A GENERALISED APPROACH FOR THE PREDICTION OF LASER CUTTING PARAMETERS

by

Mohammed A. Hussain

Department Of Mechanical Engineering

The University Of Glasgow

May 1989

Submitted

As A Full Submission For The Degree Of PhD

TO MY BROTHER

ABDUL-KAREEM

LIST OF CONTENTS

CH.6 : THE PREDICTION MODULE.

LIST OF FIGURES

LIST OF GRAPHS

LIST OF CHARTS

ACKNOWLEDGEMENT

I am much in debt to Mr. Finlay D. Buchan for his supervision, guidance and support, and to Prof. Brian F. Scott for his precious advice and criticism. My thanks are also due to Dr. Chris R. Chatwin for his advice and support.

I wish to forward my sincere thanks to the technical staff in the Department of Mechanical Engineering for their efforts in maintaining the experimental and computational, especially, Mr. Cameron Miller, Mr. George Falconer, Mrs. linda McCormic and Mrs. Carol Carmichael.

Last, and not least, I acknowledge the patience and moral support of my family, and The Government of Republic of Iraq for their sponsorship.

SUMMARY

A design of a tasking and control environment for laser based manufacturing systems is proposed. This uses an empirical approach based on recording previous manufacturing experience in a database, in order that this can be used in planning and control of future processes. The work presented gives details of the partial implementation of this design for laser cutting systems. This makes use of Computer Aided Design and Manufacture, Computer Numerical Controlled laser cutting machines, database and computerised planning and control.

A database module to record laser cutting experience, was implemented using a relational database management system. The database is systematically divided into a number of tables, for efficient data definition and manipulation.

An experimental programme was accomplished, based on 610 cutting experiments and post-operation cut quality analyses of various materials (mainly metallic) using two types of laser: a pulsed, solid state, Nd:YAG laser and a continuous wave CO_2 laser. Data describing the cutting processes and post-operation measurements were inserted into the database module to provide an initial data set for the development and testing of this module and other software modules of the planning and control system.

An 11 stage strategy to search the database for data which can be used for the planning and control of a new cutting process, has been implemented in a computer program (the data collection module). This program inquires from the user information relating to his prospective application, and interacts with the database module to collect the relevant data.

The data collected, is then used in a prediction module which predicts process parameters suitable for the prospective application, against a set of cut quality features desired by the user.

INTRODUCTION

Lasers have been used for more than two decades to perform a diversity of tasks in the laboratory and industry. Material processing is one of the well accepted industrial applications of lasers due to its ability to adapt to a wide variety of tasks, including: welding, drilling, surface engineering, heat treatment and cutting.

As a slitting (cutting) machine tool, laser based systems can exhibit a remarkable performance in their ability to cut a wide range of metallic and non-metallic materials, with thicknesses ranging from a fraction of a millimetre to more than 20 mm. This is possible due to the nature of the laser cutting process which is characterised by:

a) A powerful heat source focused to a very small spot on the material surface (which is approximately 0.1 to 0.3mm in diameter), which results in a narrow kerf width (cut slot width), a small heat affected margin, reduced thermal stresses, relatively high cutting speed and improved cut quality when compared to other thermal processes.

b) No hard fixturing being needed as no cutting forces are transmitted from the cutter to the workpiece. This relieves the workpiece from mechanical stresses, and enables the cutting of delicate or thin materials.

c) The process is appropriate for trial and small batch production, as the handling of both workpiece and laser beam is reasonably simple and no special fixturing or tooling is required.

Laser based manufacturing performance can be significantly enhanced by the utilisation of computer aided design, numerical control, and multi-axis work handling machines.

Current laser cutting processes are almost entirely dependent on human expertise, which is not easily obtained and subject to errors or misjudgement caused by human factors. The inability of existing mathematical models to completely describe the cutting process, because of the large number of interacting variables involved, precludes the accurate estimation of process parameters, cut characteristics or identification of process control handles, at the present time.

This suggests that an alternative method(s) is required for the prediction and control of process parameters capable of producing the desired cut features.

An integrated, modular system for the planning and control of cutting processes in laser based manufacture is proposed. This is based on recording previous experience from cutting experiments and using it for planning future processes. The outline of this proposal and its current status is given in chapter 2.

Chapter 3 discusses the requirements and implementation of a database management system used for recording data gathered from cutting experiments.

An experimental programme of cutting trials and their results are presented in chapter 4.

Strategies for searching the database system to collect a data set which can be used in the operation of planning and control of a new cutting process are discussed in chapter 5.

The chosen strategy for predicting a set of suitable process parameters against desired cut quality features is presented in chapter 6.

Finally, conclusions, comments and recommendations for further work are given.

CHAPTER 1

LASER CUTTING, A LITERATURE REVIEW

1.1 INTRODUCTION

The concept of LASER (Light Amplification by Stimulated Emission of Radiation) was practically recognized in 1960 when Theodore Maiman operated the first laser, a Ruby laser emitting a pulsed beam of collimated red light (ref 1.1). In 1961, the first continuous wave (CW) laser, a Helium-Neon gas laser operated in the infrared region.

The first CO_2 laser was operated in 1964 by pulsed discharge through pure CO_2 (ref 1.2). Further developments to this laser were reported during 1964-65 by the addition of Nitrogen and Helium to the CO_2 gas which created a dramatic change in the emitted laser beam power (from 200 mW to multi-kilowatt, ref 1.2). The major developments which largely gave rise to the existing generation of industrial lasers occured in the late sixties, and a steady continual development in sources since then has occurred (ref´s 1.1 and 1.2).

Many types of laser have been developed for use in the laboratory and industry, for a diverse range of applications such as: inspection, surveying, spectroscopy, data processing, welding, cutting, heat-treatment and drilling (ref 1.1). Lasers are now generally accepted as a basis for manufacturing tools in the automotive and aerospace industry (ref´s 1.3, 1.4 and 1.5), and in sheet metal processing (ref´s 1.6 and 1.7). The incorporation of computer control technology and multi-axis (or robotic) beam handling, has greatly increased the flexibility and cost effectiveness of laser based manufacturing (ref´s 1.8, 1.9, 1.10, 1.11 and 1.12).

In this chapter, a general review of laser cutting is given, the various mechanisms of laser cutting are discussed, and the factors affecting the laser cutting process and the resulting cut quality are covered.

1.2 THE LASER AS A CUTTING TOOL

Lasers have become increasingly important as cutting tools due to favourable cutting characteristics (advantages) over other cutting processes, e.g sawing, flame and plasma cutting, stamping, nibbling and water-jet cutting (ref´s 1.13 and 1.14).

Cuts made with lasers have a narrower kerf width and heat affected zone (HAZ) when compared with flame cutting, and more square edges than plasma cuts. Unlike flame and plasma cutting, lasers can cut non-metallic materials e.g polymers, fabrics, wood and glass. The cut edge can be smoother than that created by a band saw.

No mechanical force is exerted upon the material as there need be no contact between the cutting tool and the workpiece. Consequently, no hard fixturing is needed and no stresses are generated within the material. The absence of tool wear and need for tool changes, gives advantages over stamping and nibbling processes.

The development of multi-axis and robotically manipulated laser beam and/or workpiece using computerised control (CNC), has enhanced laser cutting flexibility and gives the capability of 3-dimensional contouring of large components (ref´s 1.7, 1.15 and 1.16).

Laser based manufacture has been reported as a potential application within flexible manufacturing systems (ref 1.10, 1.11 and 1.12). This is due to the flexibility of laser processing indicated by Ikeda (ref 1.12) which includes: its applicability to several processes (e.g cutting, welding and drilling), the capability of processing a wide range of metallic and non-metallic materials, and the simplicity of beam handling from source laser to work station(s). Automated laser cutting was reported as a successful technique for remote operation in the nuclear industry for cutting and welding of nuclear fuel rods inside a nuclear reactor after irradition (ref 1.17).

1.3 LASER CUTTING MECHANISMS

The ability of laser systems to deliver a high power intensity to a localised area on a workpiece by the focusing of a laser beam, is perhapes the most important feature for material processing. Fig 1.1 illustrates the various approximate regimes of laser power intensity and interaction time suitable for different laser material processes.

When a laser beam is focused on a workpiece surface (fig 1.2) part of its energy is absorbed by the material (the rest being reflected or transmitted in non-opaque materials). This absorption increases with higher material temperature (consequently, with interaction time) and higher beam intensity (ref´s 1.1, 1.2, 1.13 and 1.18). The energy absorbed heats the surface and produce a heat flow within the material. This heat flow depends mainly on the material thermal diffusivity (a) which is determined by the material thermal conductivity (K), specific heat (Cp) and density (d):

$$a = K / d\,Cp$$

The amount of energy absorbed and the heat flow within the material, specifies the way in which material is removed from the cut kerf (slot).

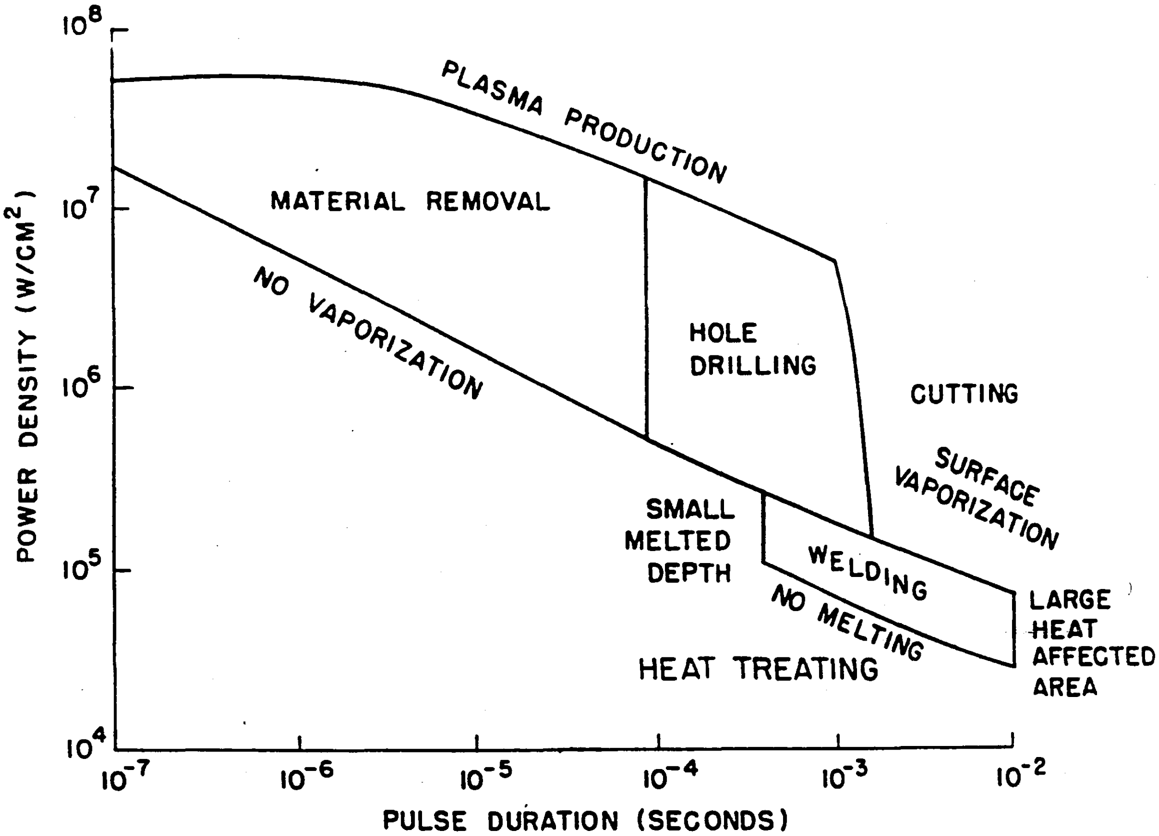


FIG 1.1 : REGIMES FOR LASER POWER INTENSITY AND PULSE DURATION SUITABLE FOR MATERIAL PROCESSING

(After Ready ref 1.1)

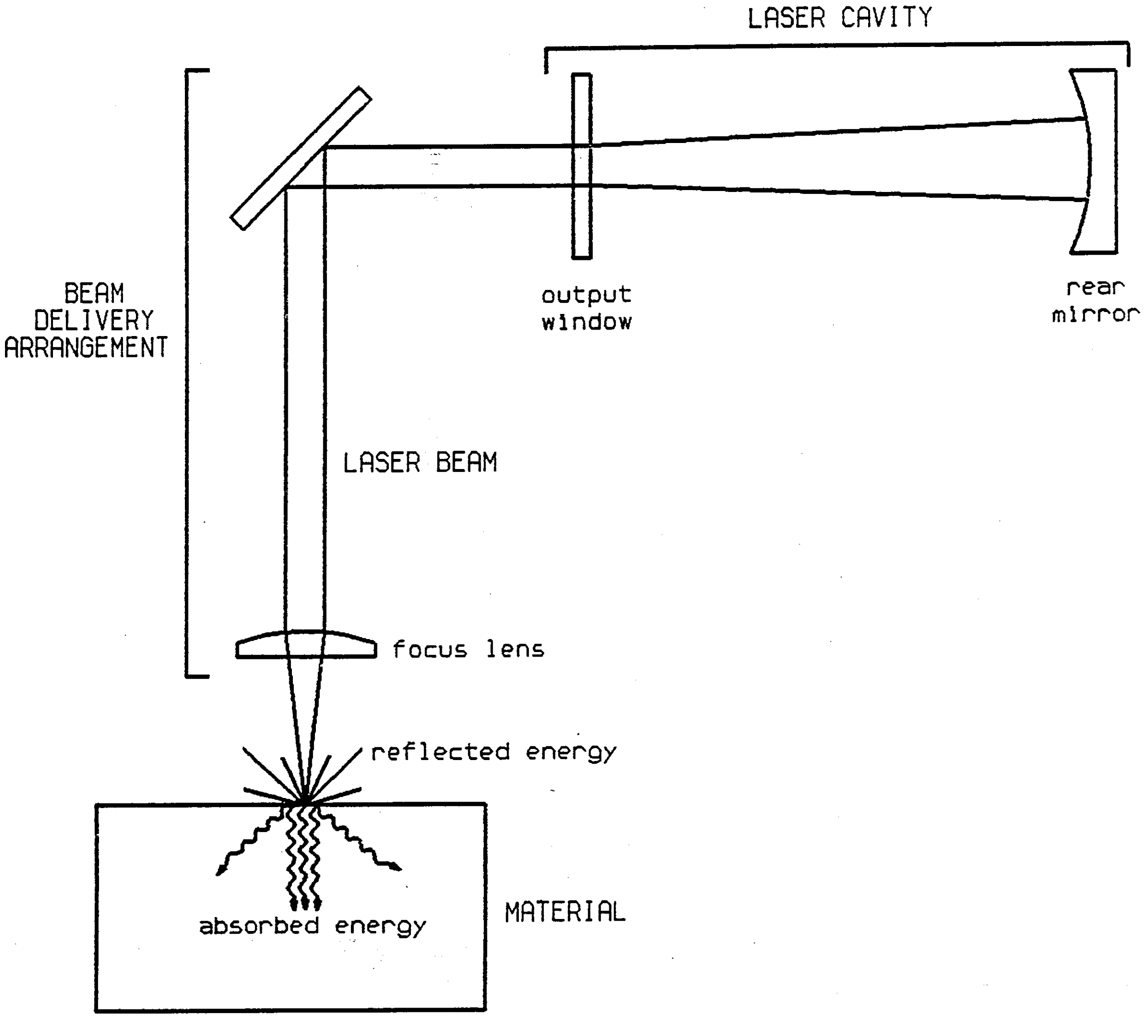


FIG 1.2 : PRINCIPLE OF LASER CUTTING

Steen and Kamalu (ref 1.13) identified five ways in which a focused laser beam can be used to slit different materials:

1) Vapourisation.
2) Melting and ejection (fusion cutting).
3) Burning in reactive gas (reactive gas assisted cutting).
4) Thermal stress cracking (controlled fracture).
5) Scribbing.

The first three mechanisms comprise the area of interest to present project, and are considered in more detail.

1.3.1 VAPOURISATION CUTTING

This is the usual mechanism for cutting or hole drilling with pulsed lasers (ref 1.13), or when cutting organic materials such as wood and plastics with CW lasers (ref 1.1). Material is removed from the cut slot as vapour and ejected molten particles, this occurs when a power intensity greater than 1 MW/cm^2 (for metals) is incident on the material surface, this threshold may vary with different metals (ref 1.1, 1.2 and 1.13).

This cutting mechanism is complicated and may involve various processing regimes depending on power intensity available. These are summarised as (see fig 1.3):

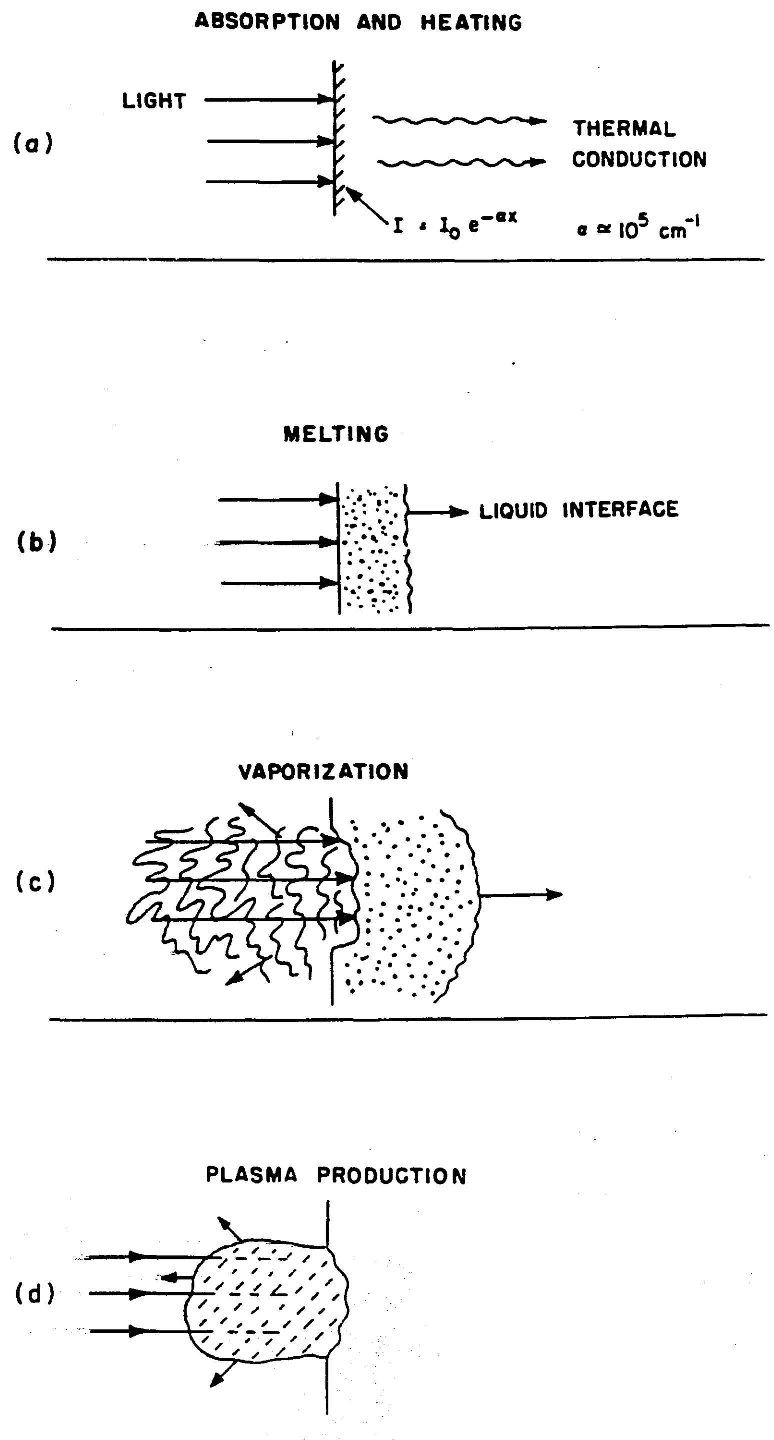


FIG 1.3 : STAGES OF MATERIAL REMOVAL BY VAPOURISATION

(After Ready ref 1.1)

a) A localised area of the material surface is irradiated with an incident laser beam. Part of the beam energy is reflected, the rest is absorbed and penetrates the material by thermal conduction.

b) When the surface reaches the melting temperature, a melting front propagates into the material.

c) With continued irradiation, the surface temperature rises to the boiling point and escaping vapour takes with it molten particles. As these particles need no evaporation energy, material removal is further enhanced, the depth of the molten layer becomes minimum and power absorption takes its maximum limit (ref 1.18).

d) A super-heated opaque plasma cloud is created near the material surface and propagates toward the laser source. This is called a Laser Supported Absorption Wave (LSAW, ref 1.1). This cloud shields the material surface from the incident beam by absorption and reflection. The material is then indirectly heated by the plasma cloud, and consequently the size of the affected area is controlled by the dimensions of the plasma, which may be considerably larger than the beam spot diameter. If, however, only a partial power absorption by the plasma cloud takes place, this may act as a lens changing the beam focusing properties, i.e refocusing the beam at a point which is shorter or longer (defocusing) than the focal length (ref´s

1.13 and 1.18).

e) If the material is partially transparent to the incident laser wavelength (e.g glass and some plastics to YAG beam), beam energy is absorbed internally by the material, and boiling may occur below the surface leading to explosion and cracking below the surface (ref 1.13).

1.3.2 CUTTING BY FUSION AND EJECTION

This mechanism is usually utilised when cutting with CW lasers and a non-reactive (inert) gas jet, with power intensity between 100 kW/cm^2 and 10 MW/cm^2 (ref 1.1). This method is used either for highly reactive materials (e.g when cutting titanium) where undesirable and unstable material burning may result when using a reactive gas (Oxygen), or for cutting non-reactive materials (as plastics) where air is cheaper to use. This cutting mechanism can be summarised as (ref 1.13):

a) The laser beam initially forms a small hole (keyhole) in the workpiece material by vapourisation and ejection of molten material.

b) This keyhole absorbs the beam energy as a blackbody (due to increased absorption), and is surrounded by walls of molten material supported by the fast flow of material vapour.

c) The melting isotherm propagates through the workpiece thickness and the assistant gas jet ejects the molten material out of the rear side of the kerf.

d) The relative movement between the laser beam and the workpiece, causes the hole (formed in c) to traverse across the material, forming a slot. The laser beam in this process is, consequently, concentrated upon the leading edge of the slot (fig 1.4), from which a flow of molten material is ejected by the gas jet. This ejection can be either as a steady flow (ref 1.13), or pulsed leaving a striated structure on the cut surface (ref 1.19).

1.3.3 REACTIVE FUSION CUTTING

This cutting mechanism is similar to the previous, except that a reactive gas (usually Oxygen) is used instead of a non-reactive or inert gas. This creates another heat source due to the reaction of the gas with the material (oxidisation or burning), which may provide a substantial proportion of the cutting energy (up to 70%, ref 1.19). The rate of this reaction depends on the gas flow and material mass transfer rate to the molten zone and subsequently to the reaction front. When the rate of reaction becomes faster than the laser beam traverse speed, i.e the process is dominated by material oxidisation rather than a laser-melting/vapourisation interaction, the kerf widens considerably with rough edges (resembling the cut made by an oxy-acetelyne torch). A detailed experimental study of the phases of this mechanism that occur in mild steel, using high speed in-process filming, has been performed by Arata et al (ref 1.20).

Several attempts to theoretically model the process of laser cutting have been reported, among these are the analyses given in ref´s 1.2, 1.13 and 1.21 through 1.25. These analyses are based on considering the heat source induced by the laser beam as a point, line (vertical through material), surface disc or a cylinder, where the heat flow within the material is considered as one, two or three dimensional flow.

However, these models are limited in their prediction capabilities due to the vast number of factors affecting the cutting process. The general deficiencies of these models are that they do not handle all process factors, e.g:

- Heat losses due to radiation, convection and conduction.
- Removal of molten metal from the kerf.
- Heat input from oxidisation reaction.
- Variations in beam diameter or profile of beam power intensity (mode).
- Variation in material reflection or absorption during the heating process.
- Beam penetration depth.

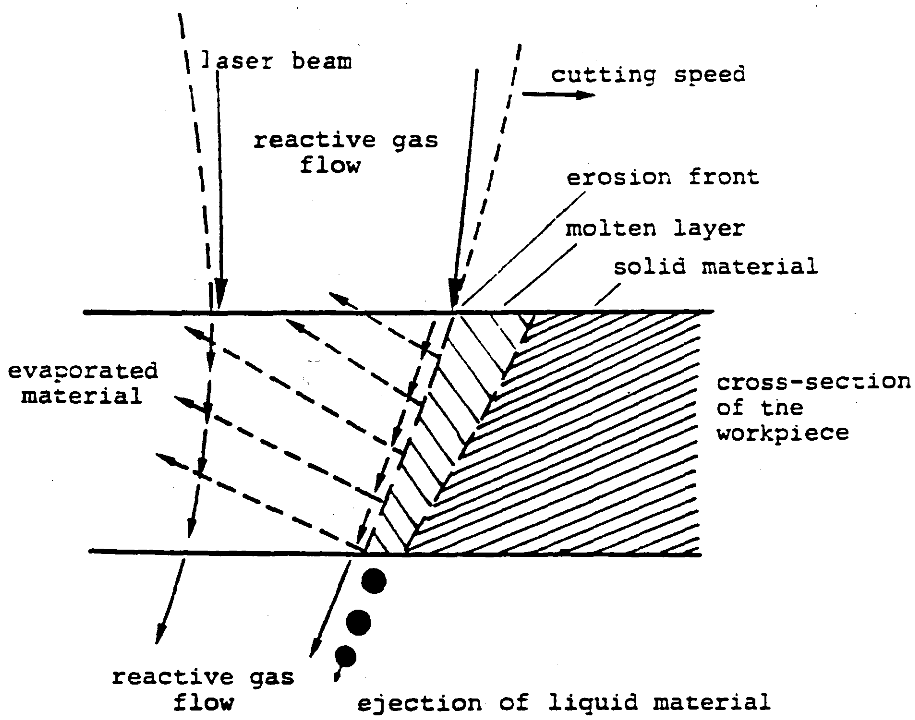


FIG 1.4 : GAS ASSISTED LASER CUTTING BY FUSION

(After Schuocker & Abel ref 1.24)

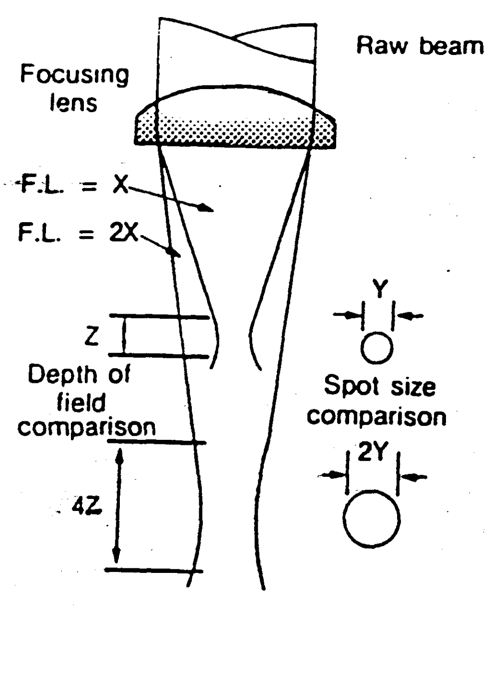


FIG 1.5 : EFFECT OF LENS FOCAL LENGTH

(After darchuk & Migliore ref 1.28)

1.4 FACTORS AFFECTING LASER CUTTING

Fig 1.2 features the three basic elements required for the process of laser cutting (and generally, laser material processing), these are:

- A laser beam.
- A method of laser beam delivery.
- Workpiece material.

The cutting process is a function of the interaction between these elements, and consequently, of their (or their components) characteristics. These characteristics affect the cutting process either independently (e.g material density and thermal conductivity), or depending on other characteristics (e.g power intensity and material absorption). In gas assisted laser cutting, other factors become influential such as gas nozzle design, nozzle stand-off distance and gas pressure. These factors shall be discussed in more detail.

1.4.1 LASER BEAM EFFECTS

The characteristics of a laser beam are related to the design of the resonator,lasing medium, optical design and method of excitation. The main characteristics of a laser beam are:

- Total power (P).
- Spatial intensity profile (beam mode).
- Diameter (D).
- Wavelength (L).
- Divergence.
- Polarisation.

These characteristics influence the focused beam intensity and its absorption at the material surface, and thus affect the cutting performance. The maximum beam intensity (I) can be calculated (assuming gaussian beam, ref 1.1):

$$I = 8P / \pi d^2$$

Where (d) is the focus spot diameter (spot size) which is calculated as (ref 1.26):

$$d = 4 L F / \pi D \quad \text{(F is the lens focal length)}$$

or $$d = 4 L F\# / \pi \quad \text{(F\# is the effective F-number)}$$

The effect of the spatial power profile is that a low order mode (e.g Gaussian or near Gaussian) results in a smaller beam divergence (ref 1.1), and smaller spot size i.e higher power intensity (ref 1.13).

The effect of laser wavelength is apparent from the relations above, i.e a shorter wavelength produces a smaller spot size and higher power intensity. Therefore, higher beam intensities are produced by YAG or Glass lasers (1.06 micorn) than from CO_2 lasers (10.6 microns) of equivalent power.

Laser beams are normally linearly or randomly polarised (ref´s 1.27 and 1.28). Higher cutting speed and better cut quality (narrower and smoother cut) can be achieved when cutting in a direction parallel to the plane of polarisation of the incident beam (ref´s 1.29, 1.30 and 1.31). This phenomenon causes inconsistent results in contour cutting (i.e arbitrary cutting directions), therefore a conversion to circular polarisation is often performed by a 1/4 wave phase retarding mirror set within the beam delivery system (ref 1.28 and 1.29).

1.4.2 EFFECTS OF BEAM DELIVERY COMPONENTS

The beam delivery system is a system of deflectors (mirrors or prisms) used to steer the laser beam from the laser cavity to a focusing lens, and then to a high intensity spot on the material surface. High reflectivity deflectors are required to avoid losses in beam power by absorption, and to minimise beam distortion. This is achieved by careful selection of deflector material and treatment of their surfaces by coating and polishing.

The focusing lens is an important component for laser material processing in general, and for laser cutting in particular. The factors affecting the choice of a focusing lens for laser cutting are:

a) Lens focal length, or more specifically the F-number (as indicated previously), the effects of this characteristic is shown in fig 1.5. A shorter focal length normally results in a smaller spot size and, consequently, a higher beam intensity. However, a smaller depth of field also results, this causes the cutting process to be less tolerant to variations in optical stand-off distance. Reproducing performance and quality become more difficult (see ref 1.13 for focus spot positioning effects). Another problem of small F-number, is that spherical aberration becomes dominant and may produce a larger spot size than predicted (ref 1.1).

b) Lens geometry, as this affect the amount of aberration and the sensitivity to beam misalignment.

c) The material of the lens, as materials vary differently in their reflectivity, absorptivity and transmissivity with wavelength (ref 1.2). For example, a lens material suitable for YAG laser beam might not be suitable for CO_2 beam, due to high absorption of the CO_2 laser wavelength.

1.4.3 EFFECTS OF WORKPIECE MATERIAL

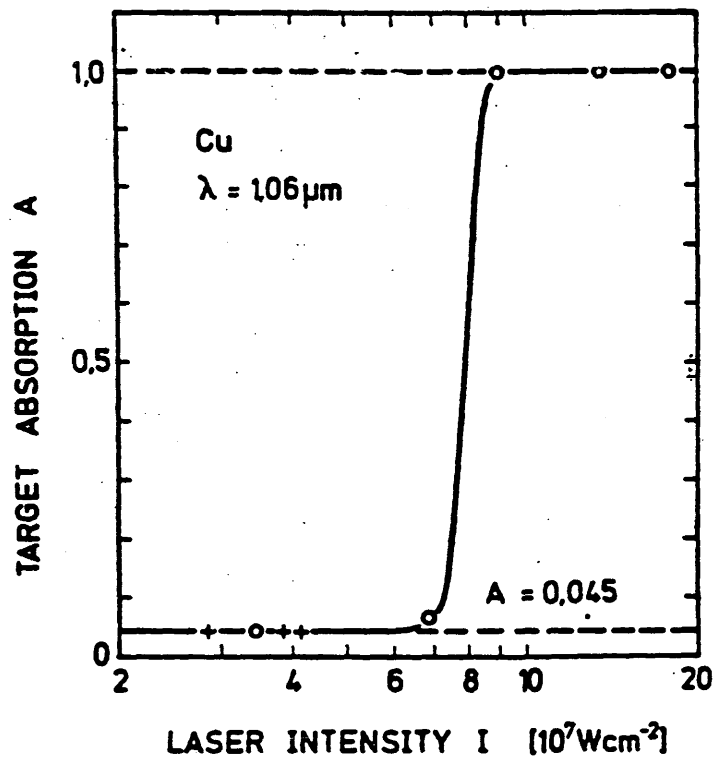
In addition to material thickness, the main properties which affect the laser cutting process, are:

- Absorptivity (1 - (reflectivity+transmissivity)).
- Thermal diffusivity.
- Thermal conductivity.
- Latent heat of fusion and evaporation.
- Heat capacity.

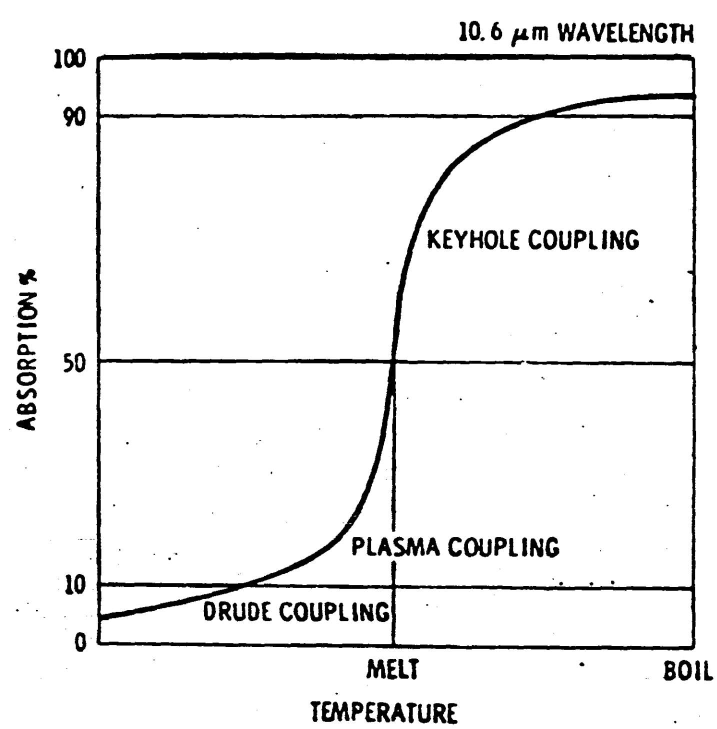
In general, these properties determine the required laser beam power intensity for cutting, and the mechanism of material removal (section 1.3).

Reflectivity is a function of the material structure and surface roughness. Absorption increases with material temperature (which is a function of interaction time and power intensity, fig 1.6). Fig 1.7 presents reflectivity and absorptivity of some metallic and non-metallic materials. In general, metallic materials have considerably higher reflectivity to the CO_2 wavelength, while non-metals are more absorbent to this beam but become difficult or impossible to cut with a YAG beam.

