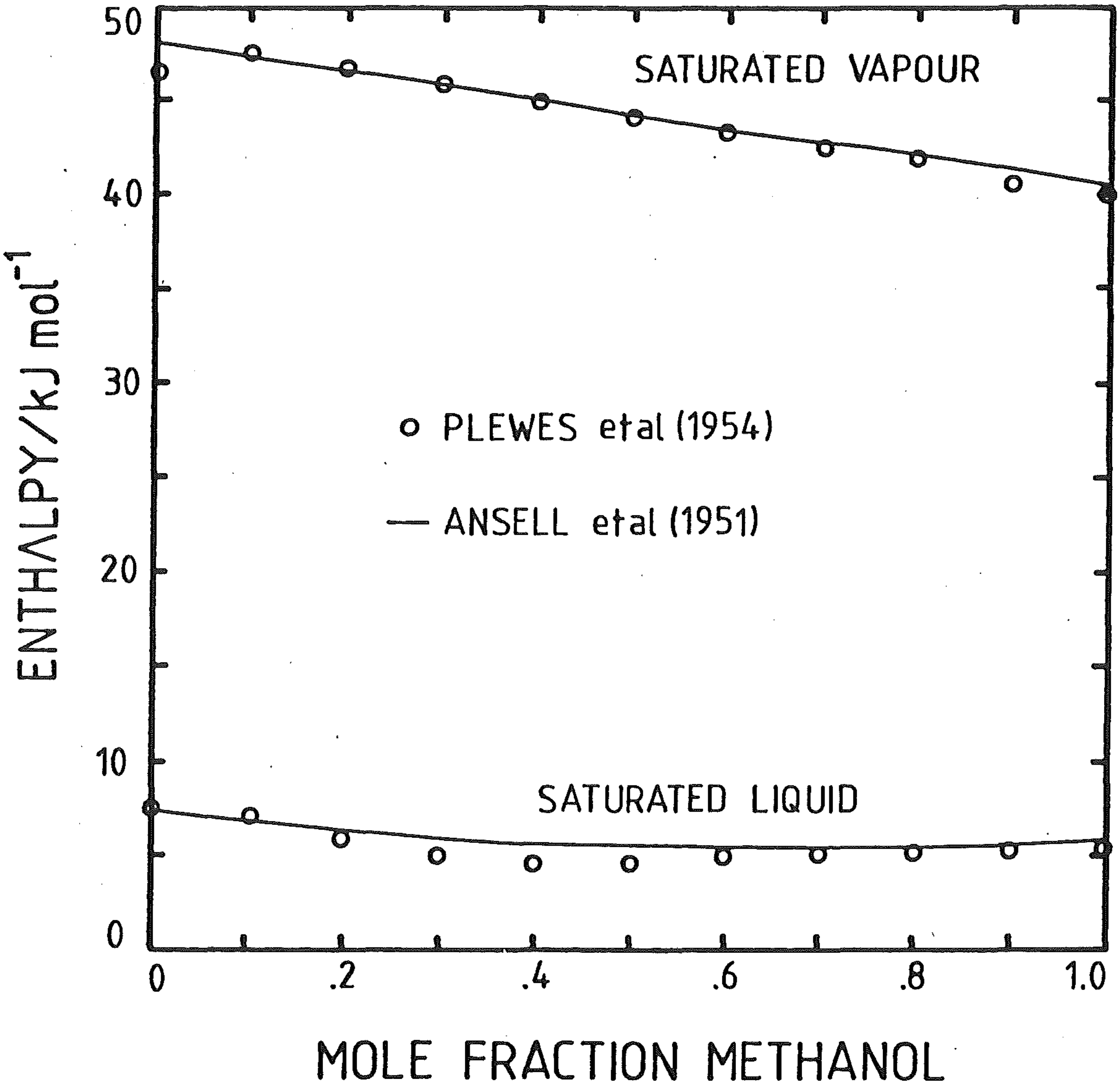


FIGURE VII-2 METHANOL/WATER ENTHALPIES

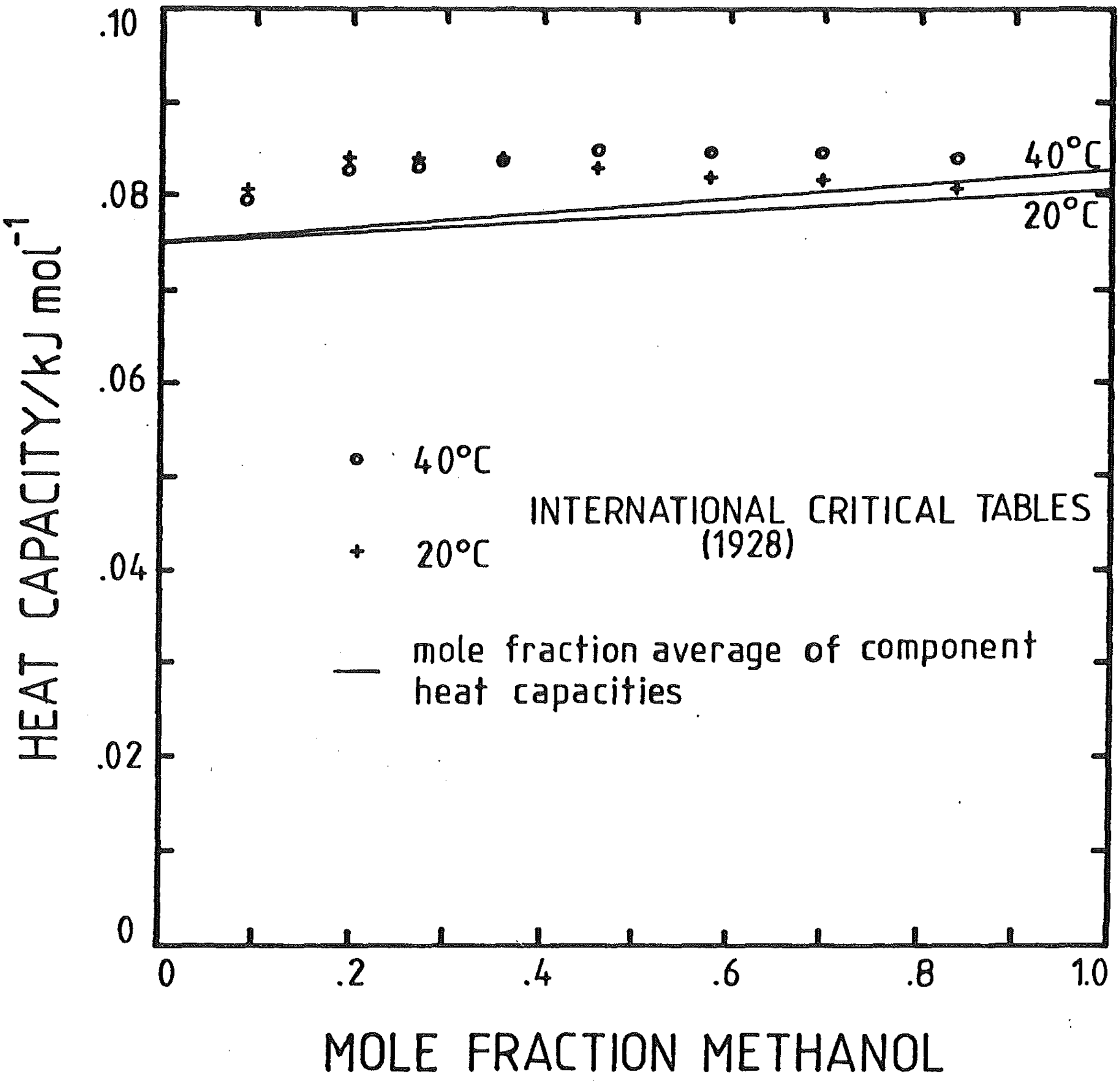


FIGURE VII-3 METHANOL/WATER HEAT CAPACITIES

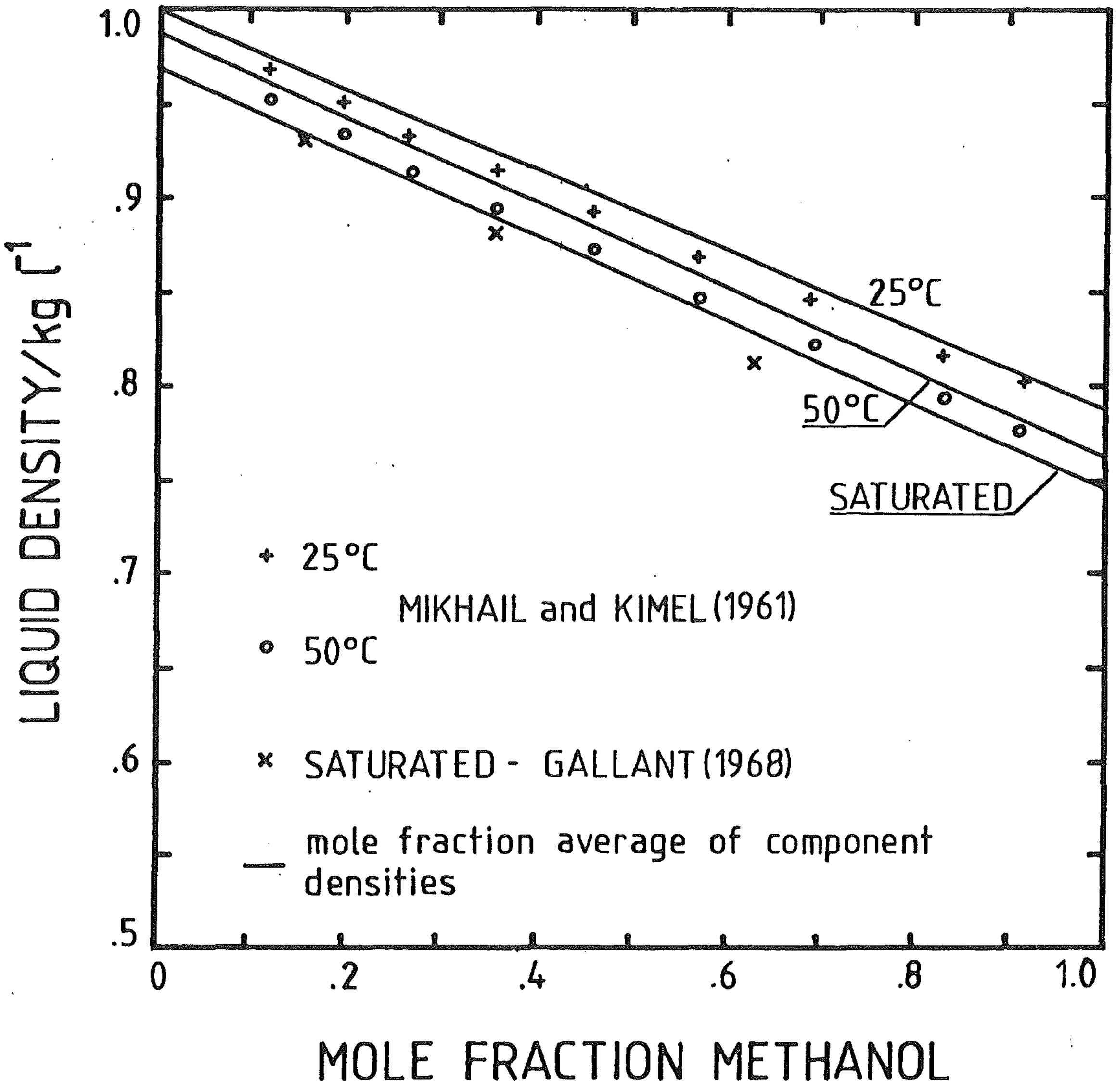


FIGURE VII-4 METHANOL/WATER LIQUID DENSITIES

APPENDIX VIII

COMPARISON OF EXPERIMENTAL DATA WITH THE STEADY STATE MODEL

VIII.1 COMPARISON OF STEADY STATE MODEL WITH EXPERIMENTAL MEASUREMENTS

The model SSGW was compared with 22 steady state experimental runs made on the column. The operating conditions for each run are given in Table VIII-1 and the comparison made in Table VIII-2. The following results were obtained from Table VIII-2:

agreement :	18 runs
1 variable disagreement :	1 run
> 1 variable disagreement :	3 runs

within the experimental errors given in the Table. The experimental errors were due to inaccuracies in the measurements and reflect both instrument error, where applicable, and process error. The likely error on the variables determined by the model program SSGW were computed using a sensitivity analysis as described in section 6.4.1. Table VIII-2 shows the comparison of the measured and predicted values of

(i)	distillate composition	(mole fraction)	x_D
(ii)	bottoms composition	(mole fraction)	x_W
(iii)	distillate flow	(mole min^{-1})	D
(iv)	bottoms flow	(mol min^{-1})	W
(v)	reflux flow	(mol min^{-1})	L_R

The difference between the two values was calculated as

$$\Delta x_D = (x_D)_{\text{experimental}} - (x_D)_{\text{model}}$$

The results for these experiments are discussed in Chapter 6. Runs 9 and 10 failed to converge because the high reflux ratios (1.90, 2.54) caused instability in the iterative solution of the model equations, and produced distillate compositions in excess of 100 mol%.