Thermal diffusivity is the speed of dissipiation of the energy absorbed through the material. Diffusivity is a function of the material thermal conductivity, density and

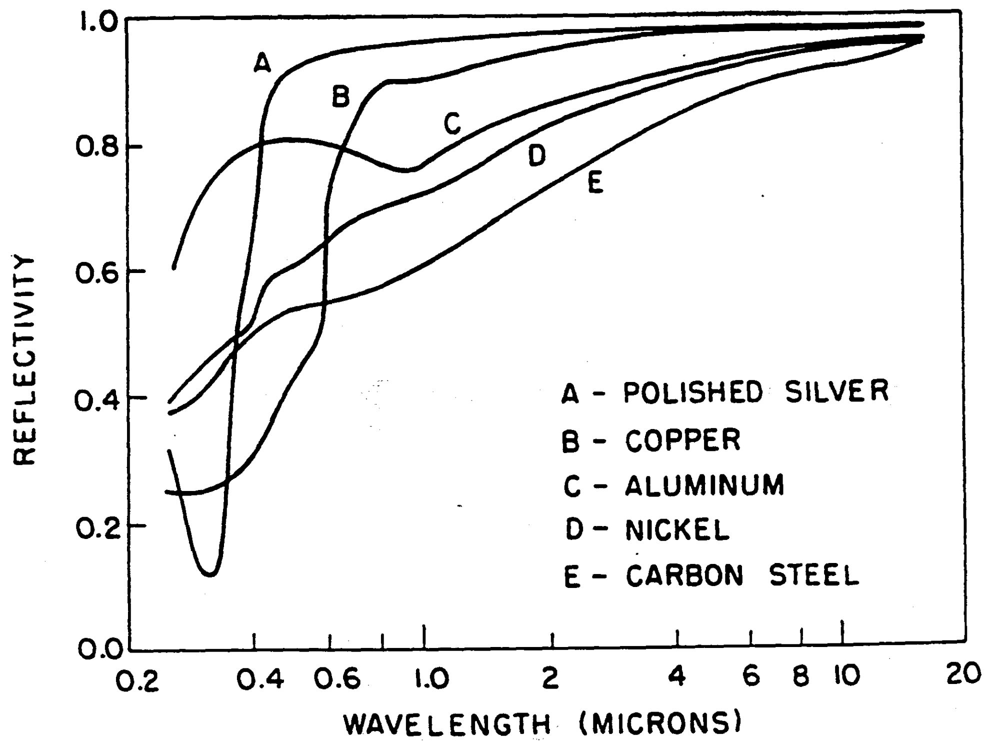


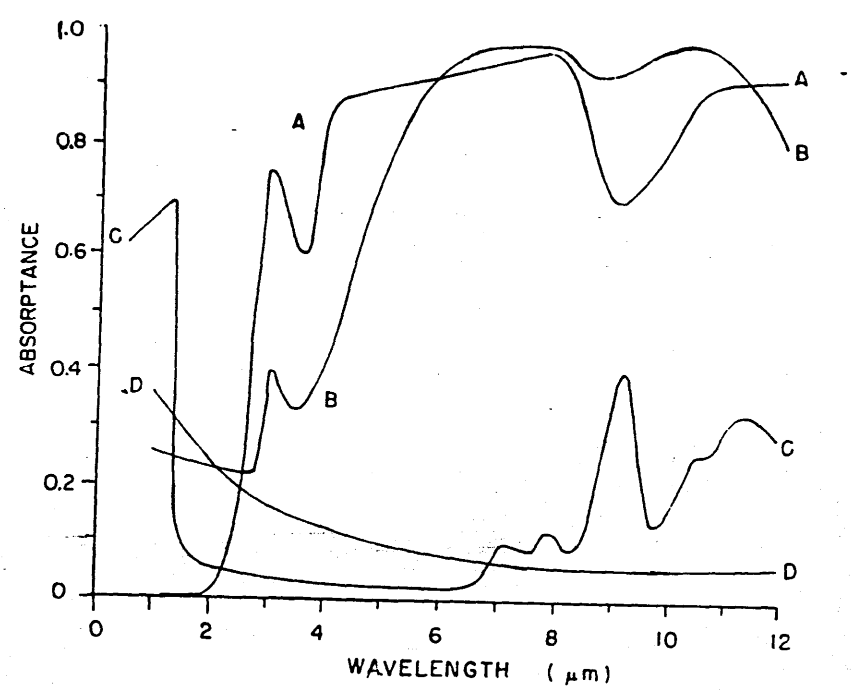
(After Herziger ref 1.19)



(After Steen & Kamalu ref 1.13)

FIG 1.6 : VARIATION OF ABSORPTIVITY WITH TEMPERATURE AND LASER POWER INTENSITY





(A) Pyrex glass .125 in. thick, (B) Alumina massive
(C) Silicon p-type, 1.68 mm thick, (D) Iron massive

FIG 1.7 : EFFECT OF LASER WAVELEGTH ON METERIALS REFLECTION AND ABSORPTION

(After Ready ref 1.1)

the specific heat (see section 1.3). With many materials, diffusivity decreases at higher temperatures (see appendicies A, B and C of ref 1.2), which can considerably aid the cutting process, as heat losses by conduction are reduced.

The latent heat of fusion and evaporation determine the amount of material removed from the cut slot as vapour or melt, in line with the amount of energy absorbed (ref 1.13).

1.4.4 NOZZLE AND GAS FLOW EFFECTS

The basic role of the cutting assistant gas was discussed in section 1.3, this can be summarised as:

- Ejection of the material vapour and molten particles from the cut kerf, influenced by the gas stream momentum, and by the melt properties such as viscosity, density and cohesion.

- A reactive gas contributes to the cutting energy due to exothermic reaction with the (reactive) material at the cut front, influenced by the gas flow rate and material transfer to the reaction front.

These effects are influenced by the following factors:

a) Features of nozzle design, e.g type and diameter.
b) Nozzle stand-off distance.

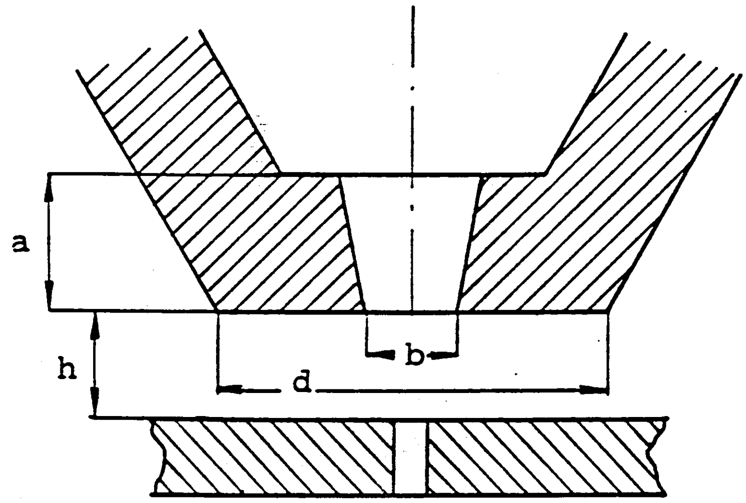
c) Gas pressure.

d) Gas properties, i.e reactivity and density.

These determine the gas flow rate from the nozzle, gas flow and pressure distribution in the gap between the nozzle tip and workpiece surface, and in the cut slot, which affects the cutting performance and quality (ref´s 1.13, 1.31 and 1.32).

A gas nozzle mounted coaxially with the focused laser beam is the most commonly used arrangement, the design features of this type are shown in fig 1.8. Several designs of coaxial nozzle have been developed to improve cutting performance and cut quality (ref´s 1.19, 1.30, 1.31 and 1.33), some of these designs are shown in fig 1.9. The use of one or more auxiliary (secondary) nozzles in various arrangements (fig 1.10) were reported to improve cut quality by increasing the effectiveness of melt ejection and prevention of metal dross attachment to the rear side of cut edges (ref´s 1.34, 1.35 and 1.36).

Supersonic characteristics of gas flow from the nozzle were studied by Ward (ref 1.37), which indicated that undesirable results may be produced by working at this region, due to the generation of shock waves in the gap between nozzle tip and workpiece. This causes large variations in the effective gas pressure on the workpiece surface, which is dependent on the nozlle design, orifice



a : Nozzle thickness

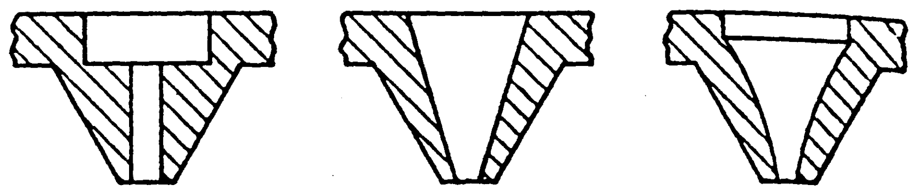
b : Nozzle aperture

d : Nozzle tip outer diameter

h : Nozzle stand-off

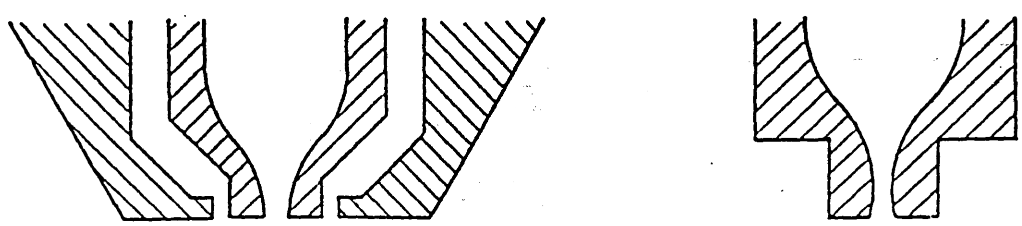
FIG 1.8 : DESIGN FEATURES OF LASER CUTTING GAS NOZZLE

(After Olsen ref 1.31)



PARALLEL. CONICAL. CONVERGENT.

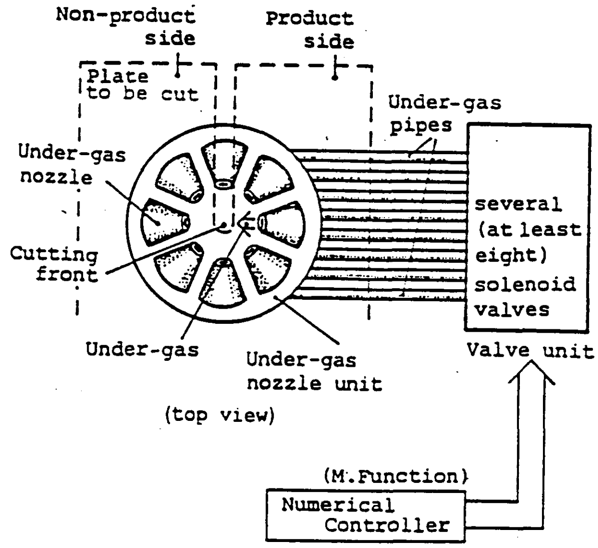
(After Thomassen & Olsen ref 1.30)



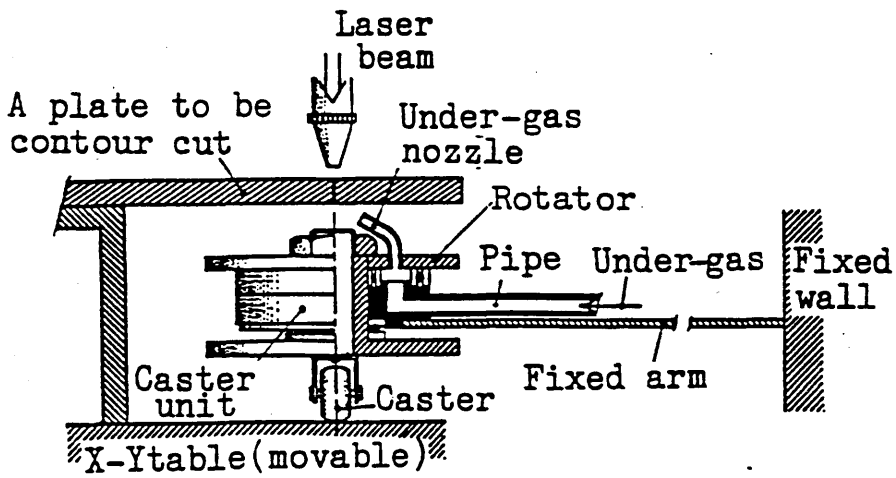
RING CONVERGENT-DIVERGENT

(After Forbes ref 1.20)

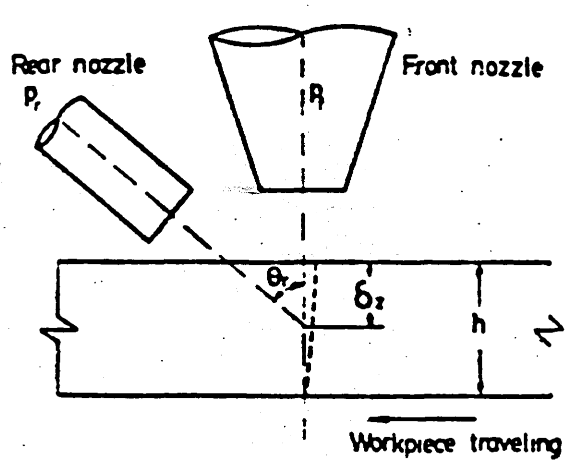
FIG 1.9 : SAMPLES OF NOZZLE DESIGN



(After Birkett et al ref 1.34)



(After Murakawa et al ref 1.35)



(After Arata et al ref 1.36)

FIG 1.10: AUXILIARY NOZLES ARRANGEMENTS FOR PREVENTION OF DROSS ATTACHMENT diameter and stand-off distance between nozzle tip and workpiece. The gas flow is transformed from subsonic to supersonic at a critical ratio of nozzle pressure to ambient pressure. This ratio is dependent on the ratio of specific heats (C_p/C_v) of the gas used, and is not affected by nozzle geometry (ref 1.38).

Further details as to the effects of nozzle orifice diameter and nozzle stand-off distance on the gas flow can be found in ref´s 1.13 and 1.31.

1.5 QUALITY FEATURES OF LASER CUTS

Cuts produced by laser beam can be evaluated by examining five features (ref 1.28), these are:

- Cut kerf width.
- Heat affected zone (HAZ).
- Cut surface roughness.
- Dross attachment.
- Cut Taper.

These features are influenced by material thickness, cutting speed and the factors detailed above. Many experimental studies have been performed to determine the effects of laser cutting parameters (factors) on cut quality, among these are: ref´s 1.13, 1.19-1.21, 1.30, 1.31, 1.34-1.36 and 1.39-1.42. These, however, concentrated on finding the optimum cutting conditions for highest productivity (cutting speed) or highest quality for various materials.

Kerf width is, directly, a function of the volume of material removed from the cut slot by vapourisation and melt ejection. Experiments showed that kerf width generally decreases with higher cutting speeds (ref´s 1.13, 1.39 and 1.43), but behaves erratically with variations in gas pressure and material thickness due to variations in gas flow.

The heat affected zone is a volume of material adjacent to the kerf, in which metallurgical changes have occured due to the temperature rise caused by heat conducted from the cut kerf, which causes alterations to the microstructure and hardness. Kovalenko et al (ref 1.39) experimented with a number of metals and indicated that the width of this zone generally decreases with cutting speed.

Laser cut edges usually show striated surface structures, caused by the way in which material is removed from the cut slot. The geometry of these striations determine cut roughness. The mechanism of formation of this striation when cutting with CW laser was studied extensively by Arata et al (ref 1.20).

Powell et al (ref 1.41) investigated a method of improving cut roughness, i.e producing finer striation, by utilising a pulsed laser emitting at a pulse repetition frequency (PRF) of twice the striation frequency produced by CW laser at the same cutting speed. They also reported that when cutting with a high PRF, the cutting mechanism is similar to that of a CW laser, due to short pulse duration and low pulse energy which is insufficient for complete metal removal (hole drilling). At low PRF, the striation frequency is determined by the pulse frequency.

Dross attachment is a prominant problem in laser cutting of some metals such as titanium, stainless steel, nickel and aluminium. This is caused by high melt viscosity and high cohesive force between molten and solid metal in the cut kerf, which cannot be overcome by the momentum of the gas jet. Methods based, mainly, on using multi-nozzle arrangements were reported to be successful in rectifying this effect (see fig 1.10).

Birkett et al (ref 1.34) designed a numerically controlled under-gas nozzle unit consisting of eight nozzles, where at any instance only one nozzle is operating in such a way as to shift the molten metal to the scrap side of the material. Murakawa et al (ref 1.35) used a single nozzle under-gas unit that is based on the operating principle above. This unit includes a nozzle mounted on a rotating caster unit that follows the movement of the X-Y table.

Both of these methods are limited in their applicability to moving (flat) work table only.

Arata et al (ref 1.36) used a different method for cutting stainless steel, that is by tandem-nozzle cutting. A secondary nozzle is mounted (above the work surface) at an angle behind the main nozzle to create an extra gas momentum for more effective ejection of the molten metal. This method can be used with moving optics cutting systems.

These authors have also proposed a pile-cutting method for dross prevention, where a mild steel sheet is piled on the stainless steel workpiece, and cut together. This works by increasing the rate of oxidisation in the cut kerf due to the additional amounts of Iron (from mild steel), and consequently increasing the fluidity of the melt which can then be ejected more easily.

However, during the cutting experiments of the present work, it was found that when cutting stainless steel with the CO_2 laser, at 25% of the maximum achievable cutting speed, that all the dross is transformed to readily removable oxides, leaving a clean cut.

REFERENCES

REF. No.	DETAILS
1.1	Ready, J. F., Industrial Applications of Lasers, Academic Press (London), 1978.
1.2	Duley, W. W., CO2 lasers, Effects and Applications, Academic Press (London), 1976.
1.3	Ruselowski, J. M. and Barrow, J., Production Laser Processing in Automotive and Aerospace Industries, 4th Conference on Lasers in Manufacturing (LIM-4), May 1987, PP83.
1.4	Williamson, J. R., Summary of Air Force Program in Laser Cutting, SPIE Proceedings, Vol.86, Industrial Applications of High Power Laser Technology, 1976, PP40.
1.5	Weedon, T. M., The Design of a CNC laser Drilling Machine for the Production of High Quality Holes in Aerospace Components, MTIRA Conference, The Machine Tool Industry Research Association, 23 April 1985.
1.6	Gukelberger, A., Industrial Applications of High

Power CO2 Lasers - System Description, SPIE Vol.650, High Power Lasers and Their Industrial Applications, 1986, PP254.

1.7 Nilsson, K. and Sarady, I., Cutting and Welding 3-Dimensional Sheet metal Parts with High Accuracy, Proceedings of LIM-1, 1-3 Nov. 1983, PP79.

1.8 Watson, H. E., The Application of Intelligent Robotic Systems and Lasers for Manufacturing, LIM-4, May 1987, PP153.

1.9 Sibayama, K., Kubo, M. and Itani, K., Sensory characteristics of the "Melcut-3DCM", Conference on Laser Welding, Machining and material Processing, 11-14 Nov. 1985, San Francisco, PP113.

1.10 Borgstrom, R., Grankvist, M. and Rask, K., A Flexible System for Research on Laser Processing, LIM-4, May 1987, PP177.

1.11 Williams, S. W. and Davidson, D., Flexible Laser Machining Workstation, LIM-4, May 1987, PP119.

1.12 Ikeda, M., Design of FMS with Laser, ICALEO'84 Conference, Vol.44, Laser Institute of America, 12-15 Nov. 1984, PP226.

1.13 Bass, M., Laser Material Processing, North-Holland Publishing Company, 1983.

1.14 Thorn, A. R., Hardware for laser Cutting, MTIRA Conference, The machine Tool Industry Research Association, 23 April 1985.

1.15 Belforte, D. A., Robotic Manipulation for Laser Processing, SPIE Vol.650, High Power lasers and Their Industrial Applications, 1986, PP262.

1.16 Bakowsky, L., Material Processing with High Power CO2 lasers, SPIE Vol.650, High Power lasers and Their Industrial Applications, 1986, PP285.

1.17 Weil, B. S., A Laser Cutting system for Nuclear Fuel Disassembly, Conference on Laser Welding, Machining and material processing, 11-14 Nov.1985, San Francisco, PP145.

1.18 Herziger, G., Basic Elements of Laser Material Processing, SPIE Conference, Vol.455, 1983, PP66.

1.19 Forbes, N., The role of the Gas Nozzle in Metal-Cutting with CO2 Lasers, Laser-75, Optoelectronic Conference proceedings, 24-27 Jun.1975, PP93.

1.20 Arata, Y., Maruo, H., Miyamoto, I. and Takeuchi, S., Dynamic Behavior in Laser gas Cutting of Mild Steel, Transactions of JWRI, Vol.8, No.2, 1979, PP15.

1.21 Caprino, G. and Tagliaferri, V., Maximum Cutting Speed in Laser Cutting of Fiber Reinforced Plastics, Int. J. Mach. Tools Manufact., Vol.28, No.4, 1988, PP389.

1.22 Sepold, G., Teske, K. and Rothe, R., Investigations on Laser Melt Cutting of metals, Colloquium on Thermal Cutting and Flame Processes, 7 Sep.1982, Ljubljana, Yugoslavia.

1.23 Bunting, K. A. and Cornfield, G., Toward a General Theory of Cutting: A Relationship Between the Incident Power and the Cut speed, Transaction of the ASME, Journal of Heat Tranfer, Vol.97, No.1, Feb.1975, PP116.

1.24 Schuocker, D. and Abel, W., Material Removal Mechanism of Laser Cutting, SPIE Conference, Vol.455, 1983, PP88.

1.25 Powell, J., Ellis, G., Menzies, I. A. and Scheyvaerts, P. F., CO2 Laser Cutting of Non-metallic Materials, LIM-4, May 1987, Birmingham, PP69.

1.26 Kogelnik, H., Imaging of Optical Modes - Resonators with Internal Lenses, The Bell System Journal, Vol.44, March 1965, PP455.

1.27 Luxon, J. T., Basic Optics for Laser Applications, Lasers & Applications, June 1984, PP97.

1.28 Darchuk, J. M., and Migliore, L. R., Guidelines for Laser Cutting, Lasers & Applications, Sep.1985, PP91.

1.29 Lepore, M., Dell´Erba, M., Esposito, C., Daurelio, C. and Cingolani, A., A quantitative Theory for the Role of Oxygen in the Laser Cutting Process, ICALEO´83 Proceedings, Vol.38 - Material Processing, Nov.14-17 1983, Los Angeles, PP160.

1.30 Thomassen, F. B. and Olsen, F. O., Experimental Studies in Nozzle Design for Laser Cutting, LIM-1, 1-3 Nov.1985, Brighton, PP169.

1.31 Olsen, F. O., Investigation in Optimizing the Laser Cutting Process, ASM Conference on Lasers in Material processing, Jan. 1983, Los Angeles, PP64.

1.32 Sepold, G. and Rothe, R., Laser Beam Cutting of Thick Steel, ICALEO´83 Proceedings, Vol.38, 14-17 Nov. 1983, Los Angeles, PP156.

1.33 Forbes, N., Application of 500 Watt CO2 Lasers in Industry, Laser-78 Conference, 9-10 Mar.1978, London.

1.34 Birkett, F. N., Herbert, D. P. and POwell, J., Prevention of Dross Attachment During Laser Cutting, Proceedings of the Second International Conference on Lasers in Manufacturing, 26-28 Nov. 1985, Birmingham, PP63.

1.35 Murakawa, M., Miyazawa, H., Sakia, S. and Ohkawa, K., Prevention of Dross Attachment in Laser Cutting, Bulletin of Japan Society of Precision Engineering, Vol.22, No.1, March 1988, PP6.

1.36 Arata, Y., Maruo, H., Miyamoto, I. and Takeuchi S., Improvement of Cut Quality in Laser-Gas-Cutting Stainless Steel, Source Book on Applications of Lasers in Manufacturing, American Society for Materials (ASM) 1981, PP72.

1.37 Ward, B. A., Supersonic Characteristics of Nozzles Used with Lasers for Cutting, ICALEO'84 Proceedings, Vol.44, Laser Institute of America, 12-15 Nov.1984, Boston, PP94.

1.38 van Dijk, M. H. H. and Brouwer, E., Reactive Gas-Assisted Cutting with Pulsed Nd-YAG Lasers, Conference on Laser Welding, Machining and Material Processing, 11-14 Nov.1985, San Francisco, PP153.

1.39 Kovalenko, V. S., Arata, Y., Maruo, H. and

Miyamoto, I., Experimental Study of Cutting Different Materials with a 1.5 KW CO2 laser, Transaction of JWRI, Vol.7, No.2, 1978, PP101.

1.40 Berloffa, E. H. and Witzman, J., Laser Materials Cutting and Related Phenomena, SPIE Conference, Vol.455, 1983, PP96.

1.41 Powell, J., King, T. G. and Menzies, I. A., Cut Edge Improvement by Laser Pulsing, LIM-2 Proceedings, 26-28 Nov. 1985, Birmingham, PP37.

1.42 Lee, C. S., Goel, A. and Osada, H., Parametric Studies of Pulsed-Laser Cutting of Thin Metal Plates, ICALEO´84 Proceedings, Vol.44, 12-15 Nov. 1984, PP86.

1.43 Dillio, A., Dionoro, G., Minutolo, F. and Tagliaferri, V., Laser Cutting of Al7075 Sheets, LIM-2 Proceedings, 26-28 mar. 1985, PP57.

CHAPTER 2

TASKING AND CONTROL ENVIRONMENT FOR LASER BASED MANUFACTURING SYSTEMS

This chapter outlines a system for planning and controlling laser based manufacturing operation, proposed by the LOSEG group (Laser and Optical Systems Engineering Group) in the Department of Mechanical Engineering at the University of Glasgow, and the stages implemented by the author.

2.1 BACKGROUND

The current and potential applications of lasers in manufacturing, indicates the flexibility of lasers as manufacturing machine tools, when enhanced by the use of CNC, multi-axis laser beam, and workpiece handling technology. It has been argued that laser system flexibility is an important feature for trial (one off) and small batch production (ref´s 2.1, 2.2 and 2.3) in a wide variety of industrial applications. This is especially so for sheet metal fabrication which is used in: automotive, aerospace, chemical, food, petroleum and marine industries.

To achieve productivity, quality and cost effectiveness in trial and small batch production, Kay and Robertson (ref 2.4) identified three requirements:

- The need for speed and accuracy at all stages from design to delivery.

- The need for true flexibility, not just the ability to reprogram.

- The need for system intelligence or at least the ability to escape from reliance on human judgement and expertise.

It is envisaged that these requirements can be achieved in laser based manufacture by the implementation of the following:

1) Efficient generation of component design and part programs, achieved by utlising computer aided design and manufacture packages (CAD/CAM). This is particularly important at the prototype (trial) fabrication stage as a part may need to be redesigned and manufactured several times before deciding on a firm configuration.

2) Knowledge of the manufacturing process parameters required and their control characteristics, against: various types and thicknesses of engineering materials, and differing requirements of quality and speed of manufacture. This knowledge should be available at a stage prior to part fabrication, i.e at the design stage or a separate planning stage.

3) Adaptive, real-time (in-process) control of the manufacturing process according to a control protocol created at the planning stage to achieve the desired effects (speed and quality), using various types of sensors for process monitoring.

4) Beam and workpiece handling facilities that allow for a wide range of component shapes and sizes to be processed.

5) Evaluation of the quality produced by the manufacturing process by using either in-process or post-operation measurement techniques, automated whenever possible to reduce operator error. This information being recorded and fed back to the planning level, where it can be used to determine the utility of the existing planning and control system and for more accurate prediction of future manufacturing processes, i.e a self monitoring and learning closed loop control system.

It is believed that the process of planning and deriving a control strategy for laser based cutting processes, should be based on the knowledge gained from previous cutting experience which can be analysed and recorded (in a database), rather than on a deterministic approach using derived mathematical models.

This is due to the inadequacies of existing models in describing the slitting process which involves a vast number of interacting parameters (see chapter 1). This leads to a wide variance between the predicted and actual performance of the cutting process. Consequently, knowledge of the results and the parameters used for the process (empirically) rather than the process itself is needed (ref 2.5). This can be achieved by learning from previous performance.

The implementation of the above would give two desirable characteristics to laser based manufacture, these are:

- Product flexibility, as products of the required design, size, material and quantity, can be manufactured on demand.

- Process flexibility, as various, adaptively controlled manufacturing processes can be performed.

These two characteristics are the fundamental requirements of flexible manufacturing systems (FMS).

There have been a number of attempts to control the laser cutting process, all of them based on real-time control of some of the process variables.

Foulloy et al (ref 2.6) developed a rule based system that controlled the laser power and cutting speed according to the results of visually evaluating the spatter cone beneath the workpiece (generated by the process of ejecting the molten metal particles). The geometrical characteristics of this cone are monitored by a TV-camera (fig 2.1), and a control action is taken according to previously coded set of rules against the results of evaluation.

This method is, however, limited by the following:

- This method can only be applied when the work is flat, the laser beam is stationary and cutting is in one direction. It would be difficult (if not impossible) to use this system in other configurations (e.g flying optics, multi-axis work handling), as the TV-camera must be underneath the workpiece and would require a separate handling system.

- A specific set of rules being required for each type of material.

- This method is confined to metallic materials where a spatter cone is produced.

Moriyasu et al (ref 2.2) developed a control system that set laser power and switched between continuous and pulsed modes of a CO2 laser, according to a predefined set of instructions (specific to a particular material), depending on the actual measured traverse speed of the work handling table.

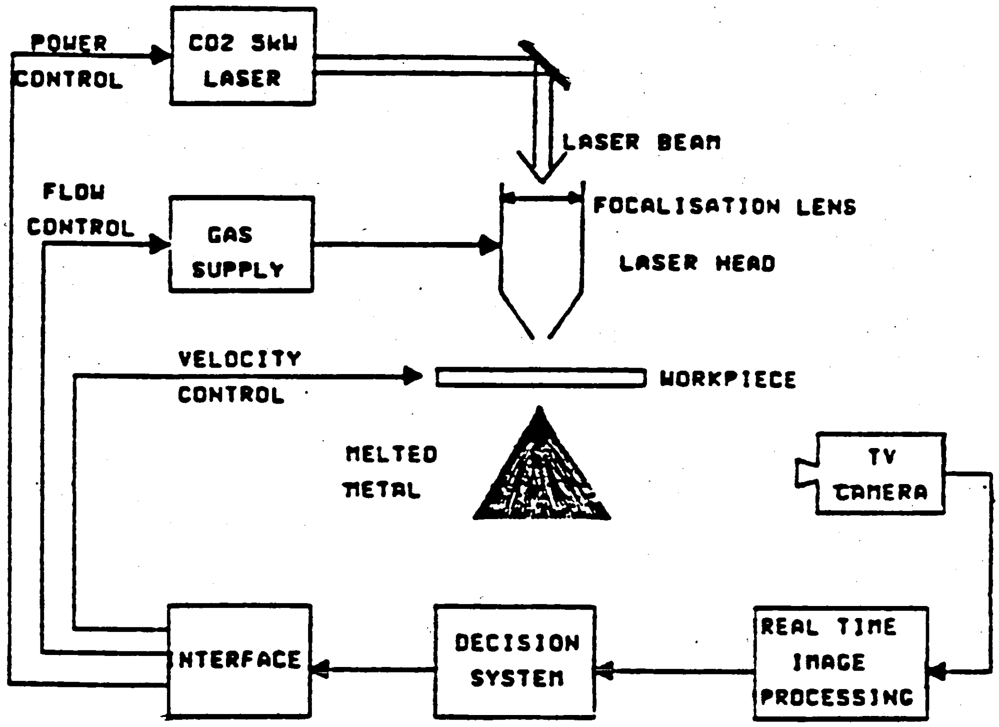


FIG 2.1 : EXPERIMENTAL ARRANGEMENT OF A RULE BASED CONTROL SYSTEM

(After Foulloy et al, ref 2.6)

Other developments have included the monitoring of the incident laser beam (e.g power and intensity profile, ref´s 2.7, 2.8 and 2.9), positional control in laser welding (ref 2.10) and nozzle stand-off or height control (ref 2.11).

2.2 OUTLINE OF THE PROPOSED TASKING AND CONTROL SYSTEM

The concepts and requirements discussed in the previous section were recognized, and a modular, integrated laser based system to implement these, was proposed.

The basic approach is the development of adaptive control techniques to meet the demand for dynamic adjustments required during the cutting process, and a computational system for recording, managing and applying experience gathered from past cutting processes (short and long term feedback control).

Fig 2.2 outlines the proposed system, which basically consists of three distinct levels of operation:

- Design.
- Planning and recording.
- Real-time control.

Fig 2.3 is a detailed presentation of these levels.

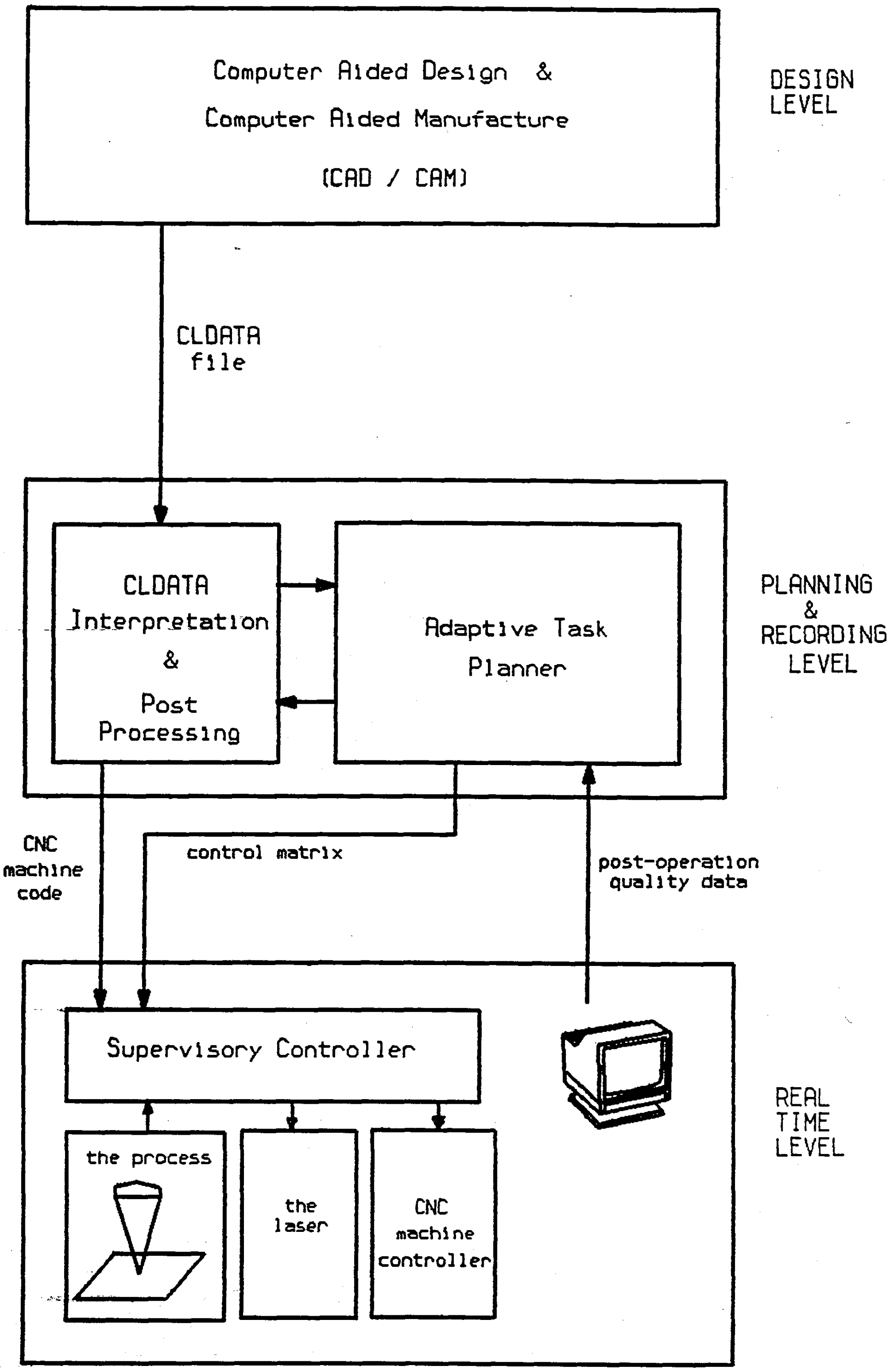


FIG 2.2 : OUTLINE OF THE PROPOSED TASKING AND CONTROL ENVIRONMENT FOR LASER BASED MANUFACTURING SYSTEMS

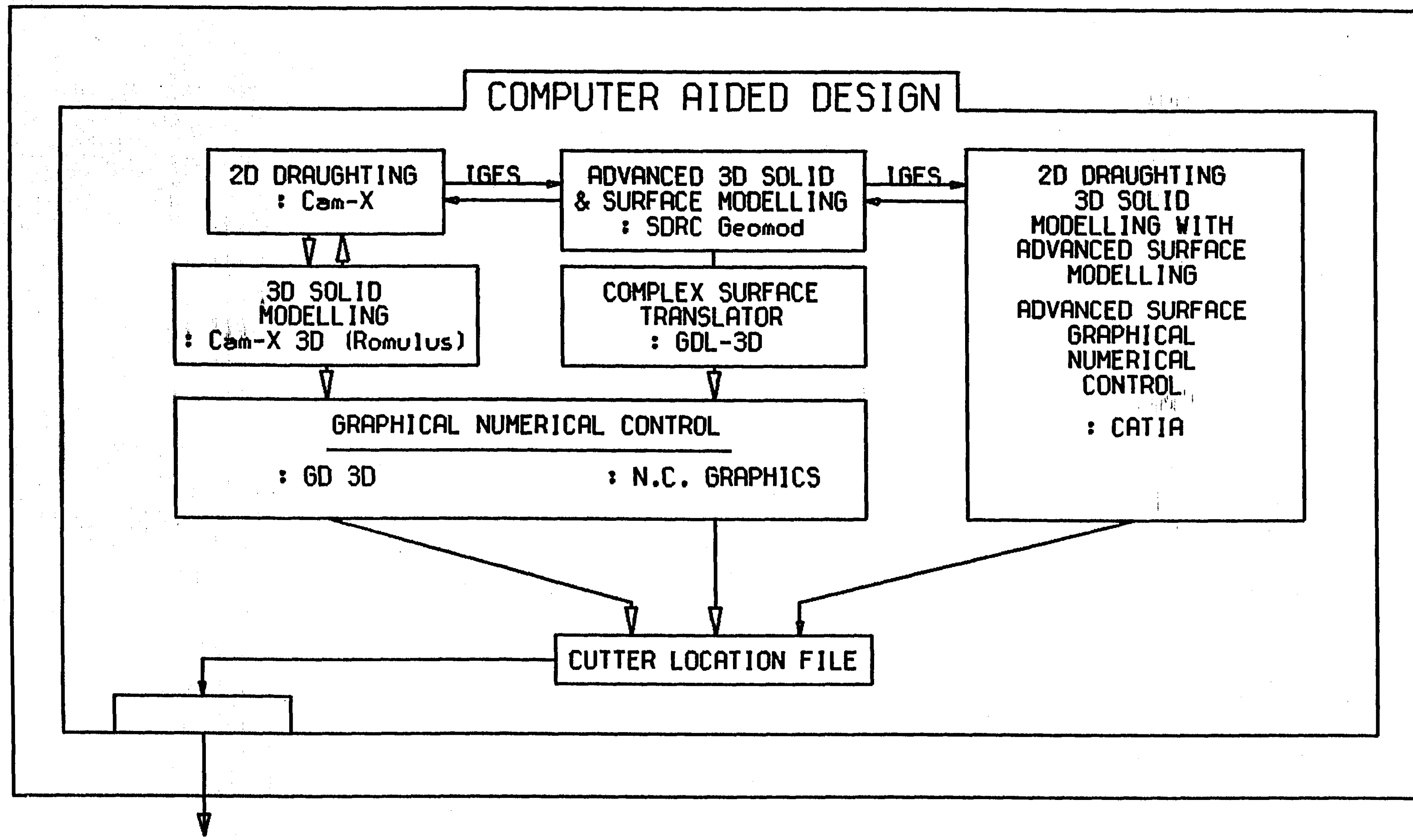
DESIGN

Commercially available computer aided design (CAD) software can be used to generate both 2 and 3-dimensional geometrical models of the required component. The geometry is then used by a computer aided manufacturing (CAM) package (e.g Graphical Numerical Control-GNC) to generate a CLDATA file containing the cutter location data (cutter path) and tool information, this file is passed to the next level of operation.

PLANNING AND RECORDING

The CLDATA file sent from the design stage is decoded into a man readable format by a computer program (CLDATA translator), which can then be used to input information as to the cutter path geometry, desired quality and method, to the Task Planning Executive and Control Strategy Generation module.

This module modifies the input file by sorting necessary process parameters and cutter path alterations (e.g forcing a loop around sharp corners). The modified file is then re-encoded into a new CLDATA file and post-processed by a generic or specific (to a machine tool) post-processor to produce CNC code compatible with the CNC controller of the target work handling machine.



DESIG

FIG 2.3 : DETAILED PRESENTATION OF THE PROPOSED TASKING AND CONTROL ENVIRONMENT FOR LASER BASED MANUFACTURING SYSTEMS

(Courtesy of Mr. F. D. Buchan, Dpt. of Mechanical Eng., Glasgow University)

The Adaptive Task Planner consists of three main, separate but interacting modules:

- Task planning executive and control strategy generator.

- Relational database management system.

- Performance monitor and self learning controller.

The database module is for the systematic recording of data received from real-time control and post-operation quality evaluation.

The task planning executive and control strategy generation module will be almost wholly algorithmic utilising structured queries to access the database for the acquisition of data necessary for the prediction of process parameters and the generation of a real-time process control strategy (control matrix).

The performance monitor and self learning module compares the post-operation quality measurement data with the data sent by the task planning executive, allowing the adaptive task planner to monitor its own operational performance.

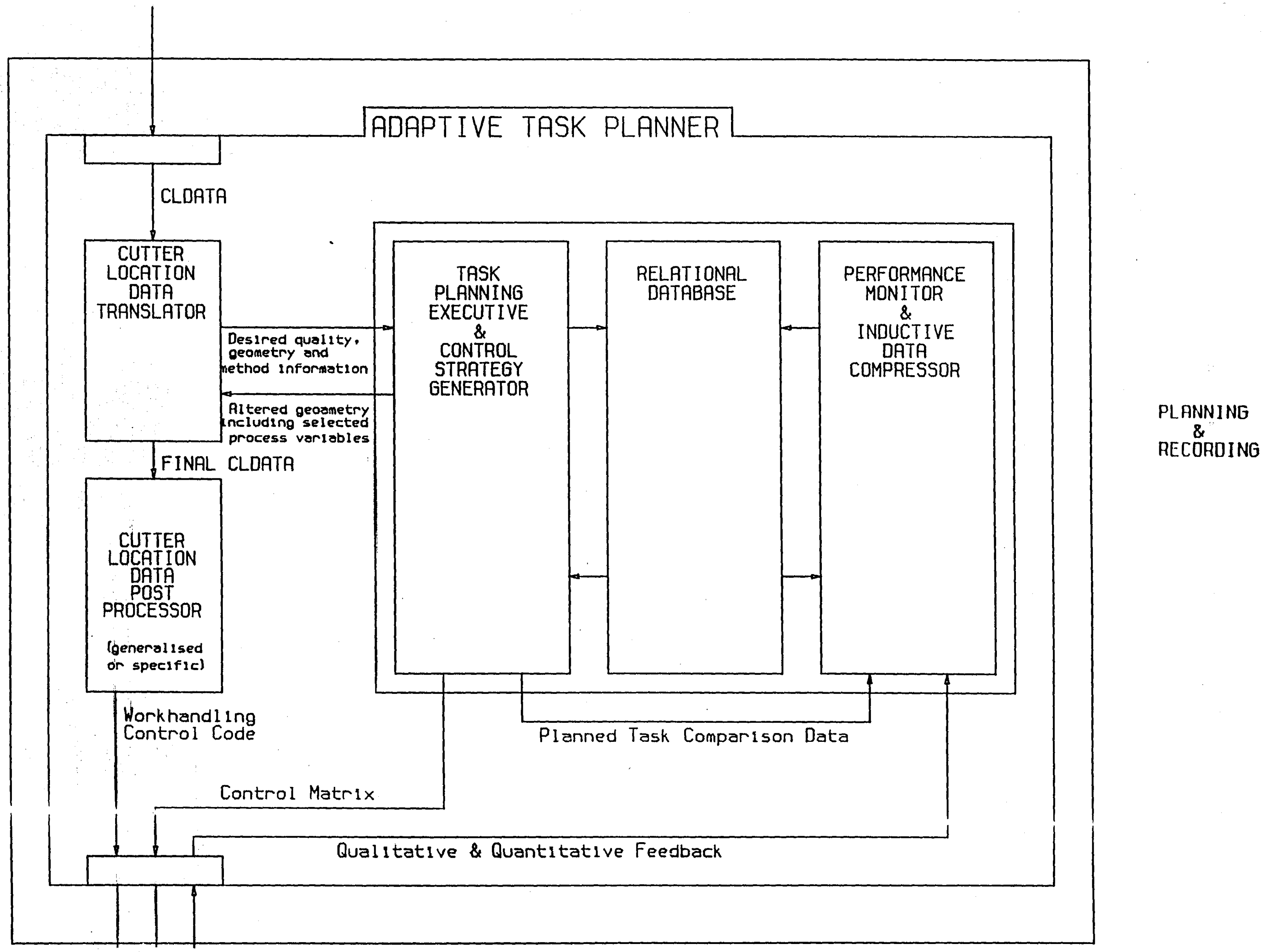


FIG 2.3 : (continued)

REAL-TIME CONTROL

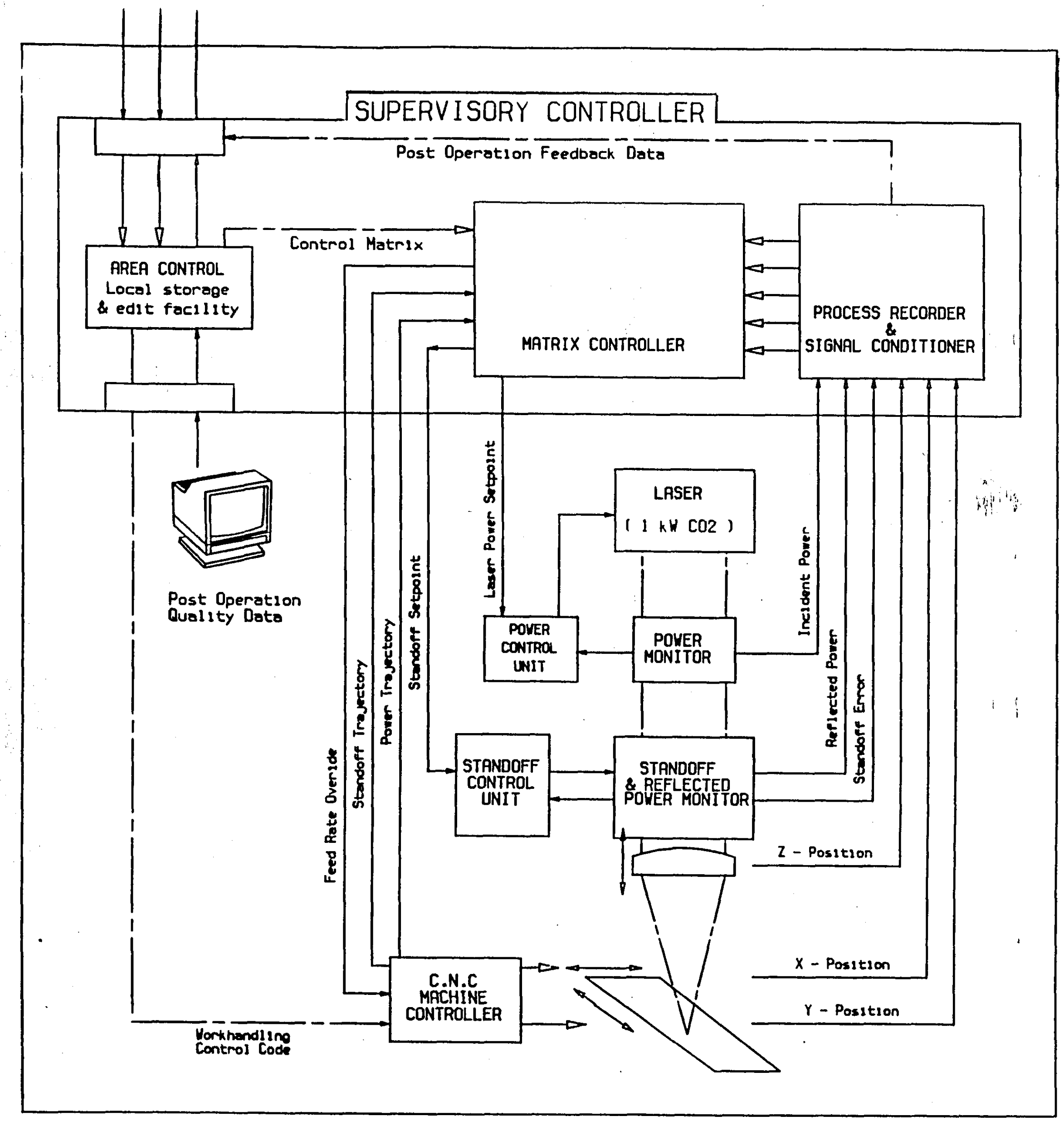
Real-time control of the manufacturing process at the shopfloor level can be categorised as:

a) Area control, which interacts with the adaptive task planner, by downloading CNC part program and control matrix, and uploading post-operation feedback and quality measurement data.

b) Supervisory control, for the supervision of the manufacturing system in real time, using the provided control matrix.

c) Closed loop control to dynamically control individual process parameters, e.g laser power, nozzle stand-off and cutting speed, according to setpoints specified by the supervisory controller.

d) In-process and post-operation quality data recording. In process measurements may include monitoring of: laser beam characteristics (e.g power and intensity profile), material surface temperature and laser power reflected from material surface. Post-operation measurements include: kerf width, cut roughness, dross attachment, heat affected zone and taper measurements.


SUPERVISORY CONTROLLER
Post Operation Feedback Data
Control Matrix
AREA CONTROL
Local storage
& edit facility
MATRIX CONTROLLER
PROCESS RECORDER
&
SIGNAL CONDITIONER
Post Operation
Quality Data
Laser Power Setpoint
LASER
(1 kW CO2)
Incident Power
Reflected Power
Standoff Error
POWER CONTROL UNIT
POWER MONITOR
Feed Rate Override
Standoff Trajectory
Power Trajectory
Standoff Setpoint
STANDOFF CONTROL UNIT
STANDOFF
& REFLECTED
POWER MONITOR
Z - Position
X - Position
Y - Position
C.N.C
MACHINE
CONTROLLER
Workhandling
Control Code

REAL-TIME

2.3 ACCOMPLISHED STAGES OF THE PROJECT

An outline of the tasks performed by the author towards the proposed project is shown diagramatically in fig 2.4.

The geometrical model of the component is produced using a 2-dimensional draughting package (CAM-X). The geometry data file is then fed to graphical numerical control software (GNC) for the generation of a cutter path (CLDATA) file, in an interactive session.

A generalised post-processor (C-TAPE) is used to post-process the CLDATA file to produce CNC machine code compatible with the target workhandling system CNC controller. This post-processor is controlled by a program written in a specialised language which allows the translation of the GNC CLDATA file into a format readable by the CNC controller (G and M codes). An example of the interpretation program code used for the HEIDENHAIN controller of the work handling machine is listed in Appendix 2A. All of these tasks were implemented using a MICROVAX II computer system.

A relational database management system (ORACLE) was used for the construction of a database module to record the necessary information about the manufacturing system, the cutting process and post-operation data (chapter 3).

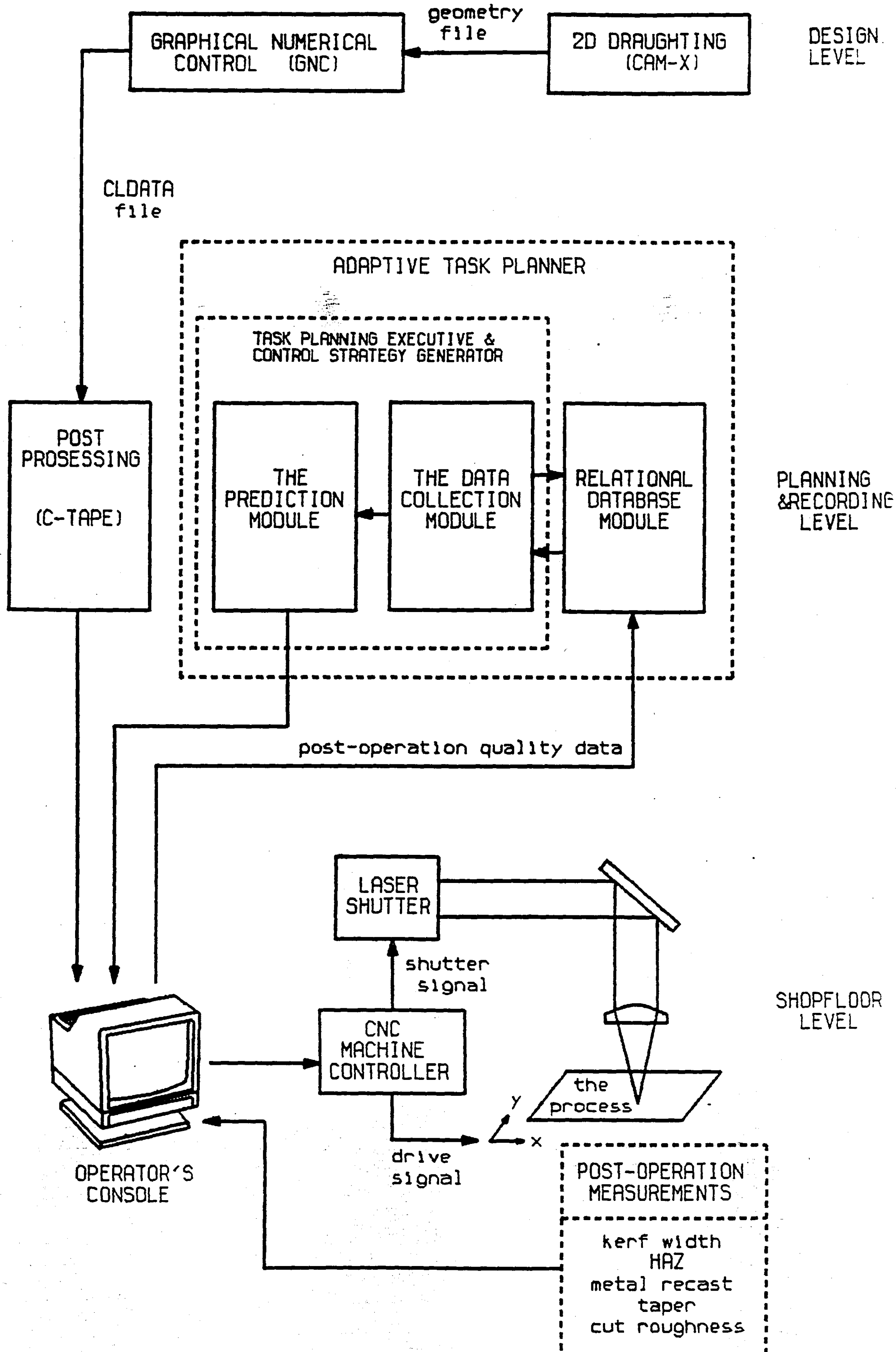


FIG 2.4 : THE IMPLEMENTED STAGES OF THE PROJECT

610 cutting experiments with full post-operation analyses were performed (chapter 4), to provide an initial set of data for the development and testing of the database, data collection and the prediction modules.

The data collection and the prediction modules constitute part of the task planning executive and control strategy generator.

The data collection module (chapter 5) interacts with the database system through using structured queries, to access (collect) the data required for the prediction of process parameters. This module and the database system were implemented using an IBM 6150 computer.

The prediction module (chapter 6) is for the prediction of cutting process parameters and cut quality features, using the data supplied by the collection module. This was implemented on a VAX 11/750 computer system. The predicted values of power, feed and assistant gas pressure, are, currently, manually inserted into the part program at the shopfloor level.

REFERENCES

REF. No.	DETAILS
2.1	Ikeda, M., Design of FMS with Laser, ICALEO´84 Conference, Vol.44, Laser Institute of America, 12-15 Nov. 1984, PP226.
2.2	Moriyasu, M., Hiramoro, S., Hoshinouchi, S. and Ohmine, M., Adaptive Control for High-Speed and High-Quality Laser Cutting, Conference on Laser Welding, Machining and Materials Processing, 11-14 Nov. 1985, San Francisco, PP129.
2.3	Williamson, J. R., Summary of Air Force Program in Laser Cutting, SPIE Proceedings, Vol.86, Industrial Applications of High Power Laser Technology,1976,PP40.
2.4	Kay, N. and Robertson, J., Knowledge Based Engineering in Advanced Automation and Manufacturing, IEE Symposium on Current Trends in the Application of Expert Systems, 2 Apr. 1986, Glasgow.
2.5	Leitch, R., Knowledge Based Process Control, IEE Symposium on Current Trends in the Application of Expert Systems, 2 Apr. 1986, Glasgow.

2.6 Foulloy, L., Kechemair, D., Burg, B., Lamott, E. and Zavidovique, B., A Rule Based Decision System for The Robotization or Metal Laser Cutting, IEEE Conference on Robotics and Automation, March 1985, USA, PP192.

2.7 Weerasinghe, V. M. and Steen, W. M., In-Process Monitoring of Laser Processes, Conference on Laser Welding, Machining and Materials Processing, 11-14 Nov. 1985, San Francisco, PP107.

2.8 Konig, W., Meis, F. U., Willerscheid, H. and Schmitz-Justen, Cl., Process Monitoring of High Power CO2-Lasers in Manufacturing, LIM-2 Proceedings, 26-28 Nov. 1985, Birmingham, PP129.

2.9 Ramos, T. J. and Lingenfeller, A. C., Laser Beam Diagnostics for Metal Working Applications, ICALEO´84 Proceedings, Vol.44, Laser Institute of America, 12-15 Nov.1984, Boston, PP306.

2.10 Watson, H. E., The Application of Intelligent Robotic Systems and Lasers for Manufacturing, LIM-4 Proceedings, May 1987, PP153.

2.11 Sibayama, K., Kubo, M. and Itani K., Sensory Characteristics of the ´MELCUT-3DCM´, Conference on Laser Welding, Machining and Materials Processing, 11-14 Nov. 1985, San Francisco, PP113.

CHAPTER 3

THE LASER CUTTING DATABASE MANAGEMENT SYSTEM

In chapter 2, it was shown that a database management system (DBMS) is an essential module in the proposed Tasking and Control System.

This chapter covers at first, the general aspects of the database management systems. It then discusses the aspects of the database system used for laser cutting, and the method of implementation.

3.1 GENERAL REVIEW

According to C.J.Date (ref 3.1), " A database system is basically a computerised record-keeping system. That is a system whose overall purpose is to maintain information and make that information available on demand". The basic idea of working with a database system is illustrated in fig 3.1. A database, users, computer hardware and computer software are involved in such a system. The system user is given facilities to perform a variety of operations, including:

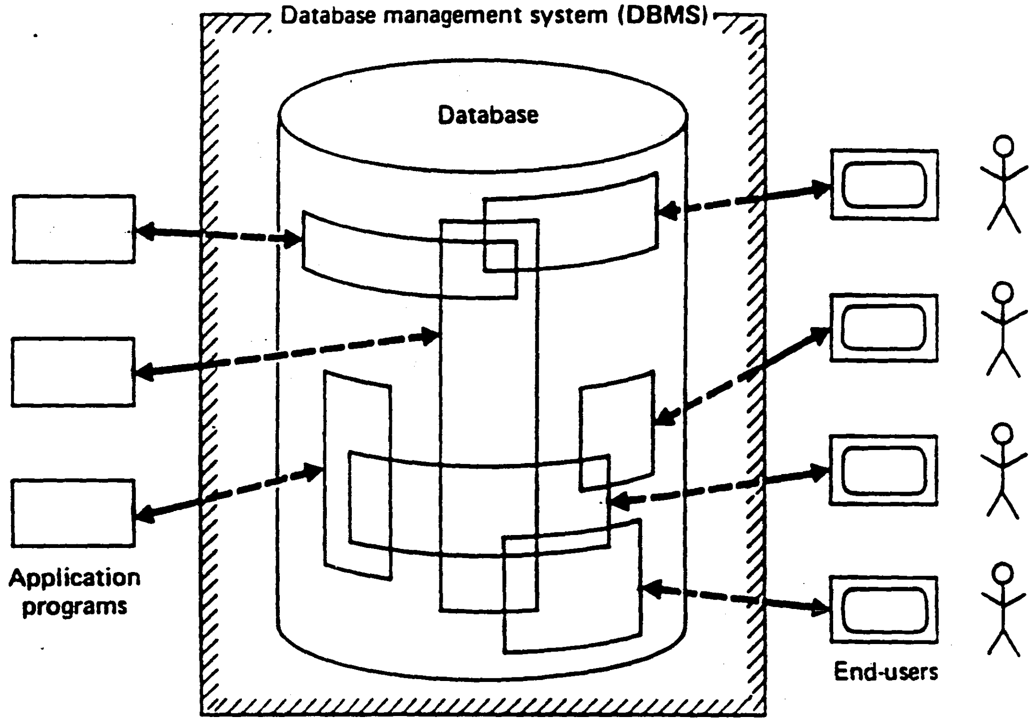


FIG 3.1 : WORKING WITH A DATABASE MANAGEMENT SYSTEM

SOURCE : DATE,C.J., AN INTRODUCTION TO DATABASE MANAGEMENT SYSTEM

- Adding new files to the database (creating files).
- Inserting new data into existing files.
- Retrieving data from existing files.
- Updating data in existing files.
- Deleting data from existing files.
- Removing existing files from the database.

These operations are the basic facilities which every database system must offer. Some systems may offer more facilities such as summation, averaging and text processing.

A database system can be envisaged to comprise two modules: an electronic filing cabinet that stores a collection of data files, and a data management module. The data files are not ordinarily man-readable ones, and are written in the internal language of the host computer system, and structured in such a way to enable data handling by the management module. The management module is responsible for handling of the data files and manipulating the underlying data in harmony with the computer operating system. A simplified picture of the principles of operation is shown in fig 3.2.

A wide range of database products are available for computers ranging from personal to mainframe systems. The facilities, speed of processing and the capacity of every database system, is governed by the hardware and software capabilities of the corresponding host computer. These

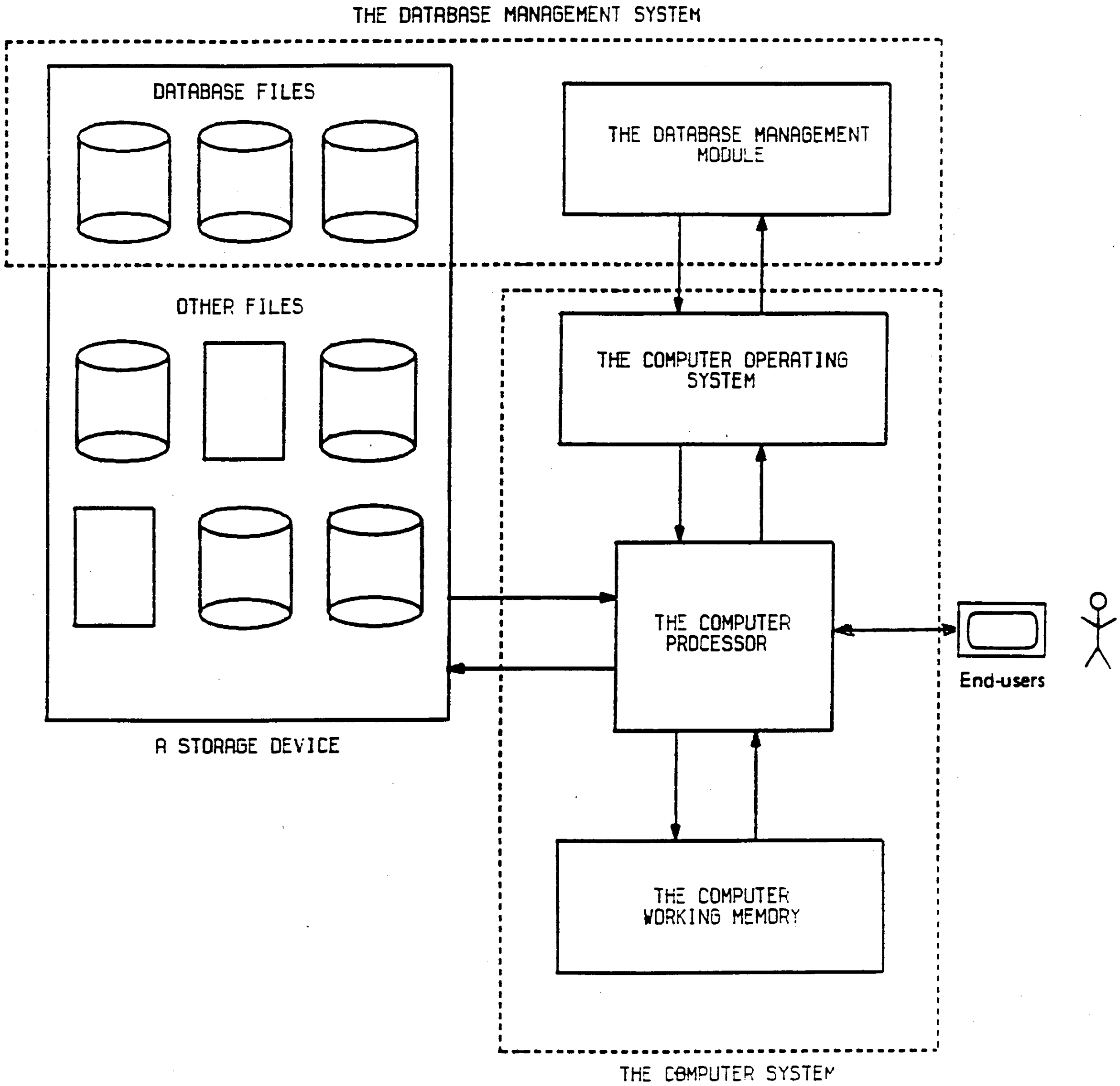


FIG 3.2 : SIMPLIFIED MODEL OF THE PRINCIPLE OF OPERATION OF THE DATABASE MANAGEMENT SYSTEM

include: performance of the microprocessor, performance of the operating system, size of the computer working memory (core) and the capacity and speed of the storage device(s).

3.1.1 ADVANTAGES OF DBMS

Some advantages of using database systems, over conventional filing systems, can be listed:

1) Physical compactness: Using computerised database dismisses the need for large file cabinets and heavy piles of papers. A wide variety of electronic storage devices exist, like: tapes, floppy diskette and built-in (hard) disks. An example of this is that a filing cabinet containing 2500 sheet of paper (4 drawers X 25 files per drawer X 25 sheet per file), assuming 2 KiloByte per sheet, will make a total 5 Megabyte. This suggests that a 20 Megabyte hard disk can replace 4 filing cabinets.

2) Speed: An operator on a terminal can retrieve, alter and manipulate large amounts of information in less time than that required to reach a specific file in a conventional system.

3) Currency: This feature can be considered as a consequence of point 2. That is because fast data processing can make accurate and up-to-date information, available at any time.

4) Consistency and integrity: The enforcement of these features depends on the degree of internal examination of the supplied data, performed by the database system. But it is still easier and faster for the user to ensure data consistency and integrity, using the database system.

5) Centralisation of data: This feature is evident in enterprises using either one mainframe computer, or a computer network system. In either case, the database is a multi-user system. In a mainframe system, the database resides in a central storage device. When a network system is used, the database modules can be distributed on several nodes and recalled from any one node. This characteristic may put in view other subsequent advantages:

- The system provides centralised control of distributed data by authorised personnel.

- The data can be shared by different applications, where each user can look at the data from a different viewpoint.

- Redundancy of data can be reduced: Redundant data can be described as data which is repeated in two or more data files in the various departments of any enterprise. A centralised database can remarkably reduce this redundancy, by allowing all the information to be viewed on one screen. Further enhancement to this theme, comes from a specific type of database system called "relational database". This type relates data in one file to data in other files. Redundant data can be eliminated in this way. The database model in fig 3.1 illustrates this relational concept.

- Security: Access to the database, or parts of it, can be granted to authorised personnel only. This is done by applying proper security checks (e.g password protection).

3.1.2 USER-DATABASE INTERFACE

Interface with a DBMS is done either interactively, i.e directly on a computer terminal, or via an application program (fig 3.1).

Interactive interface with DBMS is made using a special purpose, high level language, called the "data language", specially adapted to the particular DBMS. The user of the DBMS system either uses this language directly by typing the required commands for a specific operation. Or indirectly, by filling out special forms that are displayed by the system, in this method, the responses are interpreted internally into data language commands.

The third type of interface uses an application program written in one of the well known high level programming languages, e.g FORTRAN. Here the data language becomes the "data sublanguage", and the program language is called the "host language". Statements from the data sublanguage are embeded within the host language, to perform the required data operations, while the host language is responsible for the various computational processes of the program.

3.2 RELATIONAL DATABASE SYSTEM (RDBMS)

Database management systems are categorised into: hierarchical, inverted list, network and relational database systems. The relational database is the latest development, and almost all research and development activities in this field, over the past few years, are based on relational ideas (ref 3.1). The increased productivity (5 to 20 times faster than previous approaches), ease of use, the conceptual simplicity of the model and data independence are the major advantages over the other types (ref´s 3.2, 3.3, 3.4).

The basic ideas of the relational model were conceived in the late sixties. In 1970, Dr. Edgar F. Codd of IBM proposed a generalised relational model (ref 3.3). The model was subsequently improved and expanded by Codd himself and became, very quickly, the main focus of research activities on databases.

A relational database is a DBMS composed of relations as building blocks. A relation is a two dimensional table of data, i.e rows and columns. It should not be considered as a matrix, as the contained data are not homogeneous throughout the table, i.e different types of data. Though, the data in each column must be of the same type. Fig 3.3 (below), presents two sample relations, four rows and three columns each. The number of rows in a table called "cardinality", and the term "degree" is given to the number of columns. Hence each of relations in fig 3.3 is of cardinality 4 and degree 3.

MATERIALS

TYPE	THICK	IDNO
copper	22	101
brass	18	104
bronze	27	102
copper	33	103

PARAMS

IDNO	POWER	FEED
103	200	78
101	210	100
104	220	140
102	220	98

FIG 3.3 : TWO SAMPLE TABLES OF RDBMS

Every table (relation) in a relational database system, can be treated independently when performing the data operations listed in section 3.1. But the ability to jointly operate on the data of two or more tables simultaneously, is the main objective of having such a relational system. This type of operation is called a "relational operation", and is governed by "relational algebra". To enable this type of operation, there must be some kind of common identification key between the prospective data in the intended tables, which is unique to that data. This key is, normally, a collection of data entries in one column. The column IDNO in tables MATERIALS and PARAMS is the unique identifier between the data of the two tables. IDNO is called the "primary key" in the table PARAMS, since each entry of this column identifies the corresponding row. While it is called the "foreign key" in the table MATERIALS, because it represents foreign data to this table. The following example is a demonstration to the concepts stated above. It is a relational operation of retrieving data from the two tables in fig 3.3:

```
select    TYPE, IDNO, THICK, FEED
from      MATERIALS, PARAMS
where     THICK >= 20
and       PARAMS.IDNO = MATERIALS.IDNO   ;
```

The response to this operation will be:

TYPE	IDNO	THICK	FEED
copper	101	22	100
bronze	102	27	98
copper	103	33	78

To conclude this section, it is worth noting that the relational model is the most suitable for distributed database systems. This is because the data can readily be partitioned into any number of files. These files can be distributed across several nodes, and rejoined for the construction of the required view (set) of data at the time of query. Also, the criterion of reduced redundancy is enforced by this model, since no data is repeated in the various files.