TABLE VIII-1

COLUMN OPERATING CONDITIONS FOR STEADY STATE RUNS

RUN	$F/\ell\text{min}^{-1}$ ± .05	x_F/m.f. ± .005	T_F/°C ± 1	T_R/°C ± 1	R ± 0.03	Q_S/kg min^{-1} ± .02
1	1.75	.515	26.0	19.0	0.73	0.98
2	"	.480	"	19.0	0.76	0.82
3	"	.500	"	20.0	1.15	1.17
4	"	.540	"	19.0	0.57	1.11
5	"	.440	"	18.0	0.83	1.15
6	"	.460	"	19.0	1.13	1.13
7	"	.450	"	19.0	1.14	1.16
8	"	.450	"	19.0	1.41	1.16
9	"	.425	"	19.0	1.90	1.16
10	"	.400	"	19.0	2.54	1.16
11	"	.500	"	18.0	0.67	1.13
12	"	.490	"	18.0	0.62	1.12
13	"	.490	"	18.0	0.59	1.11
14	"	.470	"	18.0	0.91	1.20
15	"	.480	"	19.0	0.97	1.19
16	"	.470	"	19.0	0.80	1.09
17	"	.460	"	19.0	0.74	0.78
18	"	.460	"	19.0	0.74	0.84
19	"	.460	"	19.0	0.75	0.97
20	"	.460	"	20.0	0.73	1.03
21	"	.480	"	20.0	0.74	1.10
22	"	.485	"	20.0	0.78	1.20

Murphree Vapour Efficiency was given by

$$E_{MV} = .436x^3 - 1.03x^2 + .786x + .614$$

F =	feed rate	T_R =	reflux temperature
x_F =	feed composition	R =	reflux ratio
T_F =	feed temperature	Q_S =	steam flow

8 trays used, feed on to fifth tray from the top.

TABLE VIII-2

COMPARISON OF STEADY STATE MODEL PREDICTIONS AND EXPERIMENTAL RESULTS

	EXPTL	MODEL		EXPTL	MODEL		EXPTL	MODEL		EXPTL	MODEL		EXPTL	MODEL		
RUN	x_D	x_D	Δx_D ±.010	x_W	x_W	Δx_W ±.032	D	D	ΔD ±1.1	W	W	ΔW ±2.7	L_R	L_R	ΔL_R ±1.1	R
1	.925	.928	-.003	.160	.180	-.020	28.7	27.6	1.01	33.7	34.0	- .3	21.0	20.1	.9	.73
2	.920	.926	-.006	.225	.249	-.024	20.8	21.6	-.8	40.1	41.8	-1.7	17.2	16.4	.8	.76
3	.955	.950	.005	.125	.153	-.028	26.4	27.1	-.7	37.4	35.2	2.2	30.3	31.2	-.9	1.15
4	.895	.901	-.006	.020	.038	-.018	31.4	35.1	-3.7	27.3	25.3	2.0	20.1	20.0	0.1	.57
5	.915	.911	.004	.040	.028	.012	30.5	30.5	0	34.4	34.9	- .5	25.2	25.3	-.1	.83
6	.930	.936	-.006	.055	.065	-.01	28.2	29.2	-1.	33.0	35.1	-2.1	28.5	29.4	-.9	1.01
7	.930	.923	.007	.085	.087	-.002	27.2	27.3	-.1	37.0	37.5	- .5	31.1	31.2	-.1	1.14
8*	.955	.900	.055	.110	.180	-.070	24.5	24.2	.3	39.5	40.7	-1.2	34.5	34.1	.4	1.41
9*	.960	-	-	.155	-	-	20.1	-	-	44.6	-	-	38.2	-	-	1.90
10*	.980	-	-	.175	-	-	16.6	-	-	49.5	-	-	42.1	-	-	2.54
11	.905	.905	0	.020	.035	-.015	33.6	33.3	0.3	27.3	29.0	-1.7	22.5	22.3	0.2	.67
12	.890	.888	.002	.01C	.025	-.015	34.1	33.8	0.3	29.8	29.0	0.8	21.3	21.0	0.3	.62
13	.880	.882	-.002	.010	.024	-.014	35.0	34.1	0.9	29.3	28.7	0.6	20.5	20.1	0.4	.59
14	.920	.926	-.006	.025	.039	-.014	31.7	31.0	0.7	31.5	32.8	-1.3	28.8	28.2	0.6	.91
15	.920	.912	-.008	.010	.012	-.002	33.0	32.9	0.1	27.6	30.4	2.8	31.9	31.9	0.0	.97
16	.920	.925	-.005	.065	.079	-.014	30.1	29.5	0.6	34.4	34.3	0.1	24.1	23.6	0.5	.80
17	.915	.921	-.006	.250	.247	.003	19.8	20.4	-.6	43.3	44.0	- .7	14.6	15.1	-.5	.74
18	.915	.921	-.006	.150	.166	-.016	24.4	25.0	-.6	39.3	39.4	- .1	18.0	18.5	-.5	.74
19	.915	.921	-.006	.120	.137	-.017	26.0	26.5	-.5	38.6	37.9	0.7	19.6	19.9	-.3	.75
20	.915	.917	-.002	.080	.091	-.011	29.2	28.7	0.5	34.6	35.6	-1.0	21.2	21.0	0.2	.73
21	.915	.917	-.002	.050	.062	-.012	31.2	31.0	0.2	32.6	32.4	0.2	23.2	22.9	0.3	.74
22	.910	.902	.008	.020	.018	.002	33.9	33.3	0.6	27.9	29.7	-1.8	26.3	26.0	0.3	.78