3.3 A DATABASE SYSTEM FOR THE LASER CUTTING TASKING AND CONTROL ENVIRONMENT

A review of publications relevant to databases and relational databases, and discussions with specialists from the department of Computer Science in the University of Glasgow, led to the conclusion that a relational form of database is the most desirable. This was re-enforced by the investigation as to the data which is required to be accessed. This data was expected to have three general characteristics:

- It included a wide and changing domain of parameters.

- These parameters fell into a number of categories.

- The range of data may expand vigorously, and the expansion of some categories may be different from others.

Detailed explanation of these characteristics is given in section 3.4.

In the course of searching for a RDBMS compatible with the prospective application and with the available computer system (VAX 11/750 running under VMS operating system), two products were selected with the help of ref 3.2 which is a catalogue of full feature analyses of 14 RDBMS´s. These two systems are "INGRES" by Relational Technology, and "ORACLE" by Relational Software Inc. This process was overriden, however, by an offer from IBM UK LTD, which included a personal computer type IBM 6150 with SQL/RT RDBMS on it.

The IBM 6150 is a multi-user, multi-tasking, microcomputer, runs AIX 1.1 operating system, which is an implementation of a UNIX system. A single user may run up to 16 interactive tasks concurrently. A virtual view (screen) is created for each task, on the console display. Switching between these screens is done by pressing two keys only.

3.3.1 SQL/RT DATABASE SYSTEM

This system is based on the ORACLE multi-user database management system, and SQL data language for user interface.

SQL/RT as a relational database is composed of tables. Each table is made of columns and rows. Each column contains one type of data, which is numeric, character or date type. Each row is one record of information, which is made up of a fixed number of fields, one field for each column. Each field can hold one value or item of information (data entry). The records in one table, can be related to records in other tables using the criteria of common identifiers mentioned previously. The identifying elements of the related data must be of the same data type.

Structured Query Language (SQL), also called SEQUEL II, is a relational data language that was first developed at IBM Research Laboratory in San Jose, California (ref 3.1). After a number of usability and performance tests, both inside and outside IBM, the present SQL was developed. This language is intended for non-specialist in data processing as well as for the professional, and is designed to allow easy definition of, and access to databases.

The visible structure of SQL statements, is English-like and propagates through a logical sequence of instructions. The query in section 3.2 presents a sample of this logic.

SQL/RT database can be accessed in the three ways mentioned in section 3.1.2, and more details are given in Appendix 3A.

3.4 THE CONCEPTUAL STRUCTURE OF THE LASER CUTTING DATABASE

The conceptual structure provides a model of the database which was used as the basis for constructing the data tables of the laser cutting database, within the SQL/RT database system.

3.4.1 BACKGROUND

The first prototype of this model was constructed prior to the development of the data collection (collector) and prediction (predictor) modules, as these were in an early stage of design and additionally the experimental programme had not yet started, due to the cutting equipment not being commissioned. Consequently, the prototype was built in the scope of knowledge acquired from the literature studied which addressed laser cutting (chapter 1). Subsequently it was modified many times, due to implications upon data structure imposed by the two modules above and the experimental results. These modifications were necessary due to:

- Missing parameters from the model.

- Parameters wrongly placed within database tables.

- Conflicts in requirements between the collector and the database itself.

The last item required several modifications to the database tables, to comply with the implications of the collector and the interface between the database and the collector.

3.4.2 THE DESIRED CHARACTERISTICS OF THE CONCEPTUAL MODEL

The desired characteristics of the laser cutting database include:

1) Comprehension: The database structure needs to embody all data concerning the laser cutting process. This requirement was fulfilled in the scope of the knowledge collected from the studied literature and the experimental work.

2) Expandability: Each of the database tables must have the ability of expanding in the degree (number of columns), and in the cardinality (number of rows). Additionally, the database must be able to expand in respect to the number of tables, this allowing the accomodation of new materials or parameters which cannot be predicted at this time.

The implementation of this characteristic relates to two aspects. The first, the SQL/RT system, which readily allows for all the expansions detailed above, due to its relational nature. The second aspect concerns the interaction between the database and the collector. A strategy was adopted when designing the collector, which implies minimum alterations to the collection program, when it is required to expand the database. This strategy was based on program modularity. The proof of this, is that insertion of new rows to existing tables needs no alteration to the collector. While a few statements only are needed when it is required to introduce a new material table.

3) Data independence: The relational structure of SQL/RT takes the major load of this feature. No dependencies exist between the various tables, or between the rows of each table. This characteristic was supported by the design of the conceptual model, by proper partitioning of the domain and the range of the data, and by proper referencing between related data.

4) Data integrity: This is one of the very important features required for the data of the database, and can be summarised by the following principles:

- If a data entry (field) is critical to the operation of the database (i.e without its insertion the row of data is meaningless), then it must be supplied.
- The supplied data must be correct.
- Referencing between the related records in the various tables, must be reliable.

The first principle was implementd when the tables were defined. Columns of crucial importance were defined to the system as "mandatory", i.e insertion of a record will not be executed if data at this column is missing. The second principle cannot be implementd in code, and must be ensured by the user, since there are no means to check a wrongly typed number or name. The third principle was implemented by the introduction of unique identification keys between related data in different tables, as will be seen later in this chapter.

5) Reduced redundancy of data: This characteristic was necessarily strived for in the early stages of planning of this module. This is required in order that the lightest possible load is placed on the computer memory and processor, by not handling redundant data. This was insured by partitioning and referencing data correctly.

3.4.3 THE DESIGN OF THE CONCEPTUAL MODEL

The first step in designing this model, is the segregation of the prospective data into logically separate groups, in such a way to fulfill the above requirements, and to match the requirements of the collector.

Grouping of data was made according to the following principles:

- Each group represents an individual relation (table) in the database.

- The parameters (columns) of each group are closely related to each other in a systematic or thematic sense.

- A reasonable number of parameters in each group, to contribute to the features of data independence, integrity of the database and speed of search by the collector.

The database structure consists of the following groups:

1. Materials.
2. Laser systems.
3. Beam Delivery Systems (BDS).
4. Process parameters.
5. Real-time parameters (in-process measurements).
6. Cut quality parameters (post-process measurements).
7. Miscellaneous information.
8. Production information.

The last two groups are auxiliary categories, containing documentary information only, about the production process.

MATERIAL GROUPS: Materials vary in their vulnerability to laser cutting. This is dependent upon their physical and chemical properties, and particularly those associated with laser-material interaction process, e.g thermal properties (chapter 1). However, materials can be categorised into classes that have similar cutting characteristics(i.e similar thicknesses of these materials can be cut at the same feed rate with the same laser process parameters).

The groups of materials which currently exist in the database module, were categorised according to perceptions gained from the literature review and cutting experiments. These groups are:

Mild steel
Stainless steel
Carbon steel
Alloy steel
Titanium and titanium alloys
Aluminum and aluminium alloys
Copper and copper alloys
Plastics
Wood

Each of these classes is labelled by its "CLASS OF MATERIAL". A further subdivision called "TYPE" gives the specific type of each material in a class. Each of these material classes resides in one physical table in the database.

The material tables are considered as master tables for the selection of process parameters from the database. This is because each record of these tables contains all the pointers (reference codes) to the related records, in the other tables. Fig 3.4 illustrates the referencing method. It shows that each material table contains a number of columns of reference codes (C1, C2 and ID). Each of these

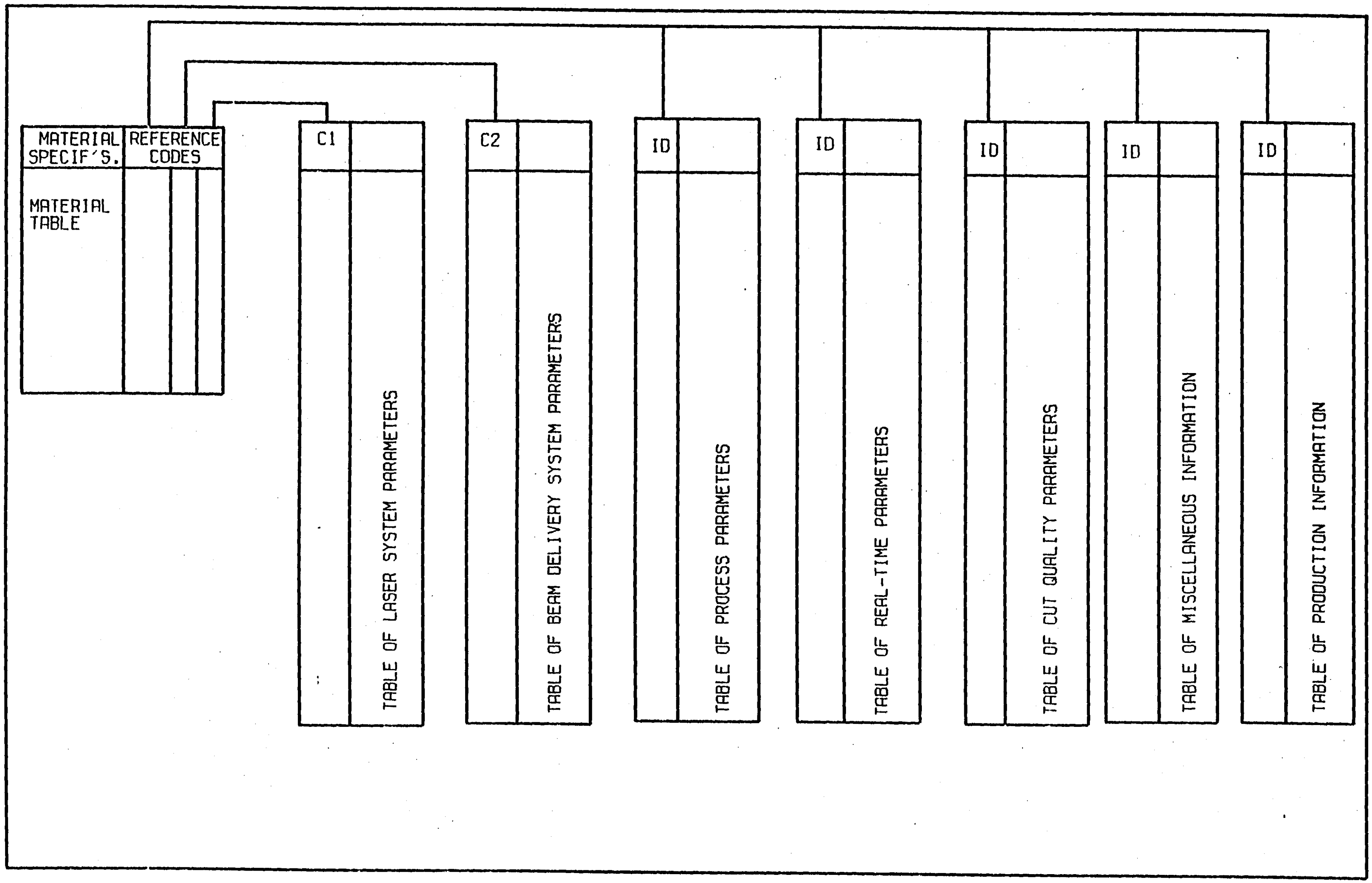


FIG 3.4 THE CONCEPTUAL MODEL OF THE LASER CUTTING DATABASE MODULE

codes is used as a primary key in one or more other tables. The reference codes assigned to a record in a material table, becomes unique to that record and to the related records in other tables.

As some of the records in tables have a one-to-one mapping with records held in the material tables, one reference code (ID) can be used to access a large number of tables (see fig 3.5). The total number of codes required was, consequently, reduced to three. This was a positive contribution to the requirements of reduced Redundancy and improved integrity.

The two columns C1 and C2, are excepted from the above reduction, and are referenced individually due to the following facts:

- A large number of cuts can be made with one laser system. This means that one data record in the table of laser systems, can be shared by a large number of records in the other tables.

- A large number of cuts can be made with one beam delivery system (BDS). This means that one data record in the table of beam delivery system, can be shared by a large number of records in the other tables.

- A laser system can be coupled to more than one beam delivery system.

- Identical beam delivery system can be used with more than one laser system.

The use of this method results in large reduction of data, since in real life the tables of laser and BDS will normally contain only a few records.

The following parameters are included in each material table (fig 3.5):

- Type of material: The specific type of material as defined by common standard or by the manufacturer. This item represents, implicitly, all the chemical and physical properties of the named material. These properties are not considered explicitly, since the model of the planning system is empirical and not analytical.

- Thickness of the material,

- Constituents: The significant elements only.

- Surface roughness of the material.

- Surface treatment (if any).

The last three items are provisional only, at the present time, but are included within the database structure for future use.

- ID, CL, CD codes: As explained above.

The structures of the tables are given in the figures 3.6 through 3.12, as follows:

- Fig 3.6 : Laser system table
- Fig 3.7 : Beam delivery system table.
- Fig 3.8 : Process parameters table.
- Fig 3.9 : Real-time parameters table.
- Fig 3.10 : Cut quality parameters table.
- Fig 3.11 : Miscellaneous information table.
- Fig 3.12 : Production information table.

3.5 CREATION OF THE LASER CUTTING DATABASE TABLES

This section is to explain the physical presentation of the tables of the groups explained above and their underlying data, within SQL/RT database system. It involves:

- Creation of the required number of tables.

- Defining the required columns in each table.

- Specifying the columns attributes.

TYPE	CONSTITUENTS	THICKNESS	SURFACE FINISH	SURFACE TREATMENT	ID	CL	CD
SPECIFIC TYPE OF MATERIAL	SIGNIFICANT CONSTITUENTS	THICKNESS OF MATERIAL	SURFACE ROUGHNESS OF THE MATERIAL	SURFACE TREATMENT OF THE MATERIAL	THE UNIQUE IDENTIFICATION CODE	LASER SYSTEM REFERENCE CODE	BEAM DELIVERY SYSTEM REFERENCE CODE

FIG 3.5 : MODEL OF THE MATERIAL TABLES

CL	LASER TYPE	STATE OF PULSATION	BEAM DIVERGENCE	MODE QUALITY	BEAM WAIST SIZE	MAXIMUM POWER	MIMIMUM POWER
LASER SYSTEM REFERENCE CODE	MAIN TYPE OF LASER (CO2, YAG)	PULSED OR CONTIUOUS WAVE (CW)	DIVERGENCE OF THE OUTPUT LASER BEAM	MODE ORDER OF THE OUTPUT BEAM	BEAM WAIST DIAMETER	MAXIMUM ACHIEVABLE LASER POWER	MINIMUM CONTROLLABLE POWER

FIG 3.6 : MODEL OF LASER SYSTEM PARAMETERS TABLE

CD	BEAM ROUTE DISTANCE	LENS TYPE	LENS MATERIAL	FOCAL LENGTH	FOCUS DIAMETER	BEAM INTENSITY	POLARI-ZATION	NOZZLE DIAMETER
BEAM DELIVERY SYSTEM REFERENCE CODE	BEAM ROUTE LENGTH BETWEEN THE OUTPUT WINDOW AND THE FOCUS LENS	GEOMETRICAL TYPE	KCL, Zn-Se, ...etc.	FOCAL LENGTH OF THE FOCUS LENS	BEAM DIAMETER AT THE FOCUS	BEAM INTENSITY AT THE FOCUS	LASER BEAM POLARIZATION TYPE (LINEAR, CIRCULAR)	DIAMETER OF NOZZLE ORIFICE

FIG 3.7 : MODEL OF BEAM DELIVERY SYSTEM TABLE

ID	POWER	FEED	GAS TYPE	GAS PRESSURE	STAND -OFF	FOCUS DEPTH	STATE OF PULSATION	PRF	PULSE WIDTH	PULSE ENERGY
UNIQUE IDENTIFICATION CODE	LASER POWER USED IN THE CUTTING PROCESS	CUTTING SPEED	TYPE OF ASSISTANT GAS (O_2, AIR, MIXTURE)	ASSISTANT GAS PRESSURE	GAP BETWEEN NOZZLE AND WORK-PIECE	LOCATION OF THE BEAM FOCUS WITH REFERENCE TO THE MATERIAL SURFACE	TYPE OF LASER (PULSED OR CW)	PULSE REPETITION FREQUENCY	PULSE DURATION	ENERGY SUPPLIED PER EACH PULSE
								IF LASER WAS PULSED		

FIG 3.8 : MODEL OF PROCESS PARAMETERS TABLE

ID	INCIDENT POWER	REFLECTED POWER	SURFACE TEMPERATURE
UNIQUE IDENTIFICATION CODE	LASER BEAM POWER FOCUSED ONTO THE MATERIAL SURFACE	LASER BEAM POWER REFLECTED FROM THE MATERIAL SURFACE	MATERIAL SURFACE TEMPERATURE

FIG 3.9 : MODEL OF THE REAL-TIME CONTROL PARAMETERS TABLE

ID	KERF WIDTH	CUT ROUGHNESS	TAPER	DROSS SIZE	HEAT AFEECTED ZONE	QUALITY GRADE	SPEED GRADE
UNIQUE IDENTIFICATION CODE	WIDTH OF CUT AT TOP SURFACE	ROUGHNESS OF THE ROUGHEST ZONE OF THE CUT SIDES	DIFFERENCE BETWEEN THE WIDEST AND THE NARROWEST WIDTHS OF THE CUT, ON ONE SIDE	AVERAGE HEIGHT OF THE DROSS FORMED ON THE BOTTOM SURFACE OF THE MATERIAL	WIDTH OF THE ZONE AFFECTED BY THE PROCESS HEAT, ON ONE SIDE OF THE CUT	GENERAL ASSESSMENT OF CUT QUALITY EXPRESSED AS A PERCENTAGE	PERCENTAGE FROM MAXIMUM FEED THAT CAN BE ACHIEVED FOR THE CUT MATERIAL AND EMPLOYED LASER POWER

FIG 3.10 : MODEL OF CUT QUALITY PARAMETERS TABLE

ID	INFORMATION SOURCE	REFERENCE CODE	CRITICAL RADIUS	CRITICAL LENGTH	LOOPING
UNIQUE IDENTIFICATION CODE	SOURCE OF THIS DATA (LITERATURE, OWN)	USER REFERENCE CODE OF THE DATA	CRITICAL RADIUS OF CONTOURE CUTTING FOR THIS PROCESS	MINIMUM CUT LENGTH FOR THIS PROCESS	WAS LOOPING AROUND CORNERS REQUIRED ?

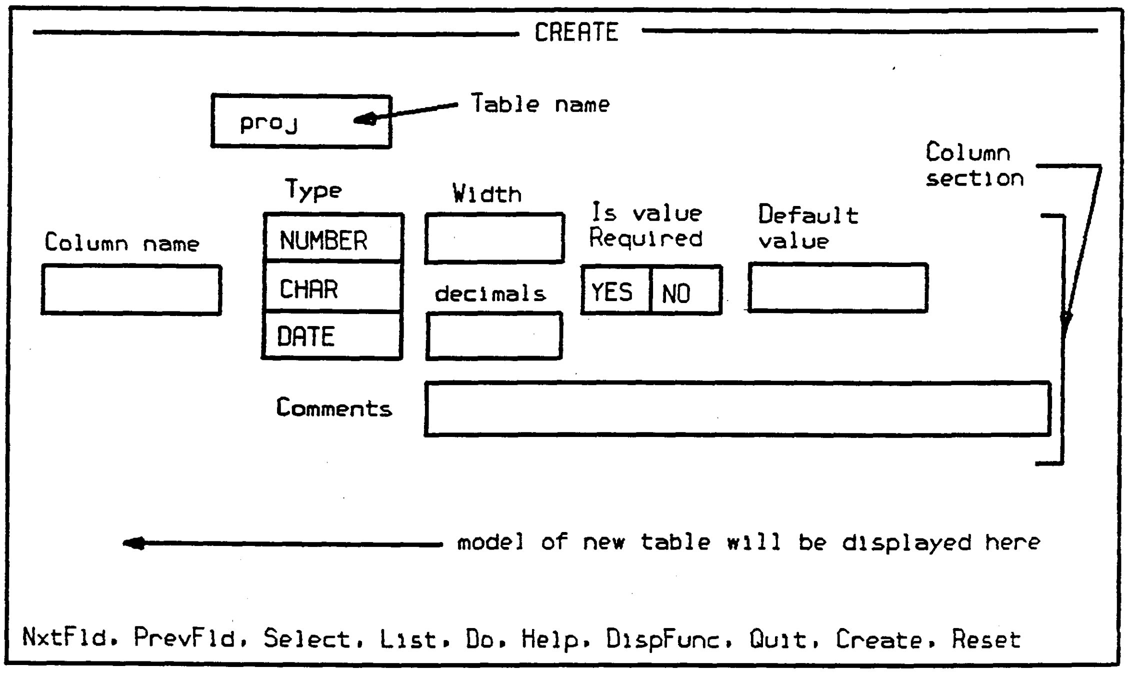
FIG 3.11 : MODEL OF MISCELLANEOUS INFORMATION TABLE

ID	PART NUMBER	BATCH NUMBER	PROCESS NUMBER	CUT NUMBER	PROCESS DATE	OPERATOR NAME
UNIQUE IDENTIFICATION CODE	NUMBER OF PART NC PART PROGRAM					

FIG 3.12 : MODEL OF PRODUCTION INFORMATION TABLE

Easy SQL/RT interface was used for these processes. it was used due to its simplicity and the ability to view a table model while still being created. Also, it facilitates the passing of help comments to the end-user, who would use this interface to insert new data into tables. Fig 3.13 shows the Easy SQL/RT panel for creating a table called "proj". A representative sample of defining the columns of this table is shown in fig 3.14. Fig 3.14(a) shows the operation of defining the first column (PROJNO). This column is of data type "integer number", three digits long, and is mandatory to supply by the user when inserting new records. Fig 3.14(b) is when switching to the next panel for defining the second column. The name, data type and width of the first column is shown at the bottom left corner of this panel. The second column (PNAME) is of data type character, 10 characters maximum width, and is optional.

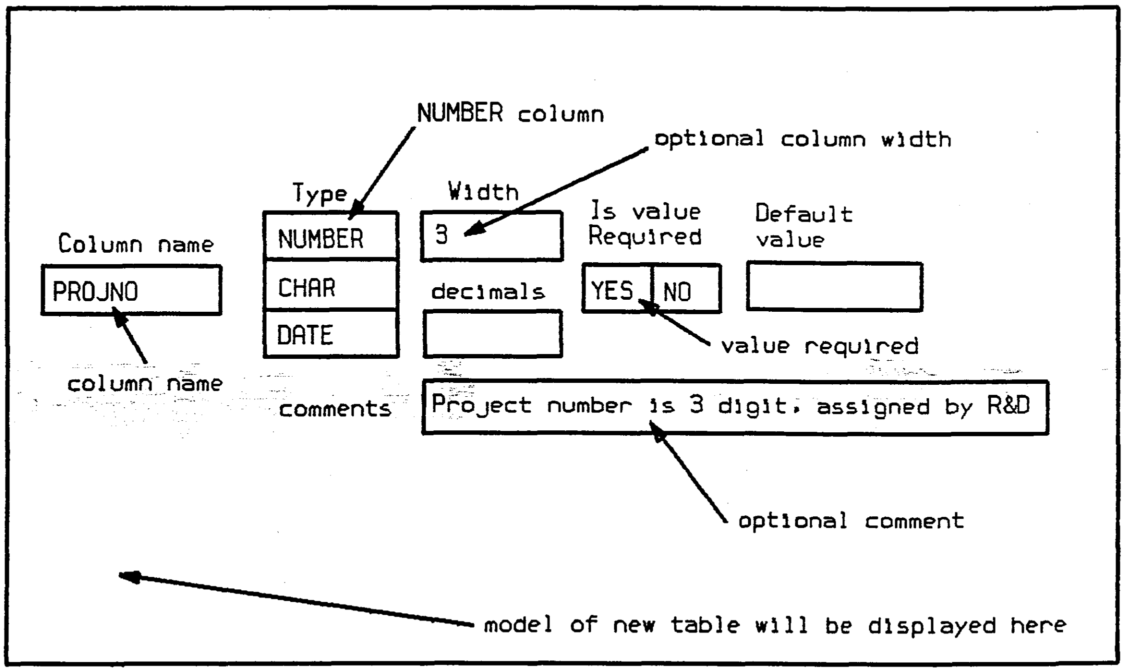
The Descriptions of the created tables and the underlying columns, together with a representative sample of each table, are contained in Appendix 3B. While the method used for the insertion of data into the database is explained in Appendix 3C.



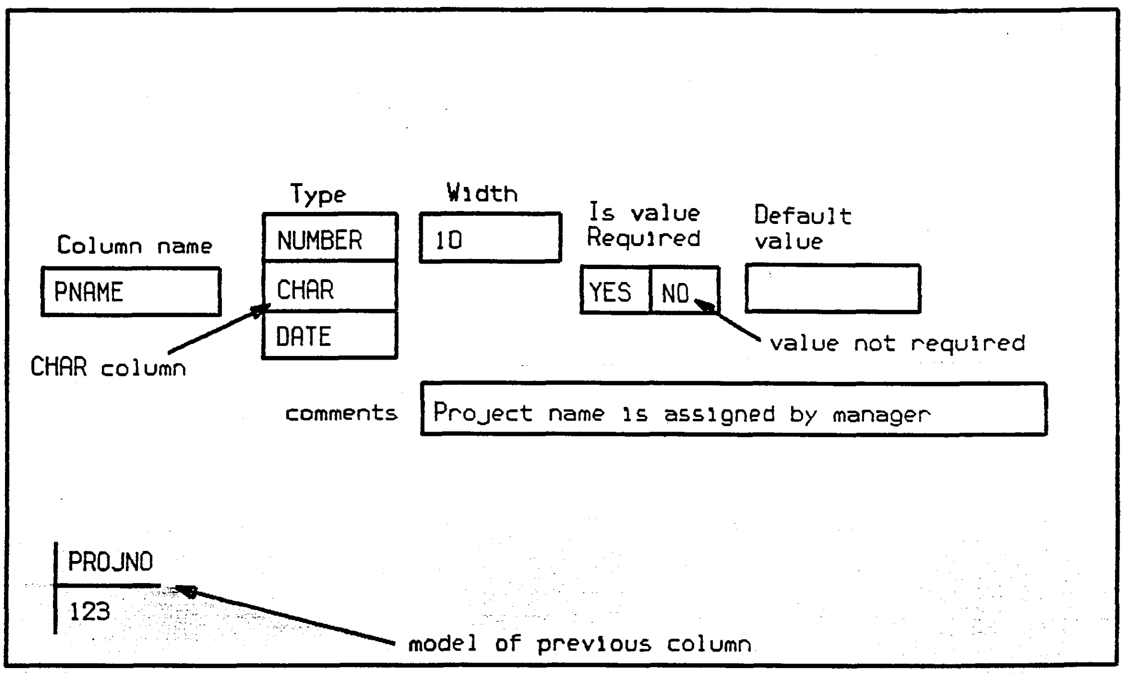
- Tab (Next Field) go to next field
- Enter move to next line
- Shift-Tab (Previous Field) return to previous field
- F4 (List) display reference box
- F5 (Select) choose option
- F9 (Do) execute and commit changes

FIG 3.13 : SCREEN MENU FOR CREATING A TABLE IN EASY SQL

SOURCE : Easy SQL/RT USER'S GUIDE (IBM UK LTD)



(a) : CREATION OF THE FIRST COLUMN



(b) : CREATION OF THE SECOND COLUMN

FIG 3.14 : CREATION OF THE COLUMNS OF TABLE (proj)

SOURCE : Easy SQL/RT USER'S QUIDE (IBM UK LTD)

REFERENCES

REF. NO.	DETAILS
3.1	Date, C. J., An introduction to database systems, Third Edition, Addison-Wesley Publishing Co., 1981.
3.2	Schmidt, J. W., and Brodie, M. L., Relational database systems, Analysis and comparision, Springer-Verlag, 1983.
3.3	Deen, S. M., Principles and practice of database systems. Macmillan Publishers Ltd., London, 1985.
3.4	Alagic, S., Relational database technology. Springer-Verlag, 1986.

CHAPTER 4

LASER CUTTING EXPERIMENTS AND ANALYSES

The principal objectives of these experiments were:

1) The provision of a reasonable amount of data for the laser cutting database management system in order that database tables were reasonably filled, within attainable performance of laser cutting equipment.

2) The provision of an operational set of data to facilitate development and testing of data collection methods and associated data collection and prediction software.

3) To provide practical experience as to the relative importance of different cutting parameters and their interactions. This contributed positively to the design and development of the software modules above.

Cutting experiments were accomplished using two types of laser machining systems. One based upon a Continuous Wave Gas (CO_2) laser, the other using a Pulsed, Solid State, Nd:YAG laser. These represent the two principal types of laser used in Laser Based Manufacturing.

The chapter describes the laser cutting experiments performed together with their results and analyses.

4.1 DESCRIPTION OF THE Nd:YAG LASER BASED CUTTING MACHINERY

This system is an integrated manufacturing unit that comprises of two major components namely, an Nd:YAG laser, and a Computer Numerical Controlled (CNC) X-Y Work Handling System. The configuration is shown in fig 4.1.

4.1.1 THE Nd:YAG LASER

This is a pulsed, solid state Nd:YAG laser (type MS 300) made by LUMONICS LTD./JK PRODUCT DIVISION, which emits at 1.06 microns. It can be operated in two modes, i.e welding or drilling (cutting). In drilling mode the maximum average power is 100 Watts, while the current pulse width (duration) can be varied between 0.1 and 10 mSec. The main components of the laser are: a resonator, two synchronised High Voltage (HV) power supplies, a laser beam delivery system, and cooling system.

The laser resonator for drilling consists of a laser rod, pumping chamber, two mirrors and a telescope. The laser rod is made of Neodymium doped Ytterium-Aluminium-Garnet. The pumping chamber comprises of two linear flashtubes in a

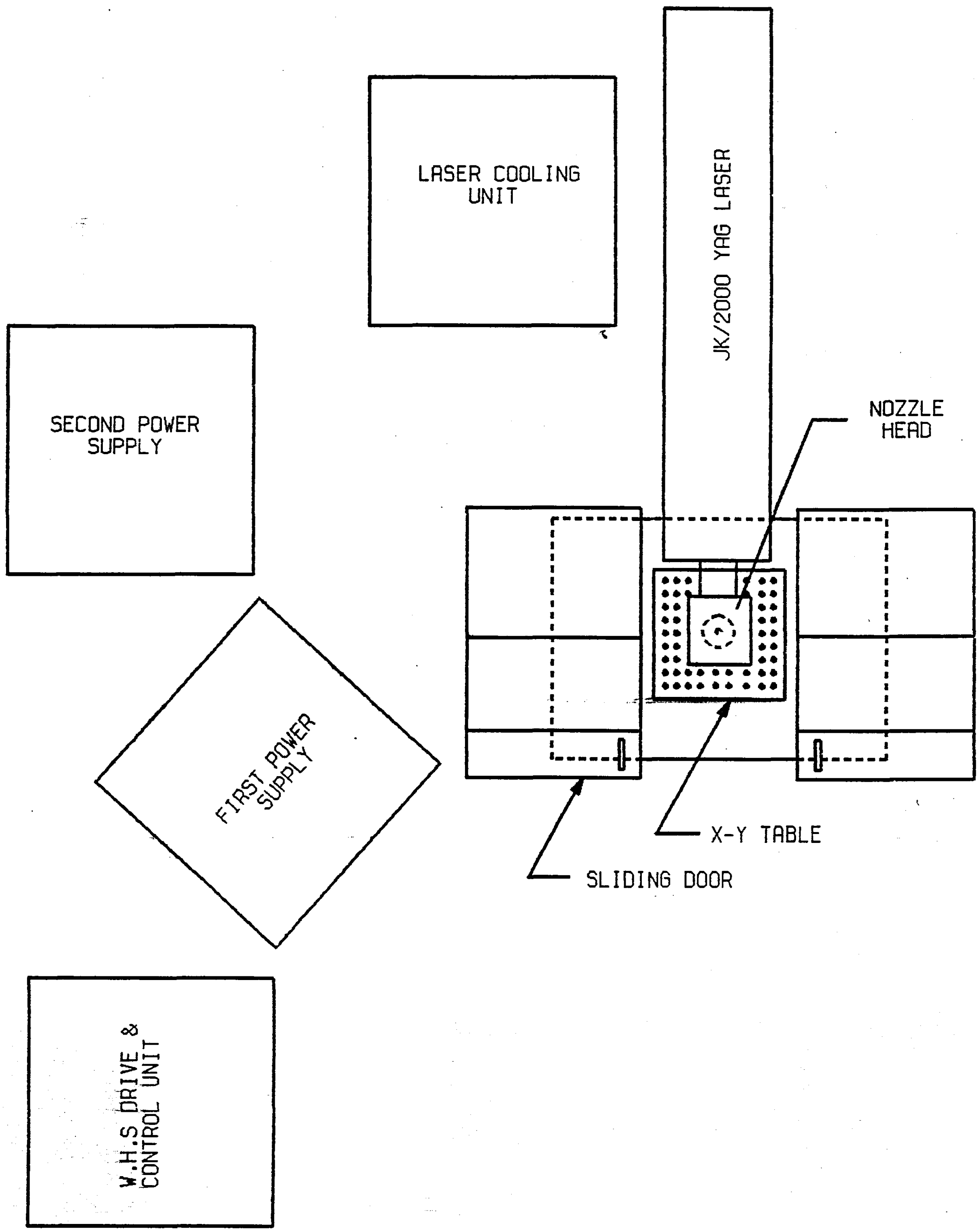


FIG 4.1 : CONFIGURATION OF THE YAG LASER CUTTING MACHINERY

ceramic diffuse reflector. The output mirror is plane of low reflectivity, while the rear mirror is concave and totally reflective. A telescope and a water cooled aperture fitted between the laser rod and the rear mirror, are used to discriminate against higher order modes.

Each power supply feeds a flashtube with a selectable electrical current at constant voltage. Current selection is either performed manually at the laser control panel (local), or programmed from the CNC work handling system controller (remote).

The laser control panel is mounted in one of the HV supply cabinets. This panel contains all the necessary switches and dials to control laser startup, output performance, and normal and emergency shutdown.

The beam delivery system consists of: a beam expander, an optically transparent deflection mirror, viewing binoculars, and focusing lens (fig 4.2). The beam expanding telescope expands the laser beam onto the deflecting mirror which deflects it into a vertical axis towards the work. The effect of expanding the beam before entering the focusing lens is to reduce the diameter of the focal spot, according to the following relation (ref 4.1):

$$d = 4 L F / \pi D$$

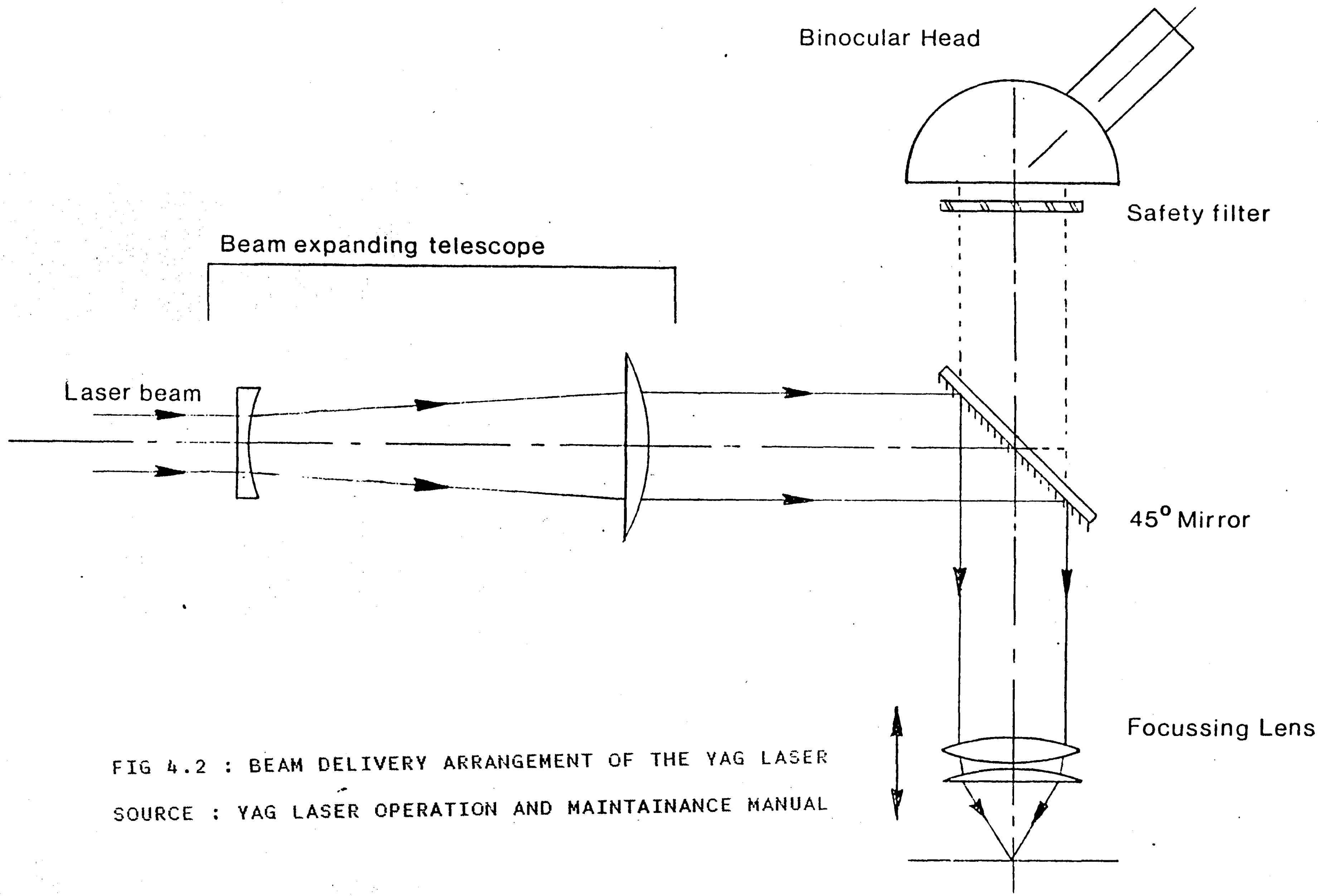


FIG 4.2 : BEAM DELIVERY ARRANGEMENT OF THE YAG LASER

SOURCE : YAG LASER OPERATION AND MAINTAINANCE MANUAL

Where d is the spot diameter, L is the wave length, F is the lens focal length and D is the beam diameter.

Consequently, the laser beam power intensity is increased by a considerable factor (i.e proportional to the square of the focal spot diameter), the power intensity is calculated as (ref 4.2):

$$I = 8\,P / \pi d^2 \qquad \text{where P is the beam power.}$$

The calculated intensity was of the order (10^9 W/cm^2).
The focusing lens (80mm focal length) is contained in the nozzle head, and is used to focus the laser beam onto the work. Viewing binoculars are used to optically align the work piece under the nozzle.

The nozzle head is height adjustable with a total travel of 10mm allowing for variation in material thickness, and control of the focused beam and nozzle stand-off. The lens and nozzle are cooled by the gas that is used to assist the cutting process. This gas is delivered to the nozzle head and ejected through the 1mm orificed nozzle coaxially with the focused laser beam.

The laser cooling system incorporates a closed primary circuit using a weak de-ionized sodium nitrite solution to carry heat from the pumping chamber to the liquid/water heat exchanger which uses mains water (secondary circuit).

The laser power is monitored by a calibrated energy monitor. A pulse counter is used to monitor the total number of pulses made by the flashtubes.

Laser output is controlled by three pulse parameters:

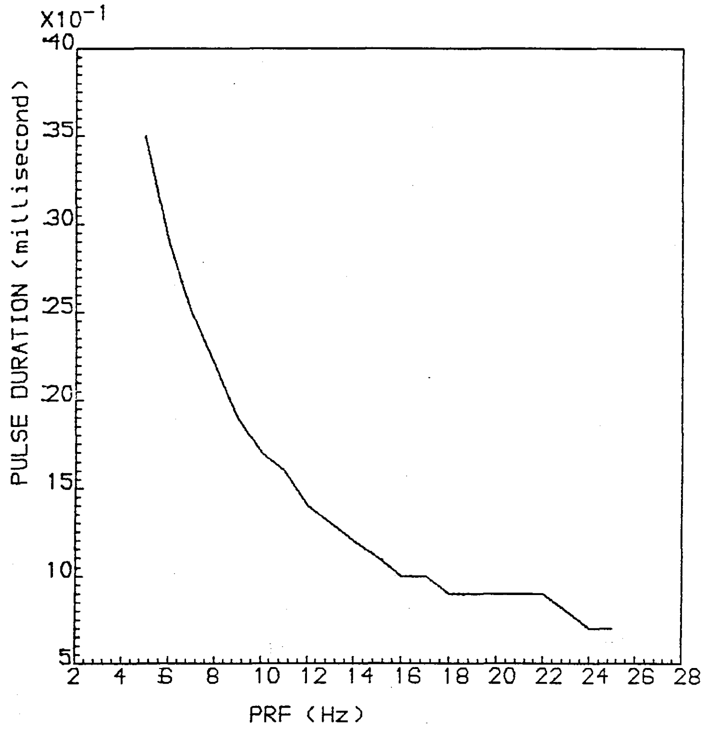
- Pulse Repetition Frequency (PRF Hz).

- Pulse Width (milliSecond).

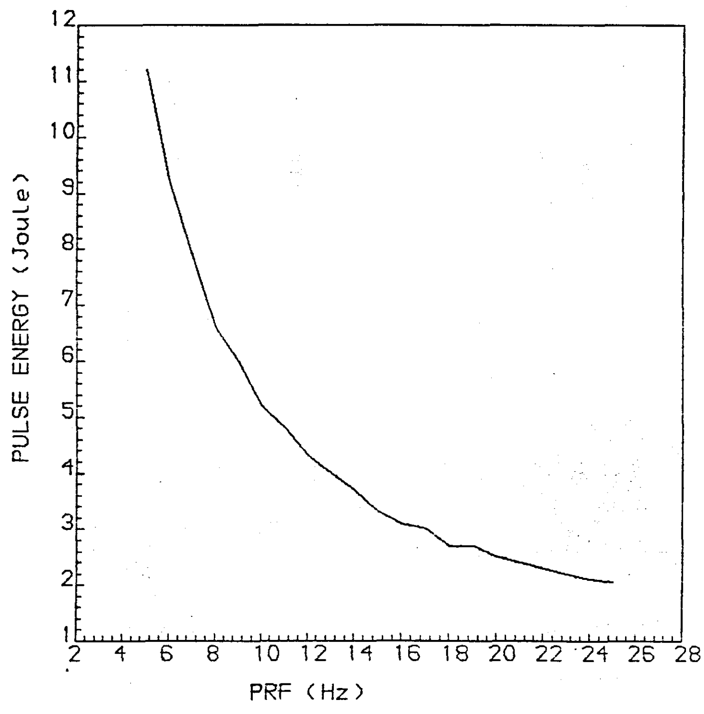
- Flashtube current (percentage of maximum current).

Different combinations of these parameters result in different mean and peak pulse power levels and output durations. A series of tests were made prior to commencing the cutting experiments, to determine the operating envelope of the laser system. This was done as one would expect that over a period of time the operating envelope would change, due to laser rod damage and flashtube degradation. Fig 4.3 presents Pulse Width versus PRF for maximum Pulse Energy (the supply current was held at 99% for the whole range of PRF).

It should be noted that the selected pulse width represents the electrical pulse width input to the flashtubes. The actual laser output pulse duration was measured by means of an oscilloscope. This indicated that both input and output



PULSE DURATION Vs PRF



PULSE ENERGY Vs PRF

FIG 4.3 : PULSE PARAMETERS OF THE Nd:YAG LASER durations where equal up to 4 milliseconds. Above this, output durations become shorter than the current pulse. The experimental programme remained within this 4 millisecond envelope.

4.1.2 THE WORK HANDLING SYSTEM (WHS)

The work handling system consists of two subsystems, a stepper motor driven X-Y table assembly and a CNC controlled drive system.

The X-Y table (300 X 300 mm), is actuated by two stepper motors with 200 mm travel in X and Y directions. These receive actuating signals from the CNC controlled drive system, according to a predefined program. A matrix of 9 X 9 round head brass studs are used to support the workpiece.

For operator safety, the X-Y table is enclosed in a metal compartment with two sliding doors which are safety interlocked with the beam shutter.

The CNC controlled drive system uses a MICON 850 CNC controller and two electronic drives to power the X-Y table motors. The controller is programmed using the Numericon 850 CNC Programming Language (near standard G & M codes) (ref 4.3).

Part programs can be input to the CNC memory in three ways:

- Manual Data Input using the keyboard.

- Reading stored part programs from cassette.

- Reading part programs via an RS 232 communication port from a computer, punched tape unit, or other suitable source.

In addition to controlling the X-Y table, the CNC remotely controls the laser by setting the three relevant pulse parameters (i.e PRF, Pulse Width & Current) from within the part program, and opens and closes the laser shutter.

4.2: DESIGN OF THE YAG LASER CUTTING EXPERIMENTS

As previously detailed, the process of laser cutting is affected by several interacting parameters, such as beam power, cutting speed, gas type and pressure. Thus, the spectrum of the cutting experiments must be wide enough to facilitate studying all the factors (parameters) that are believed to have a significant effect on the cutting process, and to satisfy the other two requirements stated at the beginning of the chapter.

Factorial design of experiments is the methodology most commonly used in such multi-factor problems (ref 4.4), where the experiments covers the possible combinations of the different levels of all the factors involved in the process. There are several types of factorial design of experiments, including: Latin square, randomized (ref 4.4), paired comparison (ref 4.5), and t-way crossed classification (ref 4.6). These can also be classified as, complete, fractional (incomplete), symmetrical and asymmetrical (mixed) factorials (ref 4.6).

A factorial design of the laser cutting experiments, not only had to yield experimental results in line with the three objectives stated at the begining of this chapter, but also need to take account of the cutting equipment and time restrictions. This was achieved as follows:

A) Specifying the factors of interest: Parameters that were selected as influencing the cutting process were:

1. Type of material.
2. Thickness of material.
3. Feed rate (cutting speed) .
4. Pulse repetition frequency (PRF) .
5. Pulse energy.
6. Assistant gas type and pressure.
7. Nozzle stand-off.

Other factors such as type and focal length of the focusing lens, focus diameter, power intensity at the focus, nozzle design and laser mode quality, do affect the process, but they were not measured or directly altered due to time limitations. However, their effects on the laser cutting process are discussed in chapter 1.

B) Grouping of the cutting experiments: It was believed that cutting experiments based on random combinations of the above parameters would not serve the purpose of these experiments, as this would require a large number of experiments in order to assure adequate coverage of all the possible combinations. Therefore, systematic grouping (blocking) of the tests into homogeneous blocks was used to yield the results required (ref's 4.4 and 4.6).

The major systematic blocks were chosen to be homogeneous in material type, due to its prominent effect on laser cutting. Further sub-grouping was made in terms of pulse parameters, stand-off and gas pressure, in such a way to reasonably cover all possible combinations (detailed in section 4.2.2).

C) Specifying the number of levels: Each parameter was assigned with a different number of levels, according to its perceived importance and practical range of operation.

D) Selecting a factorial design: A factorial design type "fractional asymmetrical t-way crossed classification" (as defined by ref 4.6) was believed to be the most suitable type for this situation. If (k) is the number of levels corresponding to each factor, and (t) is the total number of factors, the total number of experiments (K_{tot}) is the product of multiplication of these classifications,

$$K_{tot} = k_1 * k_2 * k_3 * \ldots\ldots * k_t$$

The adopted factorial is considered asymmetrical because the number of levels varies across the factor. This design is also fractional (incomplete), because it does not cover all possible factorial combinations, for the following reasons:

- Practically, it is impossible to test all existing types of materials at all thicknesses. Thus a representative set was selected (the area of interest, in factorial design terms)

- Some factors are dependent within this area of interest, for example it is not possible to manipulate the levels of PRF and pulse energy independently to satisfy the condition of complete factorial design, due to limited pulse energy (see fig 4.3). This led to implicit testing of both parameters, i.e whenever altering PRF, pulse energy is automatically altered.

4.2.1 MATERIALS FOR EXPERIMENTATION

A representative sample of commercially available materials, which are routinely laser cut, were selected for the experimental work. These included: mild steel, stainless steel, tool steel, titanium, aluminium, brass, nickel and inconel.

The prediction software must be capable of predicting cutting parameters for thicknesses not present in the database system. This implies that some form of interpolation is required between at least two existing thicknesses. To ensure that enough experimental data was taken to verify this, two materials (mild steel and aluminium) were cut at two thicknesses. Table 4.1 details the experimental materials used.

4.2.2 THE CUTTING EXPERIMENTS

The cutting experiments were performed according to the principles of factorial experiments explained previously.

The following settings of process parameters were used for each type of material:

1) Four feed rates, the highest of which is the maximum achievable feed for that material, that results in a continuous series of holes.

MATERIAL TYPE	CONTENTS (%)	THICKNESS (mm)
Mild steel (BS970EN3B)	C-0.25, Si-0.35, Mn-1.0, S-0.06, P-0.06 & Fe	3.2 , 1.6
Stainless steel (AISI321)	Cr-18.0, Ni-9.3, Mn-1.0, Si-0.8, C-0.1 & Fe	1.6
Tool steel	C-1.0, Mn-1.3, Cr-0.65, W-0.8, V-0.15 & Fe	1.6
Titanium	Commercially pure (99.6)	1.6
Aluminium	Commercially pure (99.0)	1.6 , 0.7
Nickel	Commercially pure (99.98)	0.9
Inconel	Ni-72.0, Cr-15.5, Fe-8.0, Co-3.0	0.75
Brass	Cu-63.0 , Zn-37.0	1.6

TABLE 4.1 : MATERIALS FOR LASER CUTTING

2) Two settings of PRF, and consequently two levels of pulse energy.

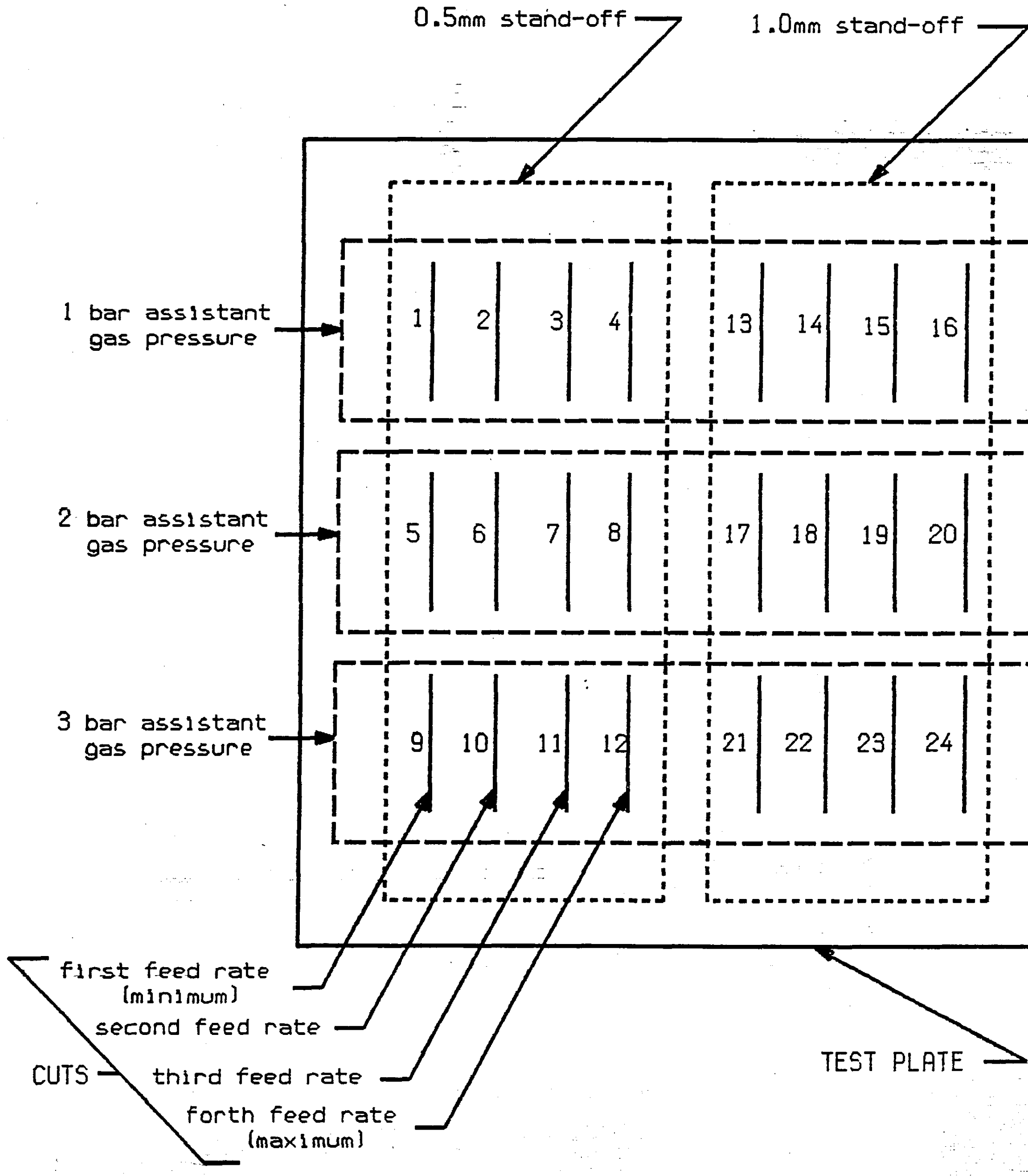
3) Three settings of assistant gas pressure: 1 bar, 2 bars and 3 bars. Oxygen was used as the assistant gas for all materials, except titanium where inert gas (argon) was used.

4) Two settings of nozzle stand-off i.e 0.5mm and 1.0mm.

This requires 48 cutting tests per material per thickness, giving a total of 480 cutting experiments. The 48 cuts were dispersed over two test plates of 120 X 120 mm, each having 24 cuts with the same PRF and pulse energy. Fig 4.4 shows the layout of the test plate.

The 24 cuts are divided into two parallel groups, the left group for 0.5mm stand-off and the right one for 1.0mm. Three rows per group, the upper row is for 1 bar assistant gas pressure, the middle for 2 bars and the lower for 3 bars. Each row consists of 4 cuts 20mm long, each of which was cut at a different feed rate, with feed increases from the cut on the left to that on the right. Cutting propagation on all cuts is from top to bottom (i.e in the negative Y direction) .

The cutting experiments were in the form of slits rather than complete cuts. This was to enable post process



N.B : All these cuts are for one setting of PRF & pulse energy

FIG 4.4 : TEST PLATE LAYOUT FOR Nd:YAG LASER CUTTING EXPERIMENTS measurements of kerf width and dross size to be taken, and to allow identification of the slits.

Work sheets were created to document experimental information (fig 4.5): material particulars, process parameters, post process measurements and other relevant data. One work sheet was required for each test plate and so carries the plate identification number in addition to the data from the 24 slits.

4.3 POST PROCESS MEASUREMENTS AND ANALYSIS OF CUT QUALITY

Two aspects will be covered in the following sections. Firstly an outline is given as to the primary visual checks and verifications made on the test plates immediately after cutting. Secondly the detailed post process measurement techniques and results are presented.

4.3.1 PRIMARY VERIFICATION

These verifications include: CUT/NO-CUT checks , surface effects on the material surface, and any in-process observations that might be helpful for symptom explanation.

MILD STEEL: No difficulty was experienced in cutting 3.16mm thick mild steel with a PRF of 10Hz, except at 1bar oxygen where heavy slag deposited in the cuts. Yet with

PLATE CODE :	MATERIAL :	MAT.TYPE	SURF.FINISH :	OPERATOR :
LASER TYPE : JK/YAG	THICKNESS :	SURF.TREATMENT :	TEST THEME :	DATE :

CUT No.	POWER (W)	FEED (mm/min)	GAS TYPE	GAS PRS. (mbar)	STD.OFF (micron)	P.R.F (Hz)	PULSE WIDTH (ms)	ENERGY (Joule)	KERF (mic.)	H.A.Z (mic.)	DROSS HEIT. (mic.)	TAPER (mic.)	ROUGH. CLA X	TYPE	STRIA PITCH (mic)	REMARKS

FIG 4.5 : SAMPLE OF EXPERIMENTAL WORK SHEET

higher pulse energy (PRF of 8Hz), cutting difficulties arose especially at 1.0mm stand-off where only two successful cuts at 2bar and 3bar oxygen could be achieved. The unsuccessful cuts were penetrated to about 70% of the material thickness and full of oxide slag.

The only setback with 1.6mm thickness was at the highest feed (225 mm/min), 10Hz PRF and 0.5mm stand-off where a disconnected series of holes was observed. This was due to relatively small hole diameter and high feed rate.

STAINLESS STEEL: A thin layer of oxide scale formed on the top surface of all cuts, which was 0.2 to 0.5mm wide on either side of the slit. Partial nozzle blockage by ejected metal was noted when cutting with 1bar oxygen, at 1.0mm stand-off.

TOOL STEEL: Observations on this material are similar to those for stainless steel. The scale margin widening (up to 0.7mm) with increasing feed.

TITANIUM: Nothing important is recorded except one occurrence of nozzle blockage at 0.5mm stand-off.

ALUMINIUM: When cutting 1.6mm thick aluminium, partial beam penetration occured at 15Hz PRF, 1.0mm stand-off and 1bar oxygen. A thin layer of easy removable metal was seen on the top surface on both sides of the kerf, which starts from 0.3mm in width, increases with feed up to 0.7mm. This phenomenon was also observed with 0.7mm thick aluminium.

NICKEL: The major difficulty with nickel is metal resolidification on the top side of the material, which was so serious that at 1bar oxygen continuous cutting became impossible. This effect is diminished with higher gas pressure and pulse energy.

INCONEL: The relatively small hole diameter generated by each pulse, seems to be the major limiting factor to feed rate. The nozzle orifice was frequently partially blocked when cutting at 0.5mm stand-off which made cutting at 10Hz PRF difficult.

BRASS: The top side of all cuts were affected by metal resolidification which became so serious at 1.0mm stand-off and 1bar oxygen such that full beam penetration was prevented. Frequent partial nozzle blockage was recorded.

4.3.2 ANALYSES OF CUT QUALITY

These analyses include precise measurements of cut quality features and presentation of the results. The sequence of presentation will be in the order of the five quality features measured, namely: kerf width, dross size, cut roughness, cut taper and heat affected zone.

The results are presented graphically on multicurve graphs where the quality items are on the ordinate. The abscissa represents the variation in feed rate.

Exception from this form of presentation will be exercised, however, as some materials show no significant variation in some quality features and/or no correlation with particular process parameters, instead the average range of data will be given.

Every graph basically contains 24 data points lying on six curves, representing the variation of one quality feature against one pulse parameter, two stand-off distances and three gas pressures. Thus to present a quality feature for one material, two of these graphs are plotted on one sheet, and for convenience are addressed as single graphs.

On any one graph, shorter or missing curves, indicate missing data points due to unsuccessful cutting. Six pairs (or less) of numbers appear below each graph. The number on the left is the assistant gas pressure in millibars and the one on the right is the stand-off distance in microns.

4.3.2.1 KERF WIDTH

The kerf width is the nominal width of the cut, taken as the distance from one edge of the cut to the other on the top surface of the material. when the cut edge is not

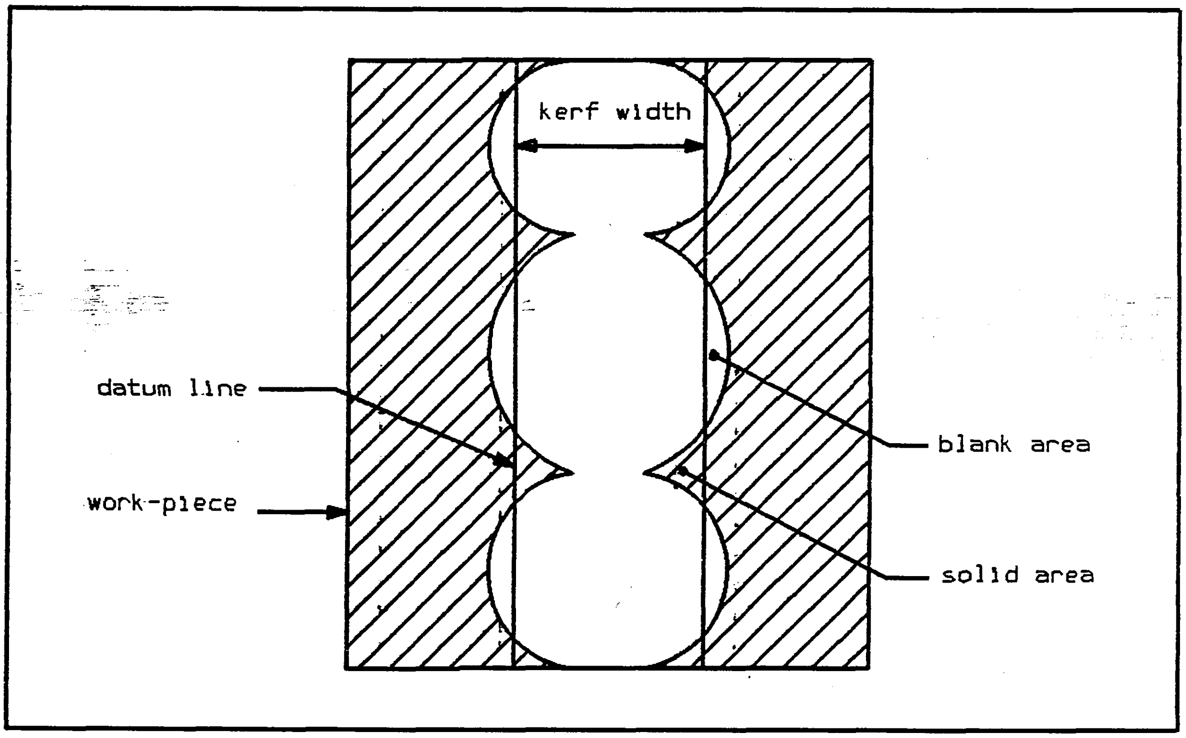


FIG 4.6 : KERF WIDTH MEASUREMENT PRINCIPLE

straight i.e serrated, the measuring datum on each edge is taken as the line that approximately equalise between the solid and the blank areas (see fig 4.6).

The measurements were made using a profile projector. Optics of 50X magnification were used to project the cut zone of the material surface on a screen provided with a crosshair.

The test plate is placed on the projector X-Y table, which is controlled by two manual digital micrometers. The table is moved in the X direction across the vertical hairline and the kerf is read from the digital readout.

Polishing the top surface was needed on all the materials to enable the cut edge to be seen.

In the following analysis, and for simplicity, PRF will often be used as an indicator of pulse energy as higher PRF means lower energy (section 4.1.1).

MILD STEEL: Observed kerf widths made in 3.16mm mild steel at 10Hz PRF ranged from 280 to 350microns (cuts at 1.0mm stand-off were in the upper band). At 8Hz PRF the range is 490 to 550microns with no significant influence from any process parameter (graph 4.1).

kerf widths in 1.6mm ranged from 420 to 540microns, lower settings of pulse energy, stand-off and gas pressure all lead to narrower kerfs (graph 4.2)

STAINLESS STEEL: Pulse energy has a clear effect on kerf width in stainless steel, where kerf width increasing considerably with higher energies. The kerf width at 10Hz PRF is in the range 380 to 480microns, while at 15Hz PRF it is 250 to 350microns (graph 4.3). At 10Hz PRF, wider kerfs were observed at 1.0mm stand-off distance. Feed effects can be observed at 15 PRF, kerf width increasing with higher feeds.

TOOL STEEL: The kerf seems to slightly widen with increasing oxygen pressure and stand-off (graph 4.4). The range of kerfs observed being 270 to 340microns.

TITANIUM: Increasing the stand-off distance tends to increase the kerf width, especially at low assistant gas (argon) pressure. Increasing pulse energy positively raised the range of kerf width from 220 to 320microns at 15Hz PRF, to 300 to 460microns at 10Hz PRF (graph 4.5).

ALUMINIUM: Aluminium is characterised by a very narrow kerf width when compared to the other materials tested. For 1.6mm thick material, the kerf width increases with pulse energy (decreases with PRF), from the range of 120 to 160microns at 15Hz PRF to 160 to 230microns at 10Hz PRF (graph 4.6).

For 0.7mm thick aluminium, stand-off distance is the dominant factor in increasing the kerf width range from 160 to 190microns at 0.5mm to 180 to 230microns at 1.0mm stand-off, at both PRF´s (graph 4.7).

NICKEL: The range of kerf widths is 320 to 400microns. No significant effective factor could be identified (graph 4.8).

INCONEL: At 15 PRF kerf ranged between 210 to 270microns for all experiments, except at 1.0mm stand-off and 3bars oxygen where it rose to 400microns. At 10Hz PRF the range is 320 to 420microns (graph 4.9).

BRASS: Kerf width is significantly affected by pulse energy, rising from a range of 130 to 200microns at 15 PRF to a range of 200 to 270microns at 10Hz PRF (graph 4.10).

4.3.2.2 DROSS FORMATION

It is believed that a quantitative method of assessing the amount of dross formed on the rear surface of the material, is more practical than a purely qualitative one (e.g heavy, medium or light dross). The average height of the dross formation is taken as being representative of "dross size". This is acceptable since the dross form and aspect ratio (width to height) is mainly dependent on the physical characteristics of the material (i.e material type), and is almost constant for each type of material. Additionally, it is practicable to measure this dross height with conventional instruments.

Dross size was measured using a vertically mounted dial gauge, scanning several arbitrary points along the cut, while the test plate was laid, face down, on a flat surface. The average of the measured heights was recorded as the dross size of that cut. It should be noted that easily removed dross (which is mainly oxides), was not measured as this would fall off during workpiece handling, and in practice would need no deburring process.

MILD STEEL: As the majority of the dross was easily removed oxides, very few deposits were left to be measured. In very few cases the dross size exceeded 50microns, while the average was less than 30microns

STAINLESS STEEL: No specific parameter can be identified as affecting dross size which mainly ranged between 30 and 60microns, while in a few cases it climbed to 100microns (graph 4.11).

TOOL STEEL: It is much like stainless steel including the range of sizes (graph 4.12).

TITANIUM: Feed rate and stand-off distance have distinct effect on dross formation. Higher feeds tend to reduce dross size, while higher stand-off generally causes an increase. The range of dross size observed was 250 to 750microns (graph 4.13).

ALUMINIUM: Similar phenomena to those seen on titanium were observed for both thicknesses of aluminium. The dross range for 1.6mm thick material is 200 to 450microns (graph 4.14). For 0.7mm thick aluminium, smaller dross size resulted from higher pulse energy and higher oxygen pressures, ranging 20 to 150microns (graph 4.15).

NICKEL: Higher feed rates and higher pulse energies resulted in smaller dross. The height ranged 200 to 650microns (graph 4.16).

INCONEL: Dross size decreased with increasing feed and gas pressure. The range of heights recorded was 40 to 180microns (graph 4.17).

BRASS: Dross size decreased with increasing feed and gas pressure. Higher stand-off distances resulted in increased dross size. The range of dross size was 120 to 400microns (graph 4.18).

4.3.2.3 CUT ROUGHNESS

The cut edges roughness was measured at the roughest zone of the edge using a roughness testing instrument (TALYSURF MODEL 105). This measures the roughness using the Centre Line Average (CLA) technique prescribed in BS1134, and gives the output in units of microinches or microns "Ra". The talysurf was calibrated for the desired range using a standard set of test pieces.

The roughness tests were always performed at the roughest zone of each specimen, 3 to 4 measurements were made at different locations of this zone along the test specimen, and the average of these readings was recorded.

The specimens for this test were prepared by cutting the test plate into individual pieces, after the cuts had been identity marked. The cut edges of ferrous metals were lightly brushed to remove oxide films generated by laser cutting process. All materials (except brass) showed serrated cuts. The serration pitch always conformed to the linear relationship between PRF and feed rate,

$$PITCH = FEED / PRF$$

This is based on the principle that a cut which is produced by the pulsed YAG laser is a series of continuous holes, and each hole is drilled by a single pulse.

MILD STEEL: Three zones, nearly equal in width were observed across the depth of cut (material thickness) in 3.16mm mild steel. The upper band is uniformly serrated, the middle more irregular and coarse, and the lower being rather coarse, not uniform and slopping in the opposite direction to the feed (fig 4.7). Roughness ranged from 4 to 6microns Ra with no distiguishable process parameter related effects (graph 4.19) .

For the 1.6mm thick material, roughness ranged 5 to 11microns Ra, and was observed to increase with feed rate. This is clearly seen at 15Hz PRF, 0.5mm stand-off and 10Hz PRF, 1.0mm stand-off (graph 4.20). Uniform serration throughout the cut depth was seen at 10Hz PRF, while a two zone (i.e uniform and irregular) structure was generated at 15Hz PRF and 1.0mm stand-off.

STAINLESS STEEL: The cut roughness pattern is much like that of 1.6mm mild steel having two zones, uniformly serrated and irregular. Roughness rapidly increases with feed, Ra ranged 3.75 to 9microns, (graph 4.21).

TOOL STEEL: At 10Hz PRF, roughness increases with feed from 3.75microns to 9.4microns, whereas cuts at higher stand-off distance (1.0mm) were in the lower band of this Ra range. At 15Hz PRF and 1.0mm stand-off, the effect of feed rate diminished, results were still predominantly in the lower band of the range 3.2 to 9microns (graph 4.22).

TITANIUM: The roughness readings were random enough to suppress the effects of process parameters, though an increase of roughness with feed can be observed. Roughness is in the range 6 to 11.5microns Ra (graph 4.23). A two zone roughness structure was seen.

ALUMINIUM: For 1.6mm thick material, when cutting at 10Hz PRF, the cut pattern is much like that of titanium with a range of roughness of 4.5 to 7microns Ra. At 15Hz PRF the roughness pattern changes to one being more irregular and coarse across the thickness, with a roughness range of 5 to 9microns Ra (graph 4.24).

For the 0.7mm thick aluminium, a two zone pattern is observed at 15Hz PRF, and only becomes wholly irregular at 20Hz PRF. Roughness generally increases with feed in the range 3 to 8microns Ra (graph 4.25).

NICKEL: A two zone pattern was seen on all cuts. Roughness increases with feed, in the range 5 to 9microns Ra (graph 4.26).

INCONEL: All the cuts are regularly serrated through the thickness. Roughness increases with feed in the range 4 to 10microns Ra (graph 4.27).