* Failed to converge. All compositions in mole fractions, all flows in mol min^{-1}

VIII.2 COMPARISON OF THE STEADY STATE MODEL WITH THE EXPERIMENTAL DATA OF SVRCEK

Svrcek (1967) used a similar column to separate a methanol/water mixture. There was no data on the bubble cap geometries used by Svrcek so a constant Murphree vapour efficiency of 90% was assumed (based on an estimated bubble cap slot 3.8 cm long and the correlation of Bakowski (1969)). In general, the experimental data of Svrcek and the predictions from the steady state model agreed within the tolerances described previously in this appendix. Two sample runs are listed below in Table VIII-3.

TABLE VIII-3

COMPARISON OF SSGW AND THE EXPERIMENTAL DATA OF SVRCEK (1967)

RUN S - 1

Feed Rate	= 1.12 ℓ min^{-1}	Number of Trays	= 8
Feed Comp.	= 0.366	Feed Tray	= 6*
Feed Temperature	= 75°C	(*from the top of the column)	
Reflux Temperature	= 65°C		
Reflux Ratio	= 1.94		
Steam Flow	= 0.81 kg min.^{-1}		

VARIABLE	SVRCEK	SSGW
X_D (m.f.)	.925	.938
X_W (m.f.)	.005	.002
D (mol min^{-1})	17.0	16.8
W (mol min^{-1})	26.3	26.3
L_R (mol min^{-1})	33.0	32.5
x_1 (m.f.)	.840	.854
x_2 (m.f.)	.735	.726
x_3 (m.f.)	.576	.551
x_4 (m.f.)	.383	.336
x_5 (m.f.)	.216	.180

x_6 (m.f.)	.148	.119
x_7 (m.f.)	.051	.032
x_8 (m.f.)	.016	.008

x_i = composition of the liquid leaving tray i

RUN S -13

Feed Rate = 1.28 ℓ min^{-1} Number of Trays = 8

Feed Composition = .375 Feed Tray = 6*

Feed Temperature = 75°C (* from the top of the column)

Reflux Temperature = 65°C

Reflux Ratio = 1.31

Steam Flow = 0.72 kg min^{-1}

VARIABLE	SVRCEK	SSGW
x_D (m.f.)	.948	.951
x_W (m.f.)	.010	.011
D (mol min^{-1})	18.9	18.9
W (mol min^{-1})	29.6	30.0
L_R (mol min^{-1})	24.9	24.8
x_1 (m.f.)	.882	.886
x_2 (m.f.)	.802	.800
x_3 (m.f.)	.706	.693
x_4 (m.f.)	.569	.565
x_5 (m.f.)	.439	.426
x_6 (m.f.)	.311	.300
x_7 (m.f.)	.178	.147
x_8 (m.f.)	.090	.046

x_i = composition of the liquid leaving tray i