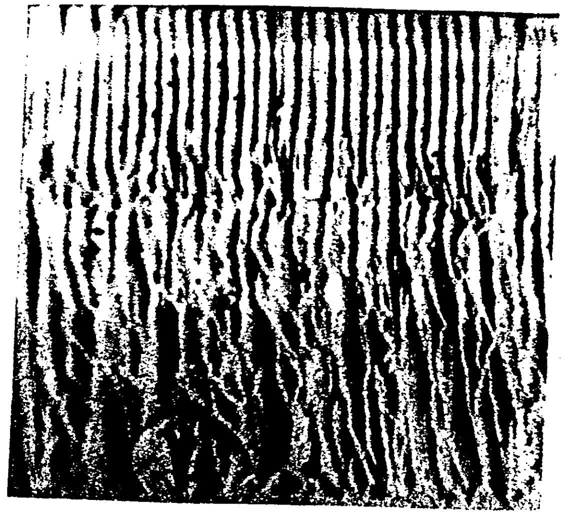


FIG 4.7 : CUT ROUGHNESS STRUCTURE OF 3.16mm MILD STEEL

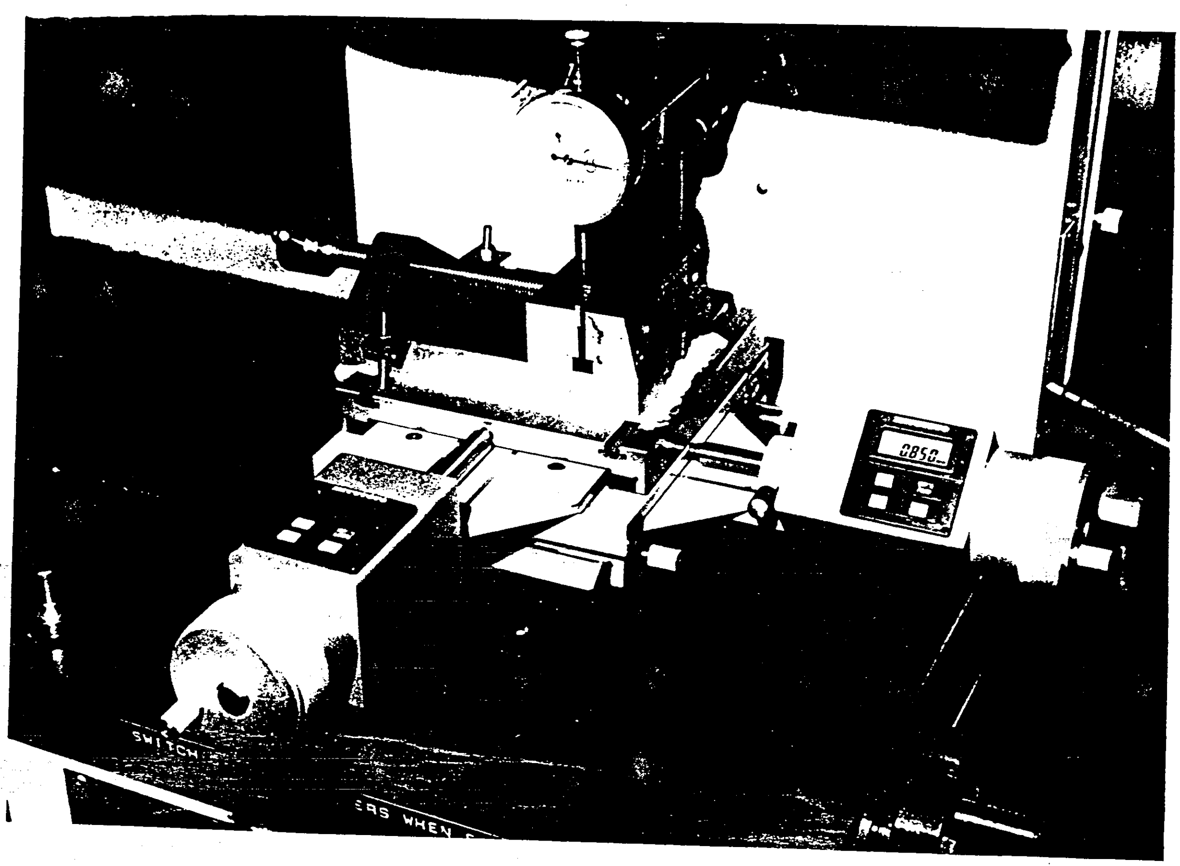


FIG 4.8 : ARRANGEMENT OF TAPER MEASUREMENT

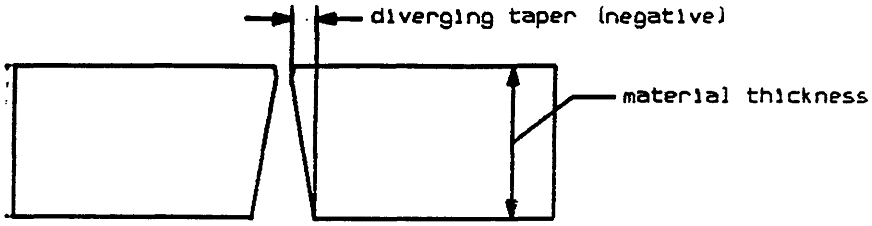
BRASS: Roughness increases with feed in the range 5 to 9microns (graph 4.28). All the cuts showed a random roughness pattern (fracture like). This structure was caused, mainly, by the rapid resolidification of molten metal on the cut edge (recast), covering the actual (serrated) cut structure. These phenomena were observed when the microstructure of the cut edges were examined to measure heat affected zone (section 4.3.2.5).

4.3.2.4 TAPER

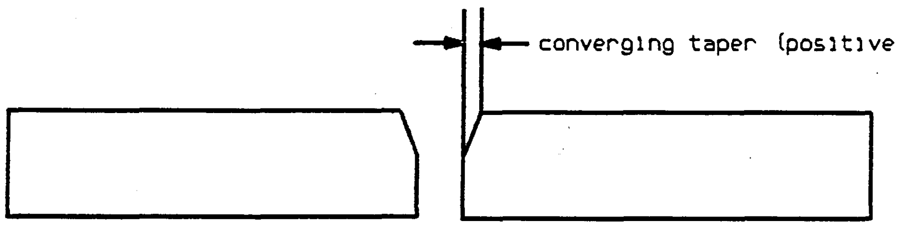
Taper is basicaly considered as the difference between the maximum and the minimum width of the cut across the material thickness, for one side of the cut (i.e half the total difference).

The arrangement of measuring taper is shown in fig 4.8. This involves mounting a dial gauge vertically above the specimen, which is held vertically in a vice with the cut edge facing the gauge tip. The vice is clamped on an X-Y table which is controlled by two digital manual micrometers.

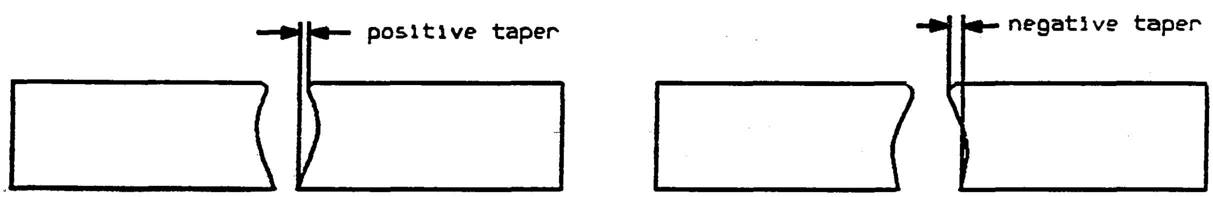
The dial gauge is scanned across the thickness at several points along the cut. The average of the difference between the highest and the lowest readings of the gauge is taken as the taper. The taper of a converging cut (wide at the top, narrow at the bottom) is recorded as positive taper, and negative for a diverging one (fig 4.9).



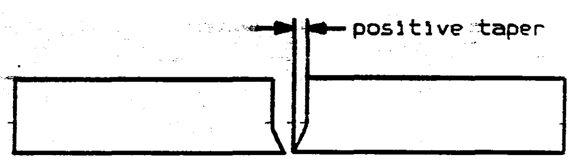
a : TAPER PROFILE OF 3.16mm MILD STEEL



b : TAPER PROFILE OF 1.6mm MILD STEEL



c : POSITVE AND NEGATIVE TAPER OF 1.6mm TOOL STEEL



d : TAPER PROFILE OF 0.9mm NICKEL

FIG 4.9 : TAPER PROFILE

MILD STEEL: 3.16mm thick mild steel exhibited negative taper for all cuts, this being in the range -20 to -70microns (graph 4.29). The cut edge profile is shown in (fig 4.9-a).

Positive taper was observed on all specimens of 1.6mm thick mild steel becoming significant at 10Hz PRF especially at higher feed rates. The range of taper is 20 to 80microns (graph 4.30). The edge profile is illustrated in (fig 4.9-b).

STAINLESS STEEL: The majority of specimens showed no taper. The remaining few of them had negative taper in the range of -20 to -40microns

TOOL STEEL: At 15Hz PRF, taper was always positive within the range of 0 to 50microns, whilst negative taper was observed at 10Hz PRF, in the range of 0 to -80microns (graph 4.31). The edge profile is shown in (fig 4.9-c).

TITANIUM: Taper increased with stand-off distance, and in the range of 20 to 100microns (graph 4.32).

ALUMINIUM: Most of the 1.6mm thick aluminium specimens, cut at 1.0mm stand-off exhibited negative taper. The range of taper for this material was in the range of -20 to 70microns (graph 4.33). No significant taper was seen on the 0.7mm thick aluminium specimens.

NICKEL: The taper was in the range of 30 to 80microns, except at 11Hz PRF and 1.0mm stand-off where taper climbed to 120microns (graph 4.34). The edge profile is shown in (fig 4.9-d).

INCONEL: Little taper (under 25microns) was observed with this material.

BRASS: Lower stand-off distance and higher pulse energy (at 10Hz PRF), tend to cause taper to be more negative. The range of taper observed was -80 to 40microns (graph 4.35).

4.3.2.5 HEAT AFFECTED ZONE

Bands of surface oxidisation, whenever seen along cut edges, where recorded as it was thought that these might indicate Heat Affected Zone (HAZ) (ref 4.7). However, investigation of the microstructure of the cut edges of four samples from each material, confirmed no link between these phenomena. For example, a sharp oxidisation strip was seen on all nickel and inconel samples, whilst an examination of the microstructure indicated no HAZ. Consequently, a microstructure examination was performed on all samples to collect HAZ data and information as to the structure of the cut edge.

Preparation for these microstructure examinations was performed according to procedures described in the Metals Handbook (ref 4.8). These steps were: sampling, mounting, grinding, polishing, etching and microscopic examination.

Sampling was performed by selecting cut pieces that lie between two adjacent cuts, thus one side of each cut was examined.

Every sample was mounted in a bakelite mould 20mm diameter and 20 to 25mm height. The sample was then sequentially ground on a hand grinder using four grades of grinding paper: 240, 320, 400 and 600. Polishing was done on an automatic lapping machine using napped cloth, 1micron diamond polishing paste and lubricant. Etching techniques will be described for each material individually.

Microstructure examination was performed with a metallurgical microscope with a provision for mounting a photographic camera. A measuring eyepiece of 10X magnification with a micrometer that moves a crosshair in the vision field, and an 11X object lens was used to measure the HAZ (i.e total magnification is 110X). various magnifications were used for micro-photography of the samples, according to the level of detail required.

MILD STEEL: Etching was made with a 2% nital solution (2 ml HNO_3 and 98ml water), for 60 to 90 seconds.

The HAZ is characterized by a distinct martinsite rich band, following the cut edge profile (fig 4.10), with an average width of 45 to 55microns, for both 3.16 and 1.6mm thicknesses. 1.6mm thick material showed metal recast on the cut edges (i.e molten metal resolidification) at slow feeds.

STAINLESS STEEL: This was etched in a solution of: 10ml HNO3 10ml acetic acid, 15ml HCl and 5 drops of glycerol, for 90sec. Solution preparation and etching has to be performed in a fume hood because of the poisonous fumes from the etchant. No significant HAZ was recorded (fig 4.11).

TOOL STEEL: This was etched in 2% nital for 90sec.

HAZ was in the range of 30 to 50microns. Metal recast was observed on all cuts, due to the rapid resolidifaction of molten metal before ejection by the jet stream of the assistant gas (fig 4.12).

TITANIUM: Titanium samples were etched in a solution of: 60ml HCl, 30ml HNO3, 5ml HF and 5ml water, for 70 to 90sec.

HAZ is characterised by a reduction in grain size to that of the bulk of the material (fig 4.13). The range of HAZ was 85 to 100microns for 15Hz PRF, and 95 to 120microns for higher pulse energy, i.e at 10Hz PRF (graph 4.36).

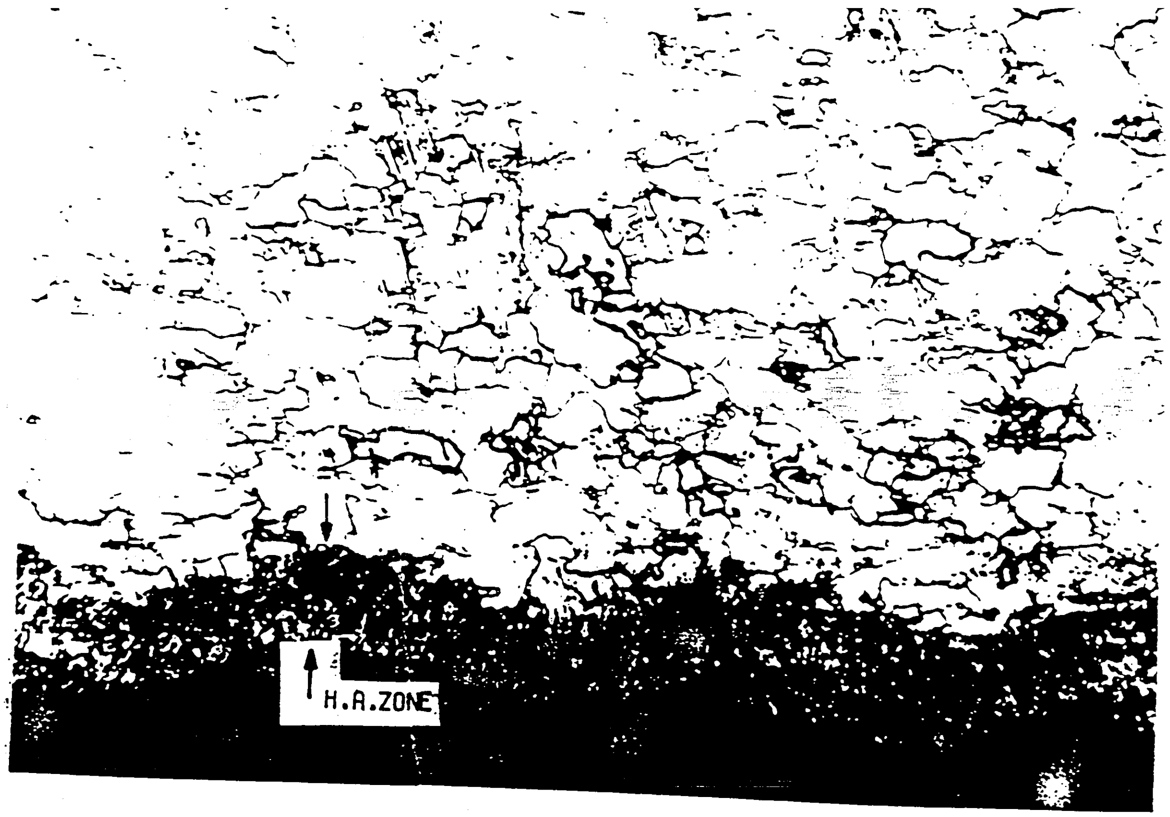


FIG 4.10 : HEAT AFFECTED ZONE IN MILD STEEL

FIG 4.11 : CUT EDGE MICROSTRUCTURE IN STAINLESS STEEL

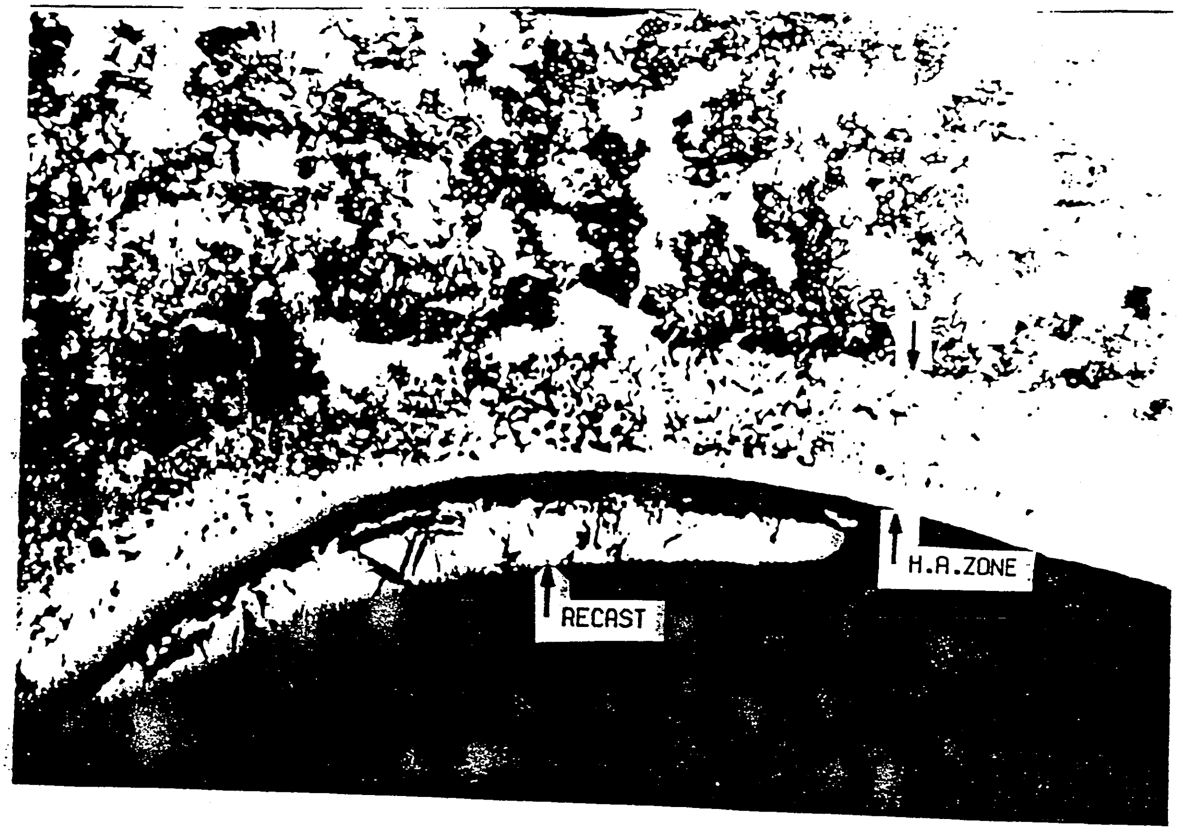


FIG 4.12 : HEAT AFFECTED ZONE IN TOOL STEEL

FIG 4.13 : HEAT AFFECTED ZONE IN TITANIUM

ALUMINIUM: The etching solution was the same as that used for titanium, etching time is 15 to 20sec.

All samples revealed a narrow HAZ, within the range of 20 to 35microns on 1.6mm thick samples, and 10 to 23microns on 0.7mm thick samples. Resolidified metal (recast) was observed on many specimens (fig 4.14). HAZ and recast widths were nearly equal in all samples examined.

NICKEL: The required etchant is a solution of: 50ml HNO_3 and 50ml acetic acid, for 40 to 50sec. No HAZ was observed on any sample (fig 4.15).

INCONEL: The same etchant for nickel is used here, for 40 to 50sec. No HAZ was observed on any sample (fig 4.16).

BRASS: The etchant is a dicromate Solution composed of: 2g Potasium Dicromate ($K_2Cr_2O_7$), 4ml saturated solution of NaCl, 8ml H2SO4 and 100ml water, is used to immerse the sample for 35sec and is followed by 10sec swabbing.

The cut edge of all the brass samples were characterised by a distinctly bound HAZ following the cut edge and metal recast (fig 4.17). These features became more serious at slow feeds and high stand-off where overlapping between the adjacent HAZ´s occurred resulting in a margin of multi-layer HAZ (fig 4.18). The HAZ layer ranged from 30 to 55microns (graph 4.37).

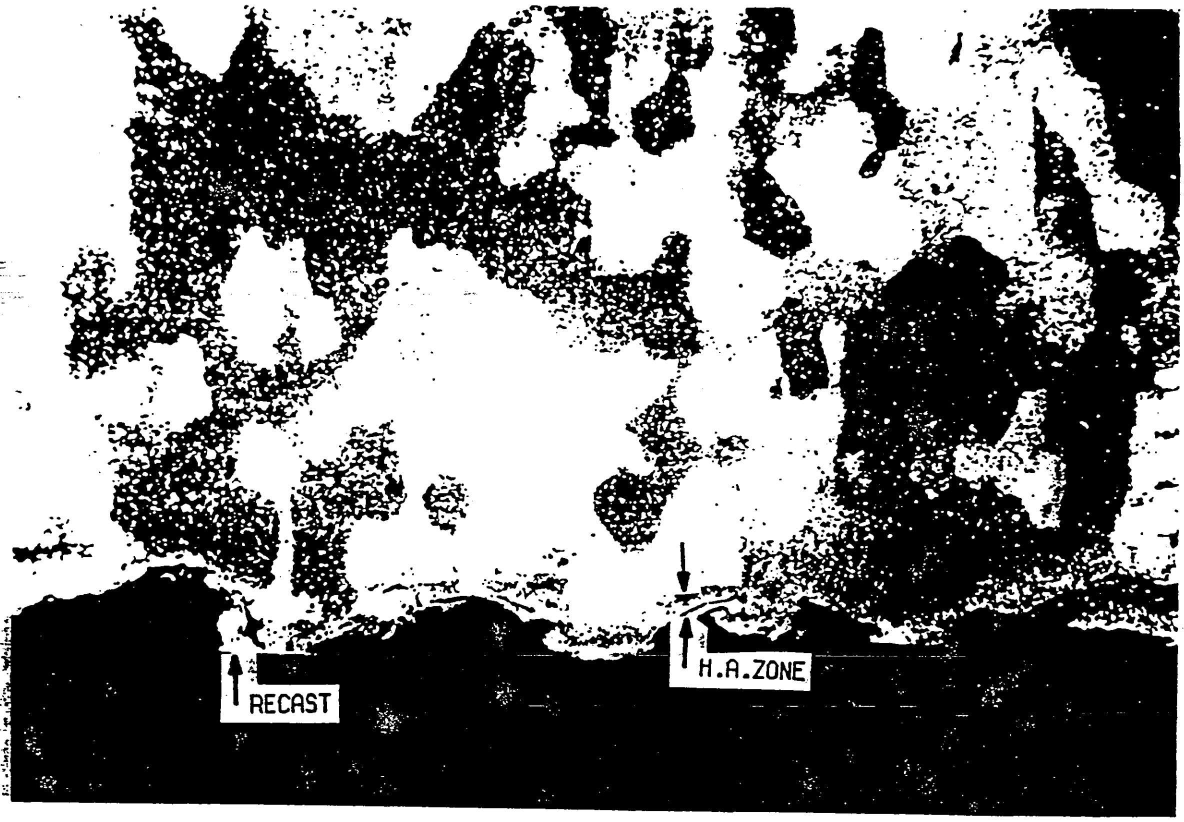


FIG 4.14 : HEAT AFFECTED ZONE IN ALUMINIUM

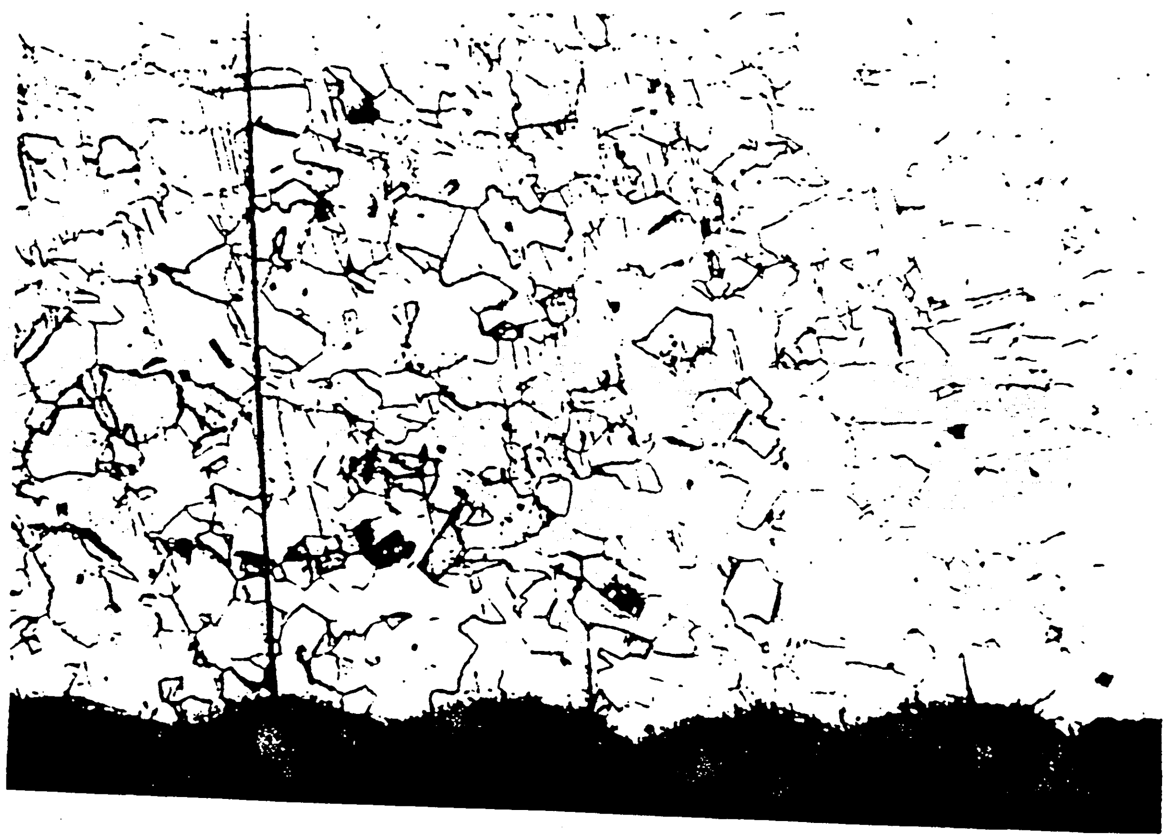


FIG 4.15 : CUT EDGE MICROSTRUCTURE IN NICKEL

FIG 4.16 : CUT EDGE MICROSTRUCTURE IN INCONEL

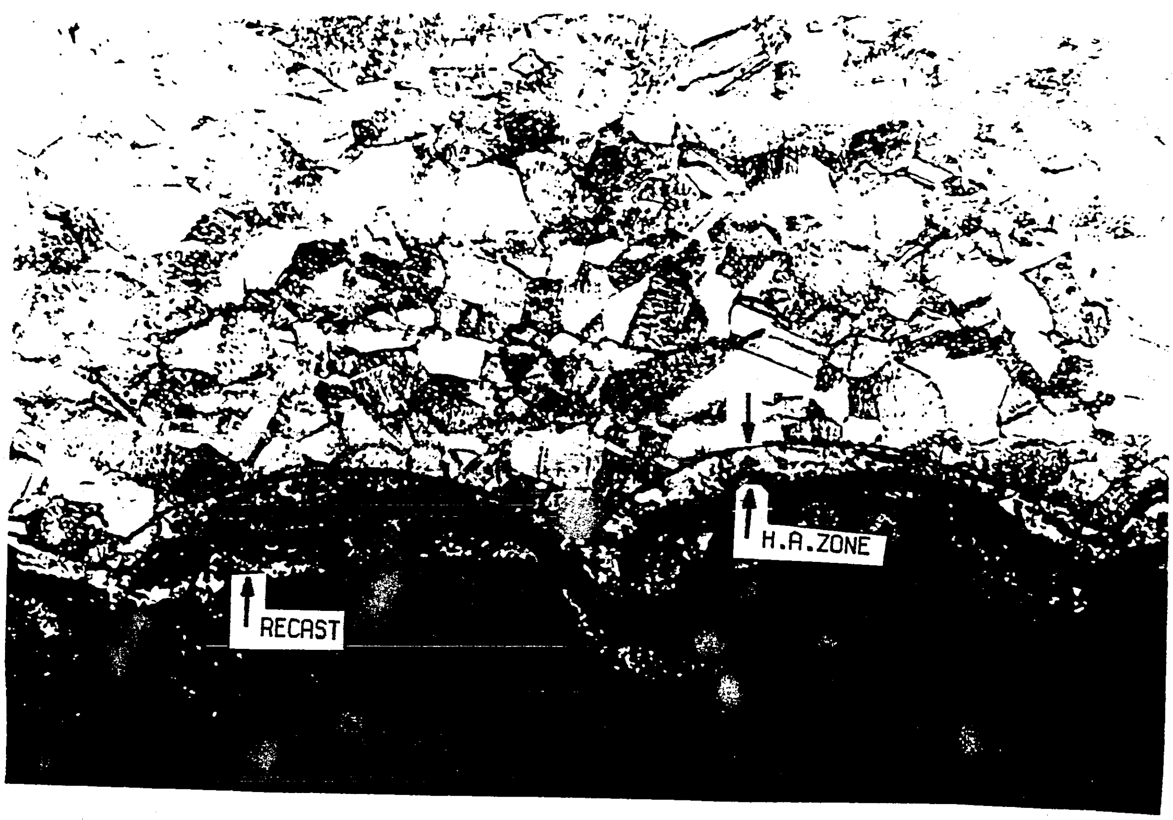


FIG 4.17 : HEAT AFFECTED ZONE IN BRASS AT FAST FEED

FIG 4.18 : HEAT AFFECTED ZONE IN BRASS AT SLOW FEED

4.4 DESCRIPTION OF THE CO_2 LASER BASED CUTTING MACHINERY

The CO_2 laser based manufacturing cell consists of two sub-systems: A CO_2 gas laser and a computer numerical controlled X-Y work handling system. The cell configuration is shown in fig 4.19.

4.4.1 THE CO_2 GAS LASER

This is a Continuous Wave (CW), Carbon Dioxide gas laser, type CLL 500 made by CONTROL LASER LTD. It emits at 10.6 microns, and is designed for industrial applications: cutting, welding and surface engineering (e.g heat treatment). The maximum nominal power is 500 Watts, but no more than 210 Watts were obtained at the time of experimentation. The laser employs fast axial gas flow principles, and uses a mixture of helium, nitrogen and carbon dioxide.

The laser system comprises of a rigid framework on which the following modules are mounted: top bench (optical cavity), a gas circulation blower, a vacuum pump, two heat exchangers, high voltage power supply (HV tank), various services and control circuitry.

The top bench has four horizontal discharge tubes mounted in series to form an optically connected path, with a pin

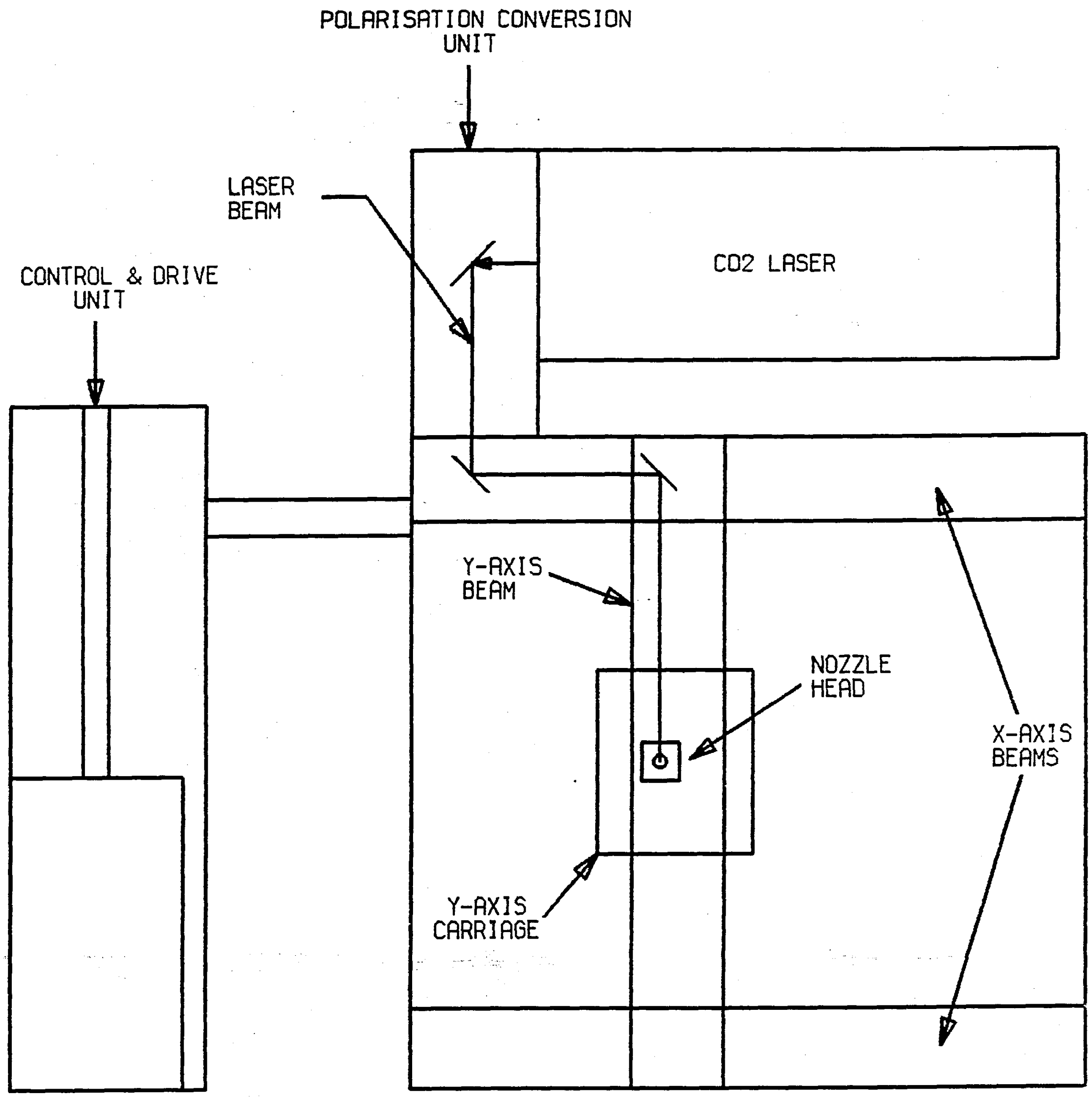


FIG 4.19 : CONFIGURATION OF THE CO2 LASER CUTTING MACHINERY

anode mounted on each of them. The optical path is contained and connected by four mirrors, one at each end and two (45 degrees) for making a 180 degrees fold in the unit (fig 4.20). The output window is partially transmissive at the lasing wavelength, and the rear mirror is concave. Four cathodes are contained in a water cooled central cathode box, which incorporates a gas manifold through which the laser gas from the discharge tubes is exhausted to the upper heat exchanger. This partially removes the heat generated in the discharge tubes.

The blower, which is of roots type, draws the gas from the top bench. The gas then passes through a lower heat exchanger and returns to the discharge tubes.

The HV tank supplies the four discharge tubes with the high voltage electrical power required for laser excitation.

The electrical discharge causes decomposition of the CO2 and nitrogen in the laser gas. This phenomena is rectified by a vacuum pump continuously drawing off a proportion of the circulating mixture, and delivering part of this to a catalytic regeneration unit, and exhausts the other part which is replaced by the admission of fresh gases to the system.

The control circuitry monitors and controls all the laser functions. This is operated via a control panel, which

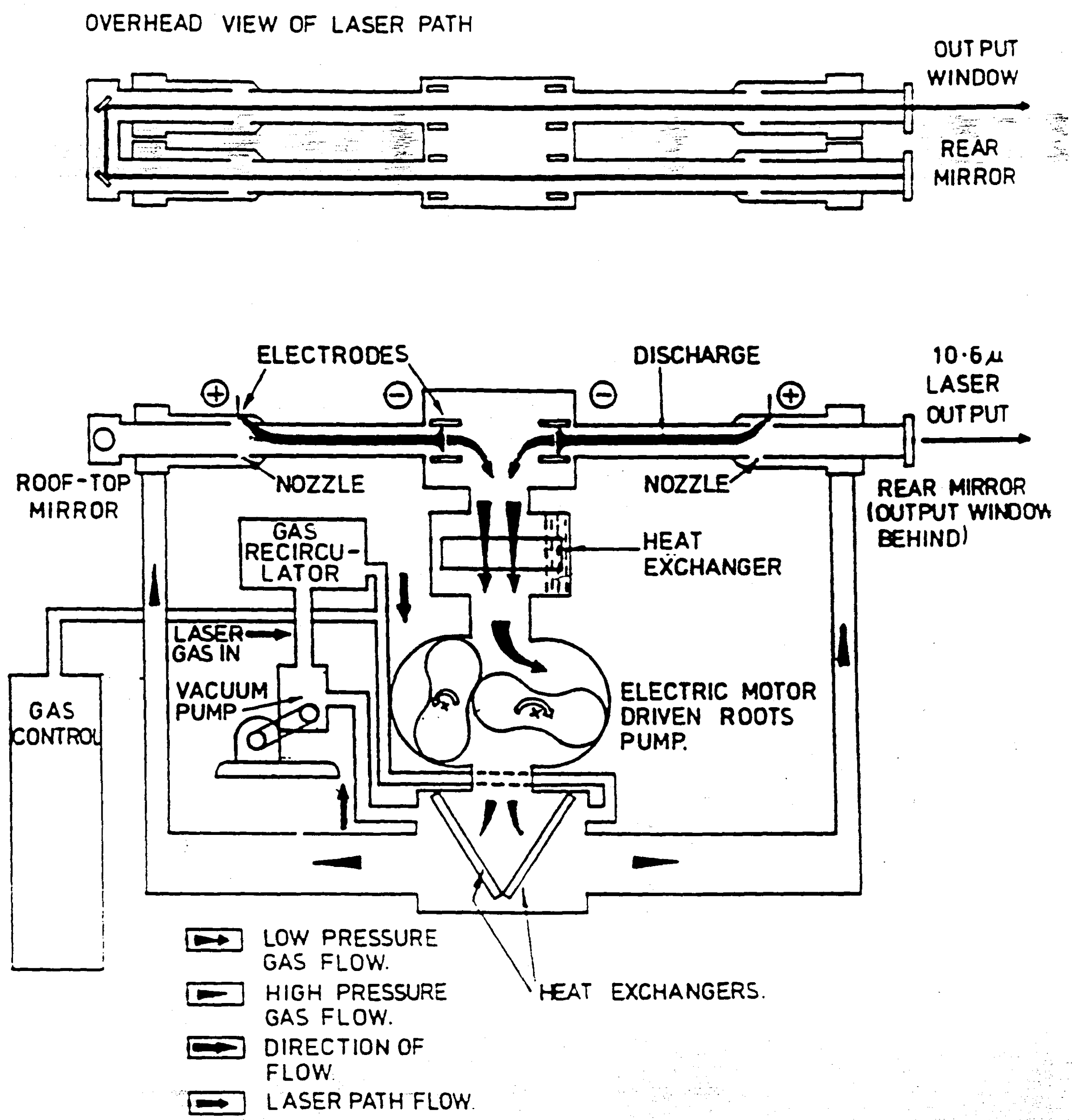


FIG 4.20 : LASER PATH AND GAS FLOW OF THE CO_2 LASER

contains the switches for startup, power control and shutdown.

A shutter assembly is positioned on the output side of the output window, a 45 degrees mirror is used to steer the laser beam downwards into a calorimeter cone, when the shutter is closed. A metered flow of water cools the calorimeter cone, and the water temperature difference between the inlet and outlet is measured by sensors, converted into laser power in Watts and displayed on a digital readout. The cooling circuit uses mains water, and supplies the cathode box, output window, rear mirror, and the shutter mirror. The two folding mirrors are cooled by oil which is pumped to them via a water-oil heat exchanger.

4.4.2 THE WORK HANDLING SYSTEM

This system consists of: two X-axis frames, Y-axis cross beam, work bed, control and drive unit, and laser beam delivery arrangement (fig 4.19).

Each of the two X-axes incorporates a sliding carriage moving on top of a rigid support powered by a servomotor, allowing a travel 2500mm in the X direction.

The Y-axis cross beam is supported by the X-axis sliding carriages and incorporates a slideway on which a carriage, carrying the nozzle assembly, is powered by a servomotor, to travel 1500mm in the Y direction. The nozzle assembly is capable of 50mm vertical movement by pneumatic actuation.

The work bed is a free standing table supporting the work piece by an array of cone headed aluminium supports.

The control and drive unit contains a CNC control unit, drive cards for the servomotors and a gas services panel. The CNC unit (type HEIDENHAIN TNC 155A) organises the nozzle head movement according to a predefined set of instructions. This is programmed using two languages: ISO 6983 format (standard G and M codes), and the Heidenhain plain dialogue format. Part programs can be fed to the CNC memory either manually using the keyboard, or via an RS 232 communication interface from a computer, tape unit or other suitable source.

The gas panel is for the filteration, distribution and control of the compressed air and cutting assistant gases.

The output laser beam is directed and focused by means of a beam delivery system of mirrors and focusing lens. The beam leaving the laser machine, is folded through 180 degrees and directed along the X-axis by two phase retarding mirrors which also convert its polarisation from linear to circular. The beam is then intercepted by a mirror mounted on the cross beam and deflected along the Y-axis. Another mirror mounted above the nozzle head, deflects the beam downwards onto the focusing lens. This lens (Zn-Se, 150mm focal length) focuses the beam onto the work surface through the nozzle. A copper nozzle with 1.2mm diameter orifice directs the cutting assistant gas, which also helps to clean and cool the lens, is ejected through the nozzle coaxially with the laser beam.

4.5 THE CO_2 LASER CUTTING EXPERIMENTS

The parameters studied in this series of experiments were: material type and thickness, feed rate, laser power and assistant gas pressure.

Stand-off distance was kept constant at 1.0mm during all the experiments due to the considerably larger depth of focus of the CO_2 cutting system compared to that of the YAG system. The depth of focus (Z) is given by ref 4.2 as:

$$Z = (4\ L\ F^2\ /\ D^2) = 6.6\ \text{mm}$$

where L is wavelength, F is the lens focal length and D is the beam diameter at the lens.

This should be compared to the focus depth of the YAG laser which is approximately 120 microns.

The materials titanium, aluminium, nickel and brass, which were cut with the YAG laser, cannot be cut with this laser, mainly due to the limited power intensity of the CO_2 laser (10^5 W/cm^2) compared with that of the YAG laser. This is not caused by low power (compared with 50W average power of the YAG laser), but is, mainly, due to larger focal spot size of CO_2 laser (see section 4.1.1). Additionally, the pulse energy of the YAG laser is discharged within a short time (1 to 1.5 millisecond) which results in a very high beam intensity. Another phenomena affecting this process, is that metallic materials have higher reflectivities at the longer wavelength produced by CO_2 laser (chapter 1).

It was not possible to cut 3.16mm thick mild steel and 0.7mm thick inconel with a power less than 200W (i.e maximum achievable power) due to the limitations above. Thus 0.8mm thick mild steel was introduced to provide data for the process of interpolation detailed in section 4.2.1.

The set of cutting experiment for each material comprised: four feed rates, two power levels, and three gas pressures. (i.e 24 cutting trials per material). The layout of a sample test plate is shown in fig 4.21.

The work sheet for these tests is much the same as that for the Nd:YAG laser tests (fig 4.5), but without columns for pulse parameters.

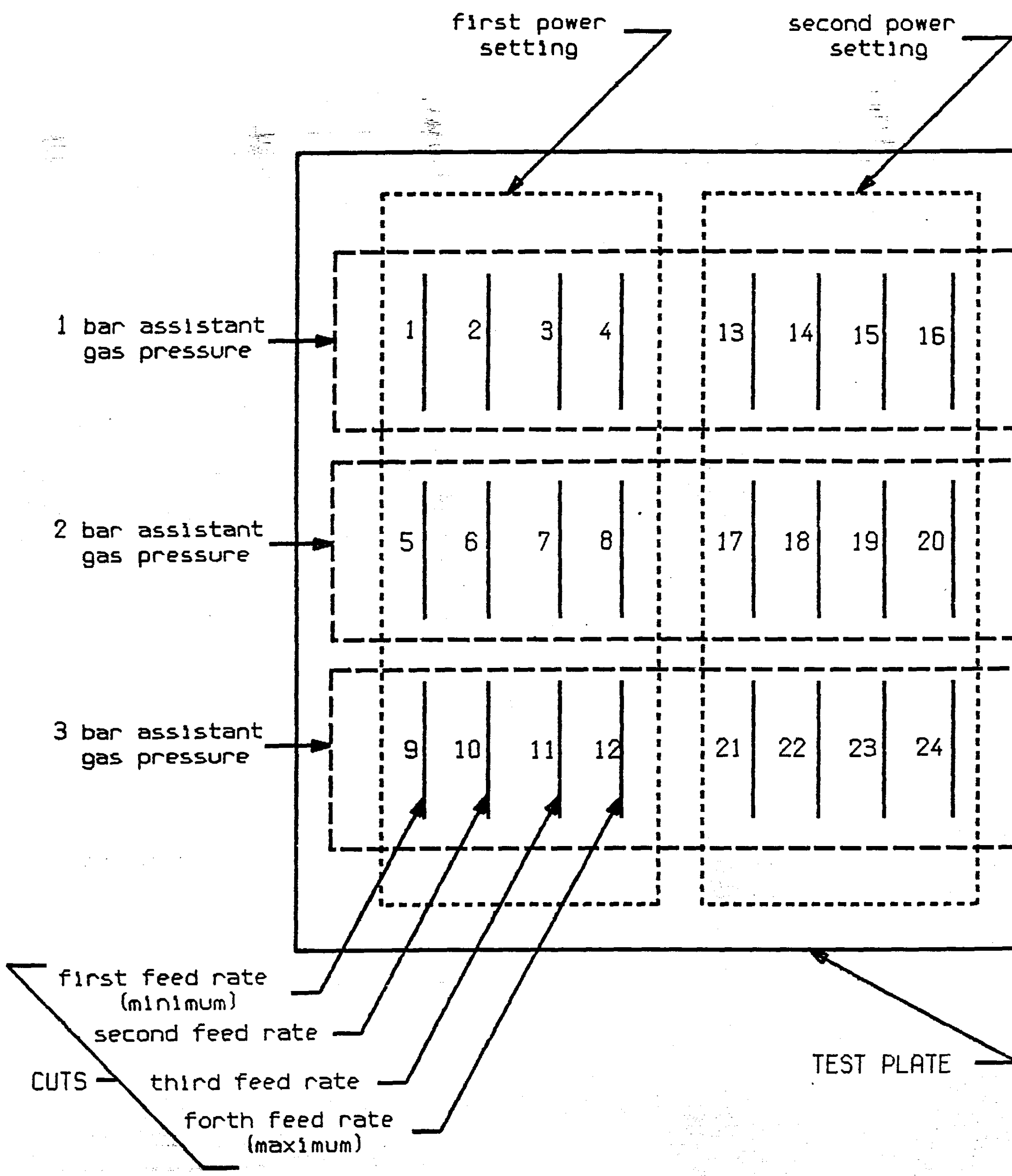


FIG 4.21 : TEST PLATE LAYOUT FOR CO2 LASER CUTTING EXPERIMENTS

4.6 POST PROCESS MEASUREMENTS AND ANALYSIS OF CUT QUALITY

The same principles and procedures followed in section 4.3 are applied here.

4.6.1 PRIMARY VERIFICATION

MILD STEEL: 3.16mm thick mild steel was reluctantly cut with 1bar oxygen at 200W. Full penetration was not achieved on one cut, and frequent blowthroughs were observed in others.

Cutting at 4bars of oxygen pressure was attempted, and normal cuts up to 180mm/min feed were observed. At 250mm/min (the maximum possible feed), a very wide irregular cut was formed. This result was not counted as being laser cut, since it was almost wholly formed by material-oxygen reaction (burning).

1.6mm thick material was easily slit with 200W power. However, cutting failed with 1bar oxygen and a power of 120W, while unstable cutting was observed at 2bars. The maximum achievable feed at this power level was 300mm/min, which is only 25% of that at 200W.

No problems were encountered when slitting 0.8mm thick mild steel. This material was cut with feeds up to 3000mm/min at 200W, and up to 1400mm/min at 100W.

Generally, mild steel exhibited a phenomena of large inverse relationship between feed and material thickness. Where at constant power (200W), the feed was decreased by a factor of 12 (from 3000 to 250mm/min) when thickness increased by a factor of 4 (from 0.8 to 3.16mm).

STAINLESS STEEL: No difficulties were experienced in cutting this material with a power level greater than 160W. A maximum feed of 500mm/min was achieved with power of 200w, while the maximum feed achieved with 160W power was 200mm/min.

INCONEL: This 0.75mm thick material was unable to be cut with power less than 200W, and at feed rates greater than 75mm/min.

4.6.2 ANALYSIS OF CUT QUALITY

All the procedures of measurement and result presentation used in section 4.3.2, are applied here. Each of the result presentation graphs, basically contains 12 data points lying on three curves, representing the variation of one quality feature against one level of power, and three gas pressures.

4.6.2.1 KERF WIDTH

MILD STEEL: The average range of kerf widths on 3.16mm thick material was 330 to 360micron. At 1bar oxygen it is a little higher (370 to 385microns). From graph 4.38 it seems that no other parameter influenced kerf width.

For 1.6mm, higher settings of feed and gas pressure lead to narrower kerfs when cutting with 200W. With 120W, kerf slightly widens with feed. Generally, kerf widths are in the range of 300 to 380microns (graph 4.39).

When cutting 0.8mm thick mild steel with 200W power, higher feed produced slits with narrower kerf widths. Cutting at 2bars oxygen was found to yield the narrowest kerf from the range of 340 to 400microns. The kerf width range is 290 to 330microns with 100W, and seems to be insensitive to any parameter (graph 4.40).

STAINLESS STEEL: From graph 4.41 it is obvious that with higher feed, narrower kerf was observed. Higher power tends to have the same effect. The kerfs are in the range 300 to 390microns.

INCONEL: 2bars oxygen appears to give the narrowest kerf from the range 280 to 340microns (graph 4.42).

4.6.2.2 DROSS FORMATION

MILD STEEL: Except at 1bar oxygen where dross size rose to 40microns, all other samples of 3.16mm thick mild steel produced dross heights in the range 10 to 20 microns.

For 1.6mm, the dross size range is 10 to 20microns except when cutting with 200W and 3bars oxygen where it was 120 to 150microns.

For 0.8mm thick, 2bars oxygen was ideal for minimum dross formation at both power levels. Larger dross was noticed on samples cut with 200W. The range of dross size is 20 to 200microns for 200W, and 20 to 120microns at 100W power (graph 4.43).

STAINLESS STEEL: Dross attachment is a serious problem when cutting stainless steel (ref 4.9, 4.10 and 4.11). Dross free cutting was found to be achievable at a feed 25% of the maximum attainable. At this feed the dross is loosely attached and drops off with normal handling. With higher feeds the dross becomes more persistently attached but smaller in size. Less power resulted in larger but less persistent dross attachment. The range of dross size was 300 to 700microns (graph 4.44) .

Samples of the loose dross were collected, ground and tested by a diffraction X-ray spectrometer. The test revealed that the formation was wholly chromium and iron oxides.

INCONEL: No noticable dross was formed on inconel samples.

4.6.2.3 CUT ROUGHNESS

MILD STEEL: The cut edges of all mild steel samples featured a two zone surface structure. A regularly serrated band occupying the upper 1/3 to 1/2 of the material thickness, and a coarser more irregularly serrated lower band.

The serration pitch on all mild steel samples was measured, and was, generally, within the range 125 to 150microns. Power, feed and assistant gas pressure seem to have combined effects on the serration forming mechanism. This leads to no direct and simple correlation between this feature and process parameters being easily identified.

For 3.16mm thick mild steel, roughness (Ra) constantly increases with assistant gas pressure. Feeds in the range 120 to 180mm/min (1/2 to 2/3 of maximum feed) produced smoother cuts. Roughness was in the range of 2.5 to 5.5microns (graph 4.45).

Oxygen pressure appears to have similar effects on 1.6mm mild steel. Roughness tends to improve with higher feed or with less power. The roughness values varied over the range 1.0 to 4.75microns (graph 4.46).

When cutting 0.8mm thick material with 200W, roughness rapidly decreased with increasing feed from 0.5 to 4microns. Higher oxygen pressures produced a marginal improvement (graph 4.47). Roughness exhibited a wide scatter at 100W power setting, being in the range of 1.25 to 1.75microns.

STAINLESS STEEL: The cut edge of all samples was unserrated i.e of matt appearance. Microscopic examination showed that the cut edge was originally serrated, but that these serrations were filled with recast metal during the cutting process, before being able to be ejected by the assistant gas stream. This phenomena resulted in the matt unserrated appearance of the cut edges.

As shown in graph 4.48, the roughest edges are on the samples cut at 1bar oxygen pressure, for both power levels 210 and 160W. At this pressure Ra is 2.0 to 3.25microns for 210W, and 3.5 to 6.0microns for 160W. Roughness was in the range of 1.25 to 2.25microns for both powers at 2 and 3bars oxygen. The smoothest edges were found to be those cut at about 3/4 of the maximum achievable feed for each power.

INCONEL: The highest roughness was observed on the samples cut at 1bar oxygen pressure. Graph 4.49 shows that higher feed rate marginally improves the edge quality. the Roughness ranged from 1.0 to 1.75microns.

4.6.2.4 TAPER

MILD STEEL: 3.16mm samples cut 1 bar assistant gas pressure exhibited highly negative taper in the range of -80 to -100microns, the other samples show positive taper in the range of 20 to 45microns (graph 4.50).

For 1.6mm thick, taper increases with higher feed and gas pressure. Graph 4.51 shows that the roughness range was 10 to 45microns, except for two cuts.

Most of the 0.8mm thick mild steel samples exhibited no significant taper, 25microns being the maximum recorded.

STAINLESS STEEL: Low oxygen pressure tended to yield negative taper, while higher oxygen pressure produced higher positive taper. The taper results were in the range -80 to 120 microns (graph 4.52).

INCONEL: Taper ranged from 10 to 70microns with no clear correlation with feed or gas pressure (graph 4.53).

4.6.2.5 HEAT AFFECTED ZONE

MILD STEEL: HAZ increased marginally with feed on 3.16mm thick samples, in the range 60 to 110microns (graph 4.54).

For 1.6mm thick a slight decrease in HAZ with oxygen pressure can be observed from graph 4.55, where HAZ is in the range of 60 to 90microns.

No direct influence from process parameters can be determined from the results of the 0.8mm thick mild steel samples (graph 4.56), where HAZ is in the range of 50 to 80microns.

STAINLESS STEEL: No noticable HAZ was seen on any sample of this material.

INCONEL: No HAZ was seen on any of these samples.

4.6.3 CUTTING TRIALS ON POLYMERIC MATERIALS

These are supplementary experiments to verify the CO2 laser performance in cutting polymers. They are not intended for detailed evaluation of cut quality.

Graph 4.57 presents the summary of cutting trials made on PMMA (perspex). The relation between feed and material thickness for two powers 85 and 170W is illustrated. From graph 4.57, it can be concluded the zone around 10mm material thickness is a zone of considerable change in slope of both curves. Where these curves are featured by sharp slope at thicknesses less than 10mm, while they become shallow at greater thickness. Another phenomena can be observed from that graph, that is for thickness less than 10mm, the effect of higher power is less than that for thickness greater than 10mm.

Air was used as cutting assistant gas. Air pressure variations had no effect on the cutting performance.

Acrylic and PVC sheets of different thicknesses were also cut. The cutting parameters were found not to differ from those of perspex.

REFERENCES

REF. NO.	DETAILS
4.1	Kogelnik, H., Imaging of Optical Modes-Resonators with Internal Lenses, The Bell System Journal, Vol.44, March 1965, PP455.
4.2	Ready, J. F., Industrial Applications of Lasers, Academic Press Inc (London) Ltd, 1978.
4.3	Anon, MICON CNC 850 User´s Reference Manual, MICON Systems Ltd.
4.4	Petersen, R. G., Design and Analysis of Experiments, Marcel Dekker Inc, NewYork, 1985.
4.5	van Berkum, E. E. M., Optimal Paired Comparison Designs for Factorial Experiments, Mathematisch Centrum, Amsterdam, 1978.
4.6	Raktoe, B. L., Hedayat, A. and Federrer, W. T., Factorial Designs, John Wiley & Sons Inc, USA, 1981.
4.7	Kovalenko, V. S., Arata, Y., Maruo, H. and Miyamoto, I., Experimental Study of Cutting Different

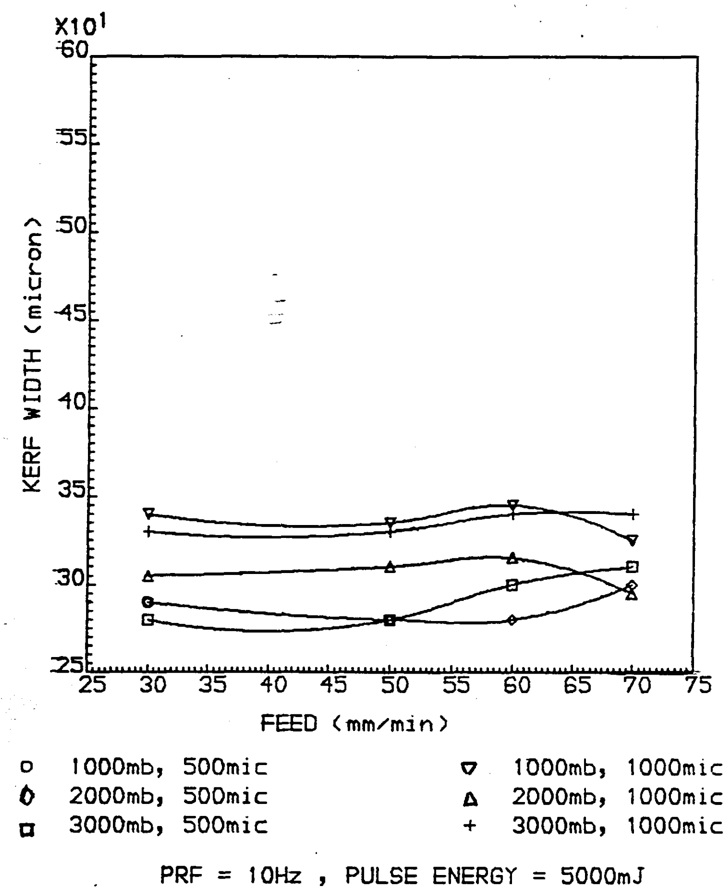
Materials with a 1.5 KW CO2 laser, Transaction of JWRI, Vol.7, No.2, 1978, PP101.

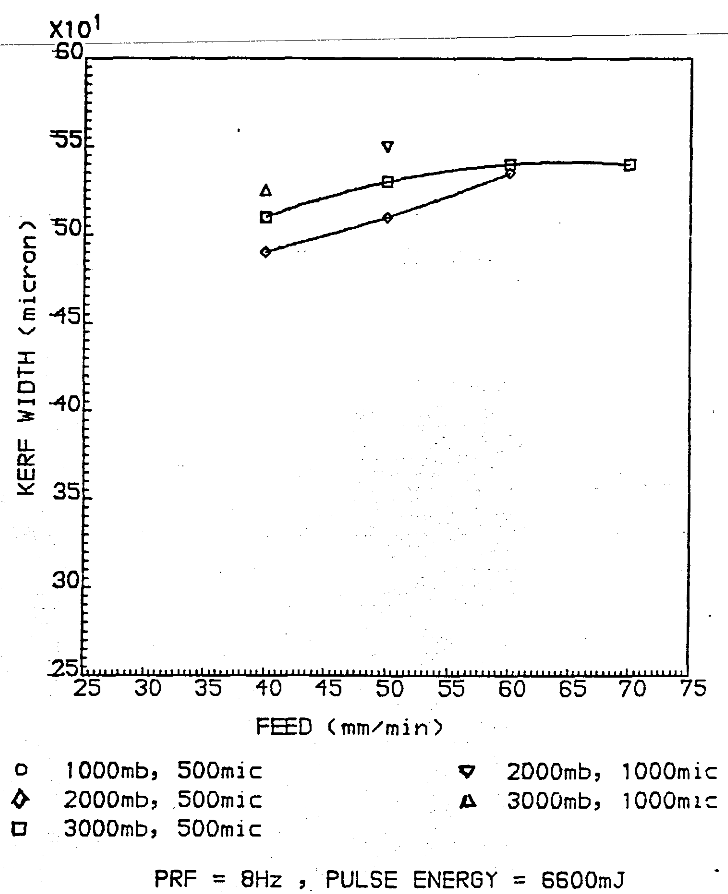
4.8 Metals Handbook, Vol.8,18th edition, Metallography and Phase Diagrams, American Society for Metals, 1973, Ohio, USA.

4.9 Birkett, F. N., Herbert, D. P. and Powell, J., Prevention of Dross Attachment During Laser Cutting, Proceedings of the Second International Conference on Lasers in Manufacturing, 26-28 Nov. 1985, Birmingham, PP63.

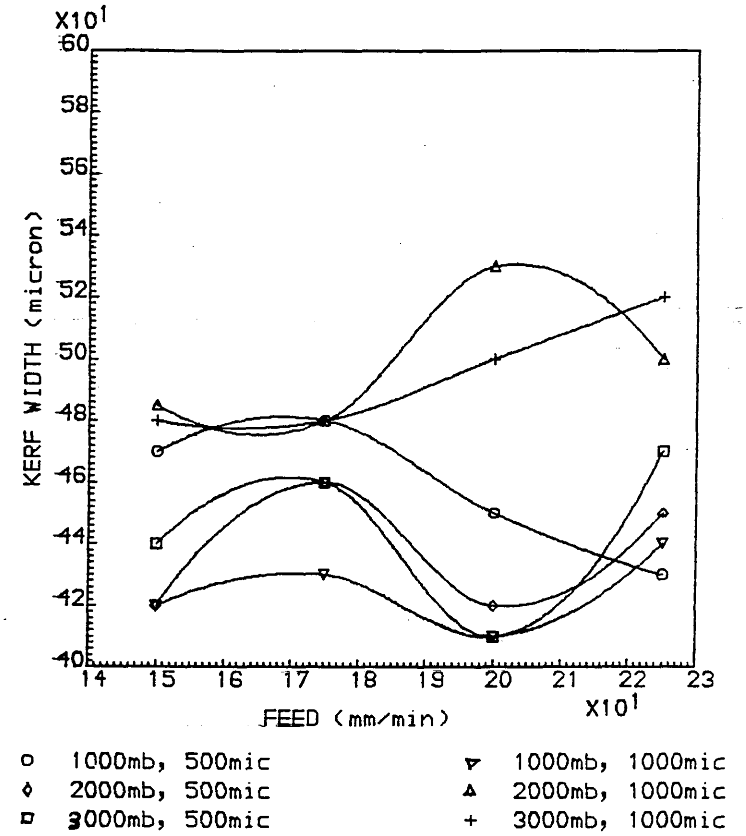
4.10 Murakawa, M., Miyazawa, H., Sakia, S. and Ohkawa, K., Prevention of Dross Attachment in Laser Cutting, Bulletin of Japan Society of Precision Engineering, Vol.22, No.1, March 1988, PP6.

4.11 Arata, Y., Maruo, H., Miyamoto, I. and Takeuchi S., Improvement of Cut Quality in Laser-Gas-Cutting Stainless Steel, Source Book on Applications of Lasers in Manufacturing, American Society for Materials (ASM) 1981, PP72.

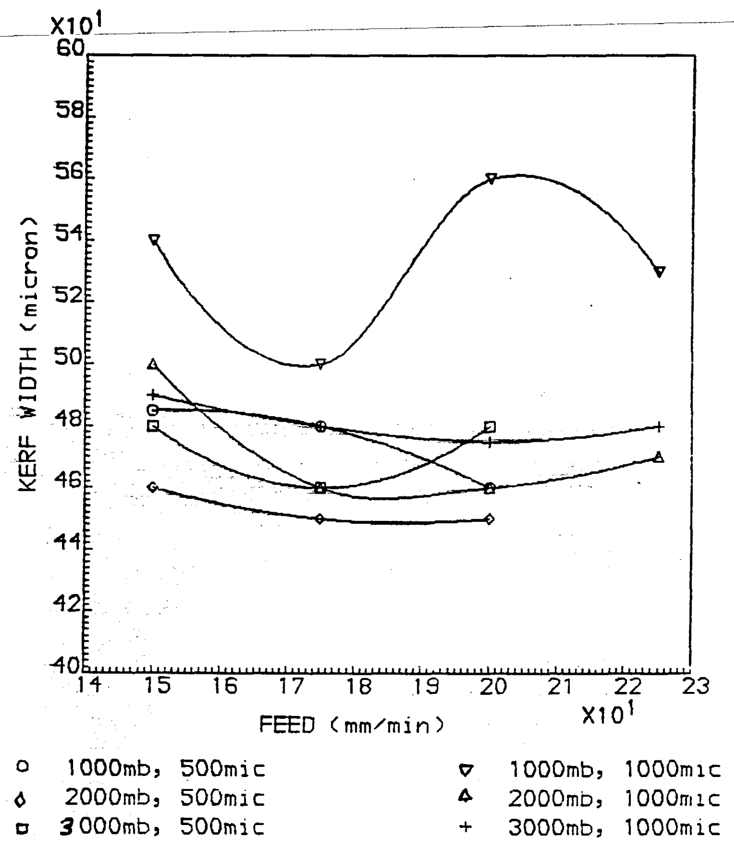




GRAPH 4.1 : KERF WIDTH Vs FEED FOR 3.16mm THICK MILD STEEL

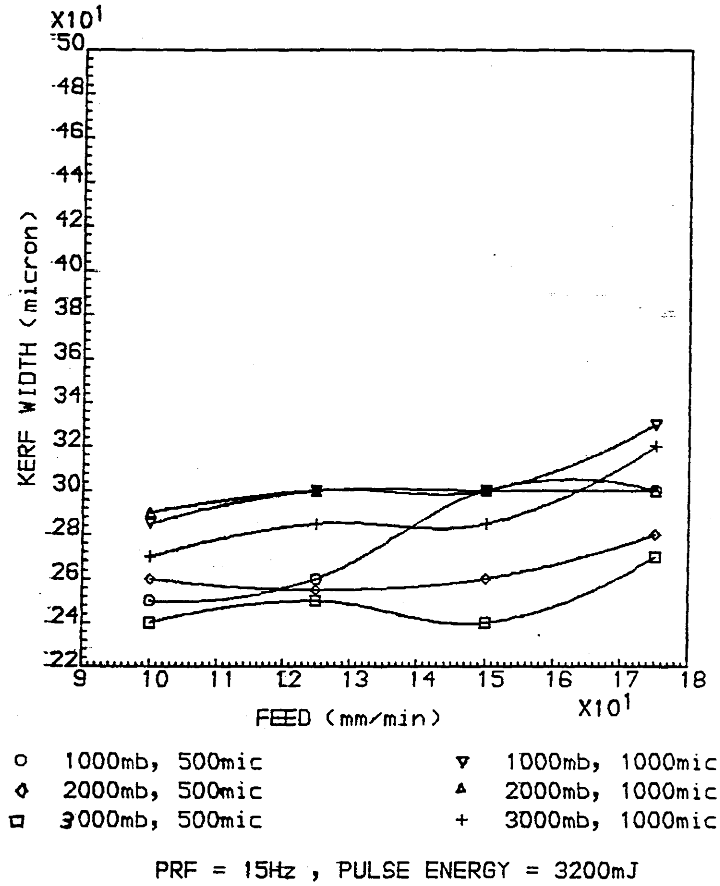


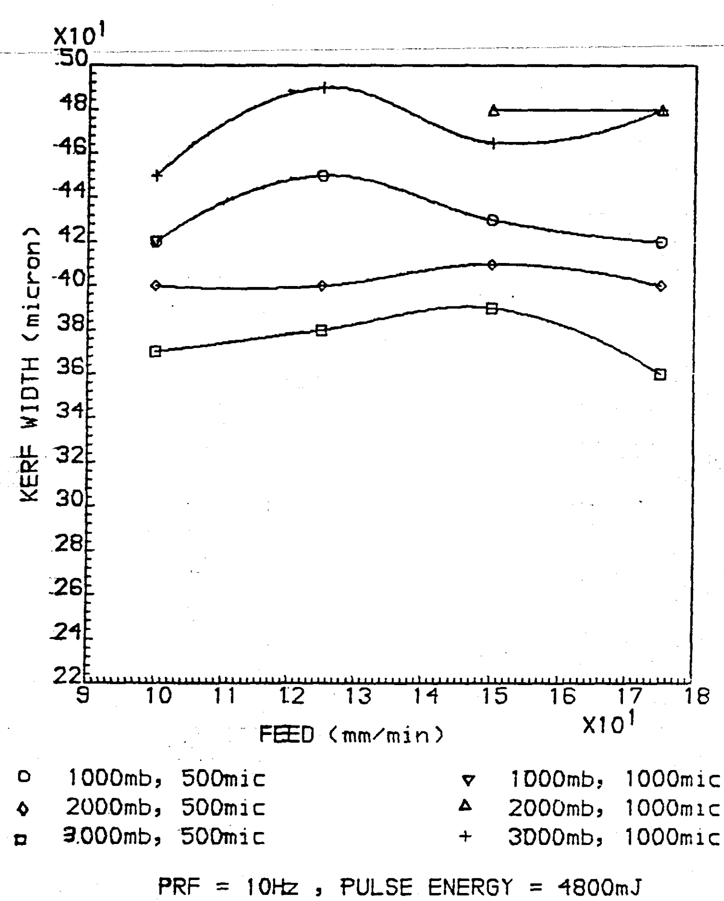
PRF = 15Hz , PULSE ENERGY = 3200mJ



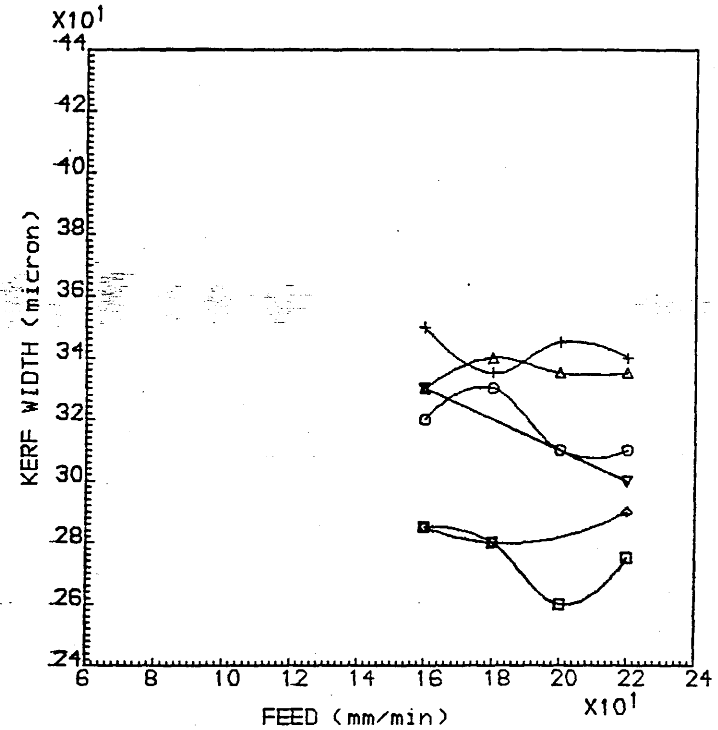
PRF = 10Hz , PULSE ENERGY = 4800mJ

GRAPH 4.2 : KERF WIDTH Vs FEED FOR 1.6mm THICK MILD STEEL



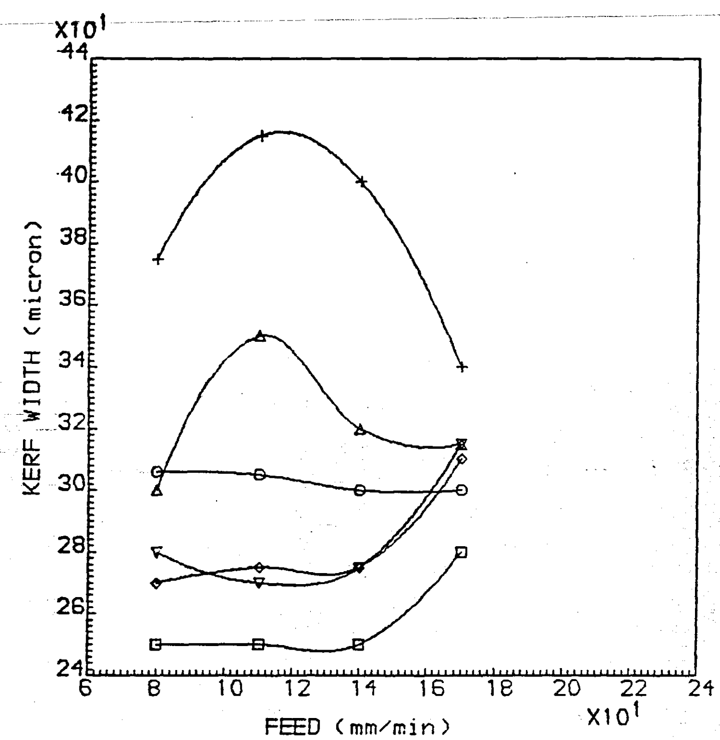


GRAPH 4.3 : KERF WIDTH Vs FEED FOR 1.6mm THICK STAINLESS STEEL



o 1000mb, 500mic
◇ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
△ 2000mb, 1000mic
\+ 3000mb, 1000mic

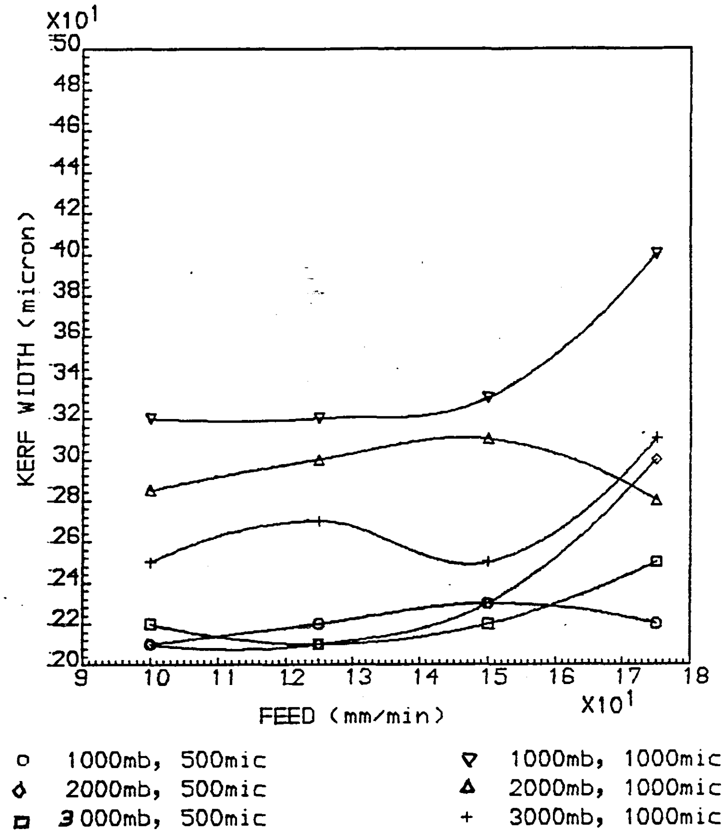
PRF = 15Hz , PULSE ENERGY = 3100mJ



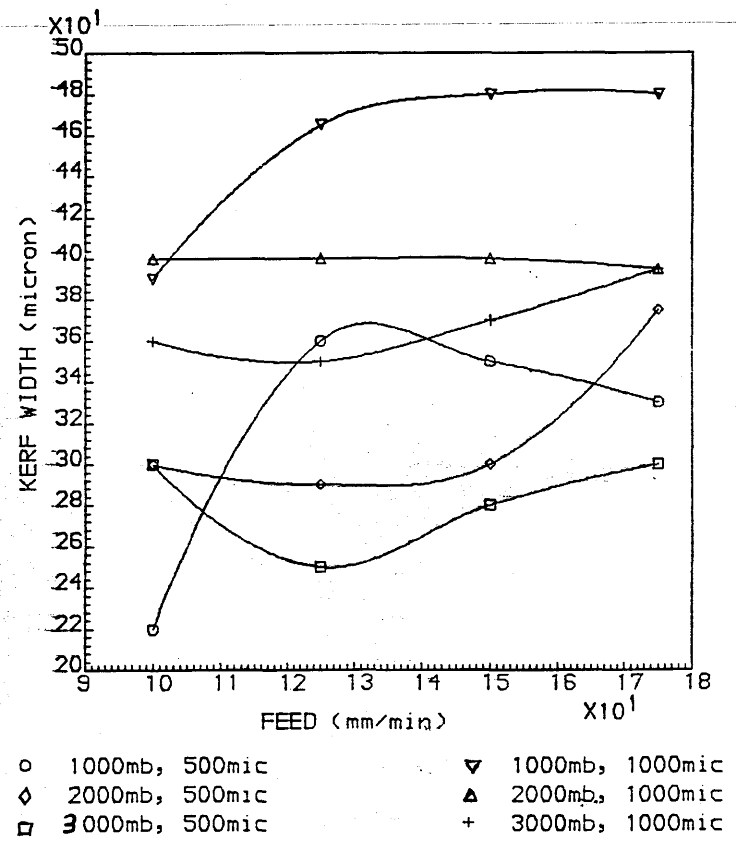
o 1000mb, 500mic
◇ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
△ 2000mb, 1000mic
\+ 3000mb, 1000mic

PRF = 10Hz , PULSE ENERGY = 5000mJ

GRAPH 4.4 : KERF WIDTH Vs FEED FOR 1.6mm THICK TOOL STEEL

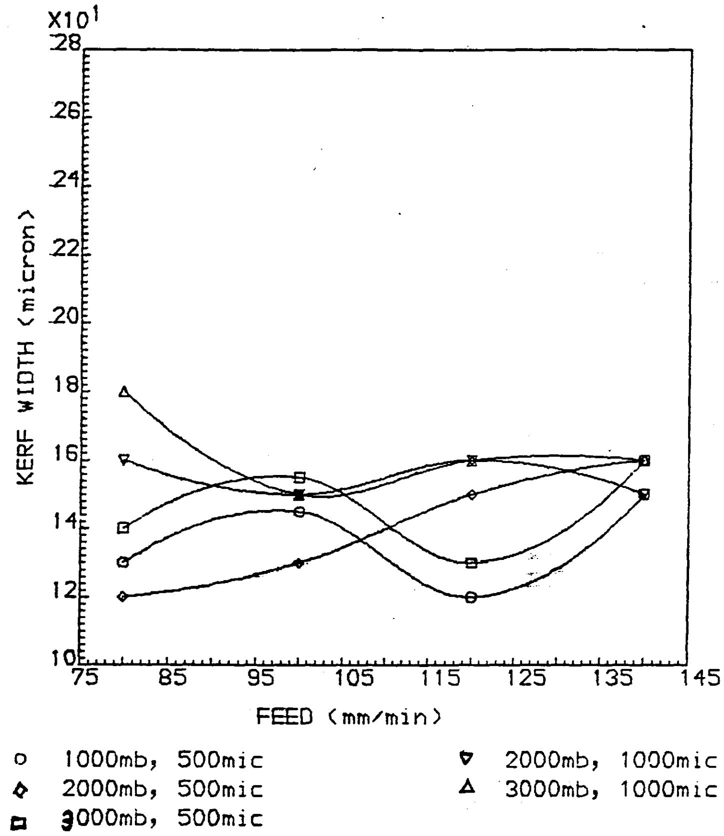


PRF = 15Hz , PULSE ENERGY = 3200mJ

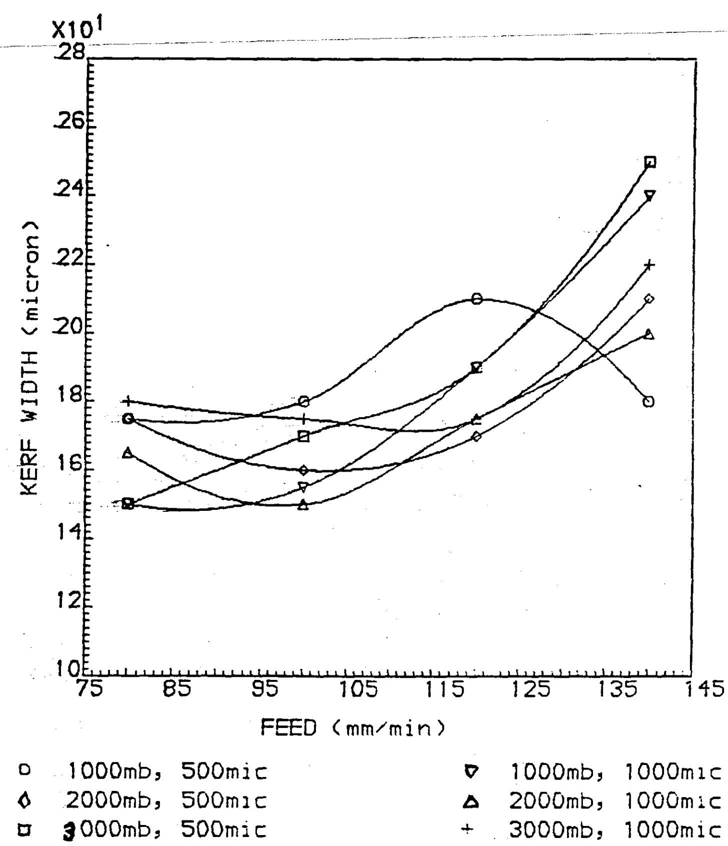


PRF = 10Hz , PULSE ENERGY = 4800mJ

GRAPH 4.5 : KERF WIDTH Vs FEED FOR 1.6mm THICK TITANIUM

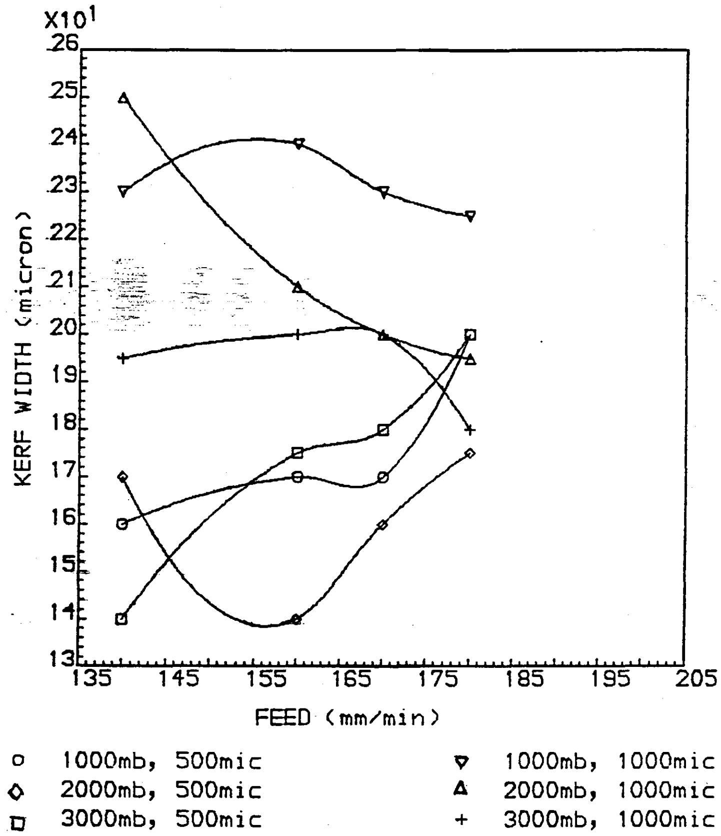


PRF = 15Hz , PULSE ENERGY = 3200mJ

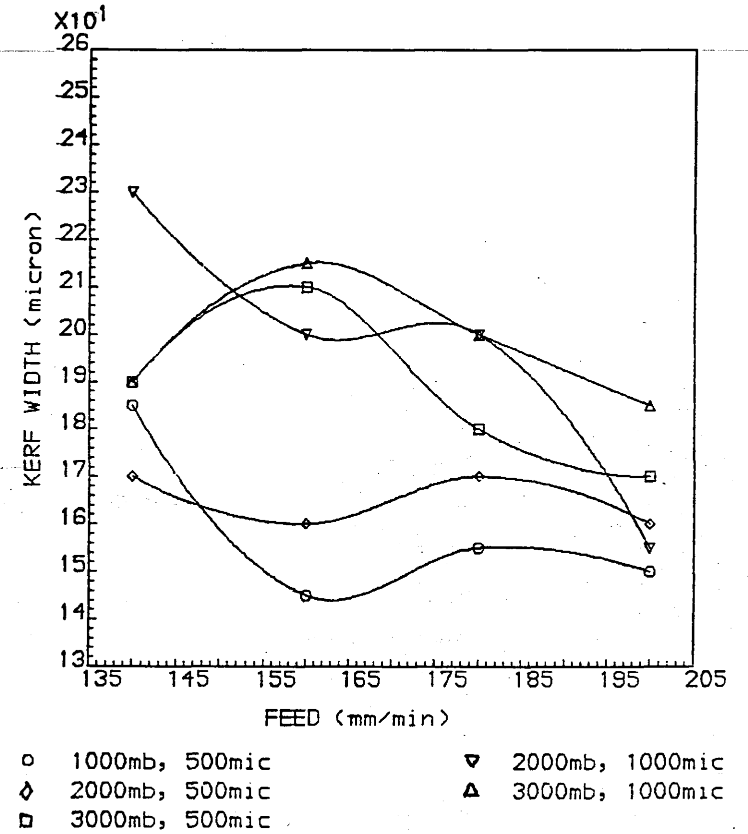


PRF = 10Hz , PULSE ENERGY = 5000mJ

GRAPH 4.6 : KERF WIDTH Vs FEED FOR 1.6mm THICK ALUMINIUM

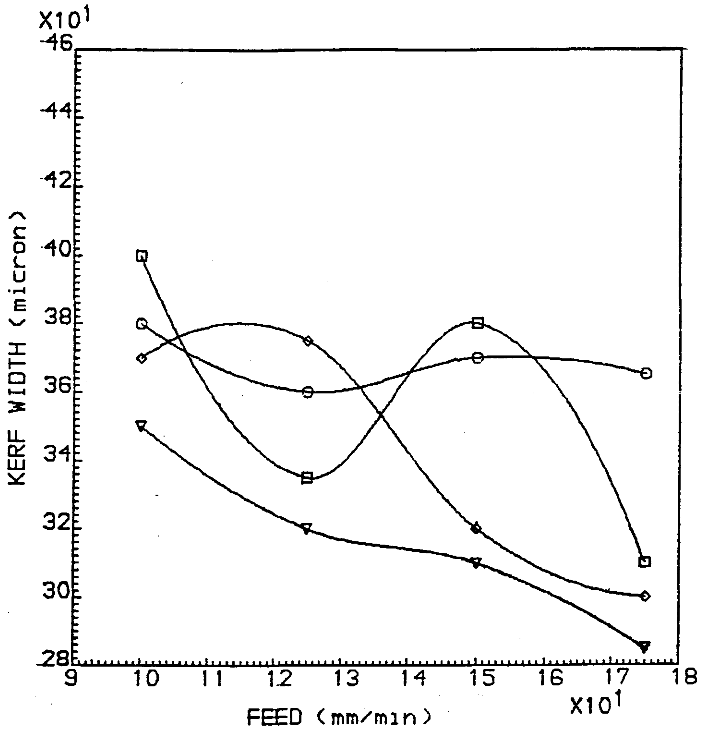


PRF = 20Hz , PULSE ENERGY = 2400mJ



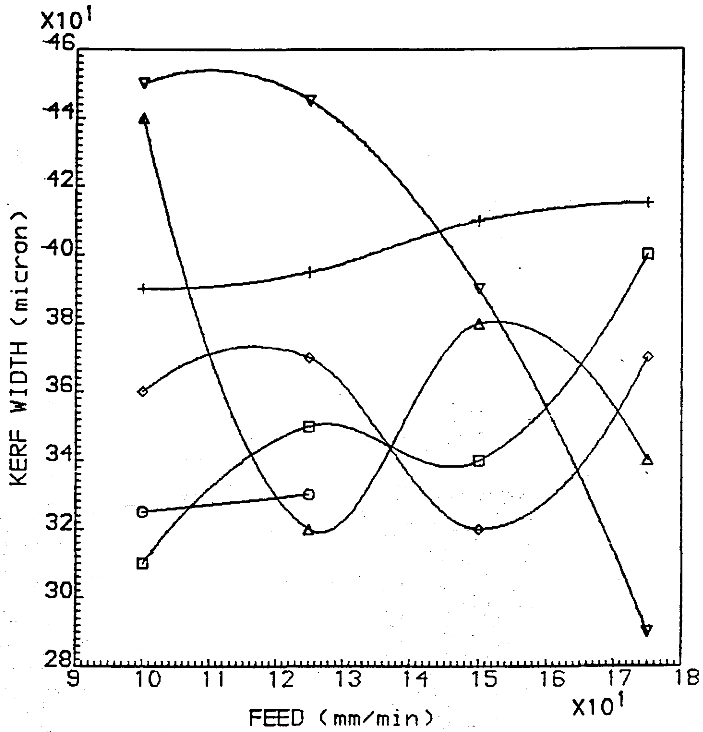
PRF = 15Hz , PULSE ENERGY = 3200mJ

GRAPH 4.7 : KERF WIDTH Vs FEED FOR 0.7mm THICK ALUMINIUM



o 2000mb, 500mic
◊ 3000mb, 500mic
□ 2000mb, 1000mic
▽ 3000mb, 1000mic

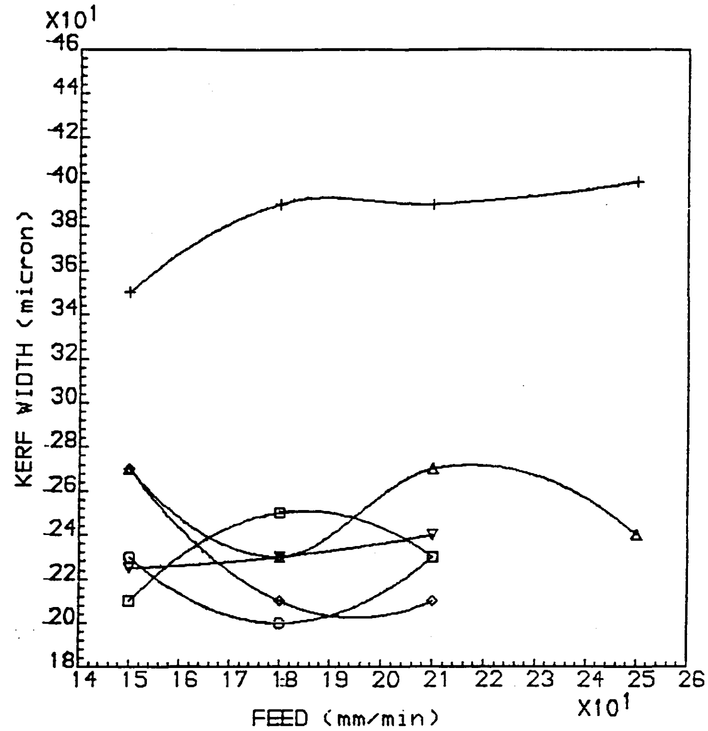
PRF = 15Hz , PULSE ENERGY = 3500mJ



o 1000mb, 500mic
◊ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
Δ 2000mb, 1000mic
\+ 3000mb, 1000mic

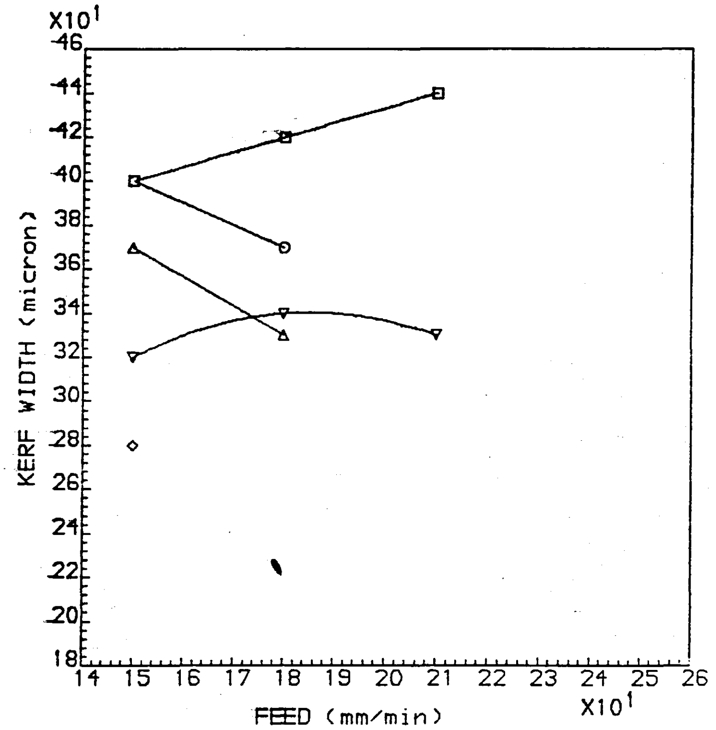
PRF = 11Hz , PULSE ENERGY = 5100mJ

GRAPH 4.8 : KERF WIDTH Vs FEED FOR 0.9mm THICK NICKEL



o 1000mb, 500mic
◊ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
Δ 2000mb, 1000mic
+ 3000mb, 1000mic

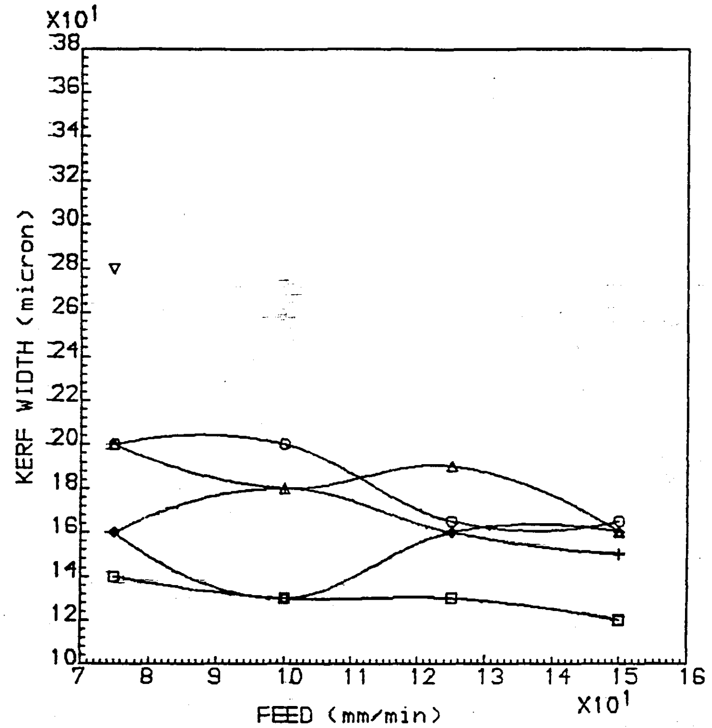
PRF = 15Hz , PULSE ENERGY = 3200mJ



o 1000mb, 500mic
◊ 2000mb, 500mic
□ 1000mb, 1000mic
▽ 2000mb, 1000mic
Δ 3000mb, 1000mic

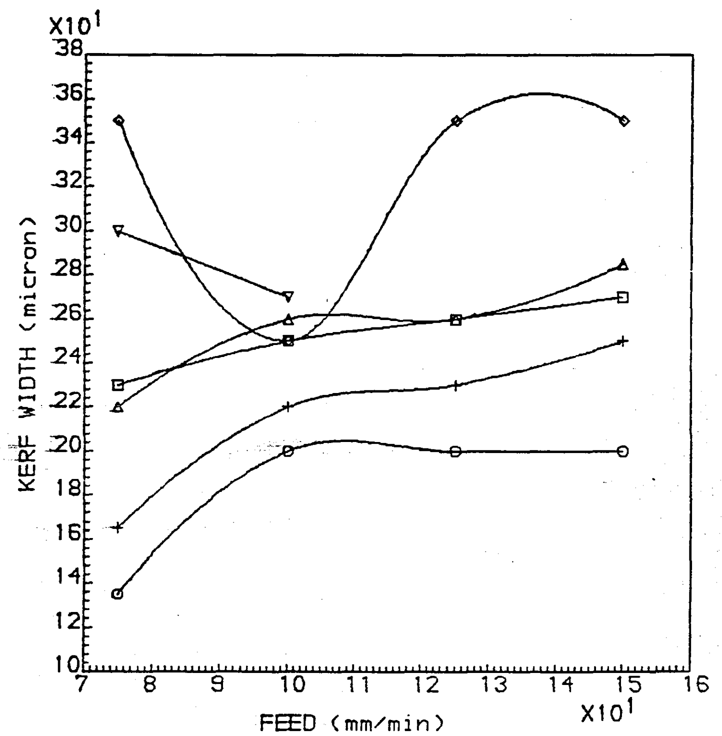
PRF = 10Hz , PULSE ENERGY =4800mJ

GRAPH 4.9 : KERF WIDTH Vs FEED FOR 0.75 THICK INCONEL



o 1000mb, 500mic
▽ 1000mb, 1000mic
◊ 2000mb, 500mic
△ 2000mb, 1000mic
□ 3000mb, 500mic
+ 3000mb, 1000mic

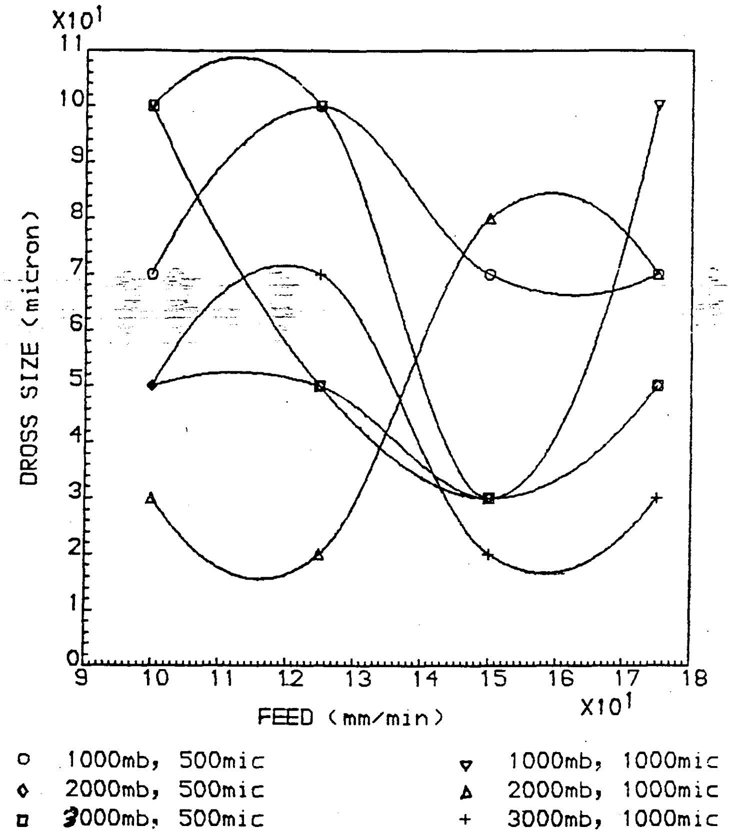
PRF = 15Hz , PULSE ENERGY = 3200mJ



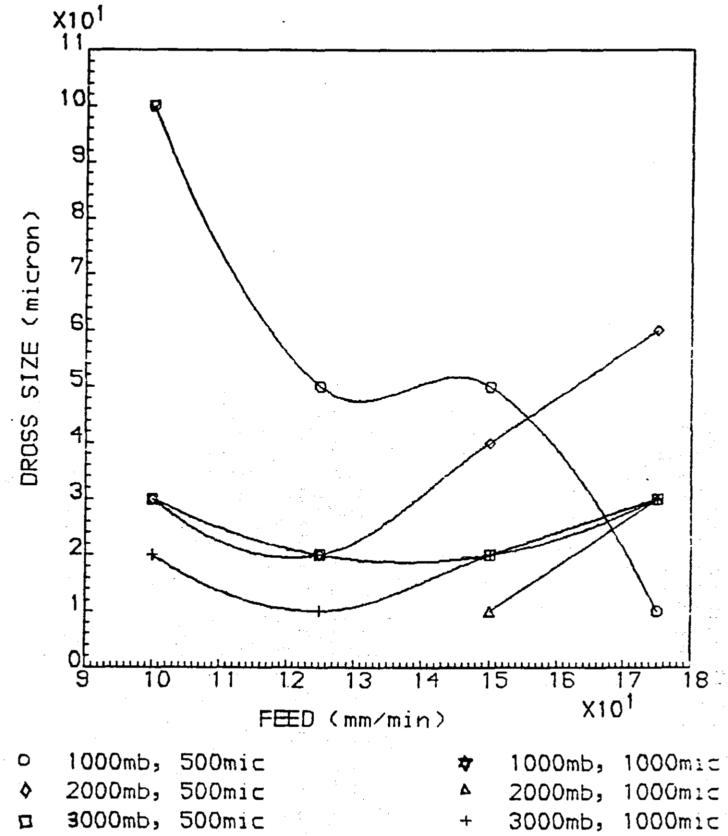
o 1000mb, 500mic
▽ 1000mb, 1000mic
◊ 2000mb, 500mic
△ 2000mb, 1000mic
□ 3000mb, 500mic
+ 3000mb, 1000mic

PRF = 10Hz , PULSE ENERGY =4800mJ

GRAPH 4.10 : KERF WIDTH Vs FEED FOR 1.6 THICK BRASS

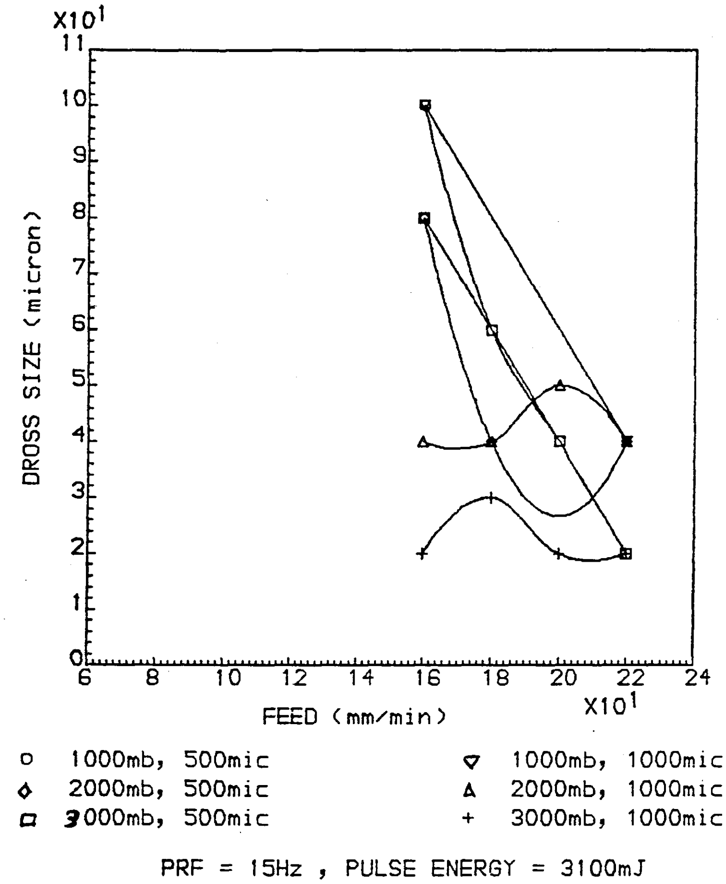


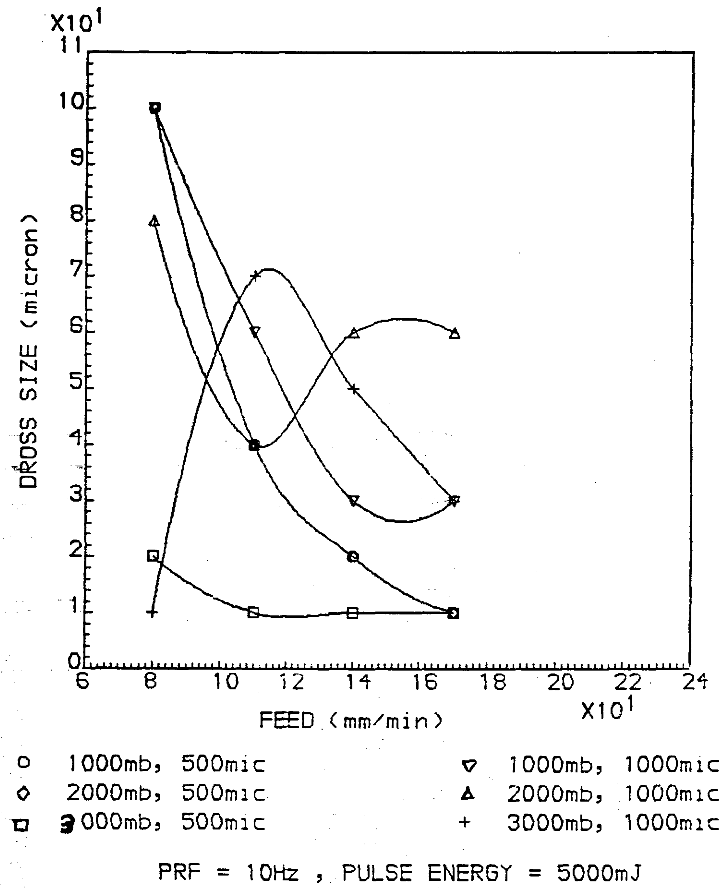
PRF = 15Hz , PULSE ENERGY = 3200mJ



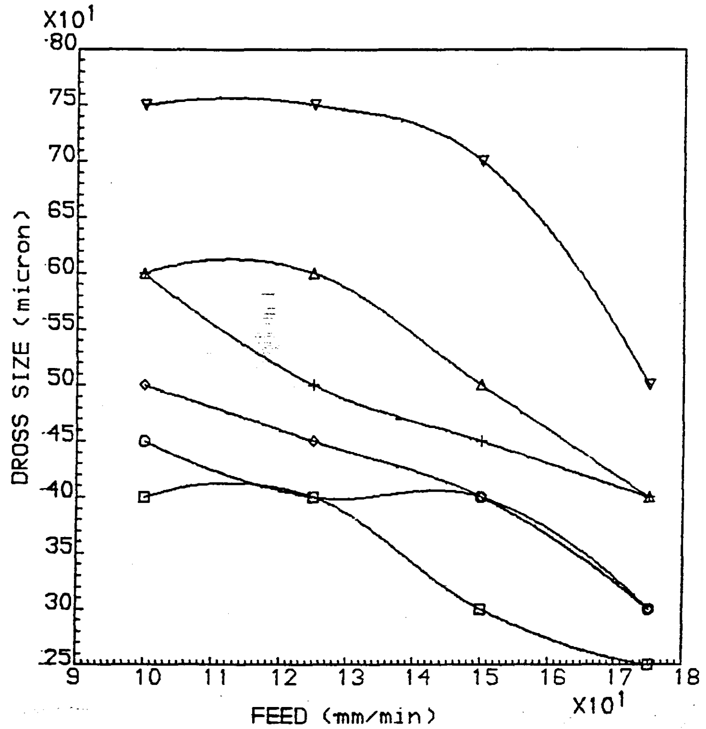
PRF = 10Hz , PULSE ENERGY = 4800mJ

GRAPH 4.11 : DROSS SIZE Vs FEED FOR 1.6mm THICK STAINLESS STEEL



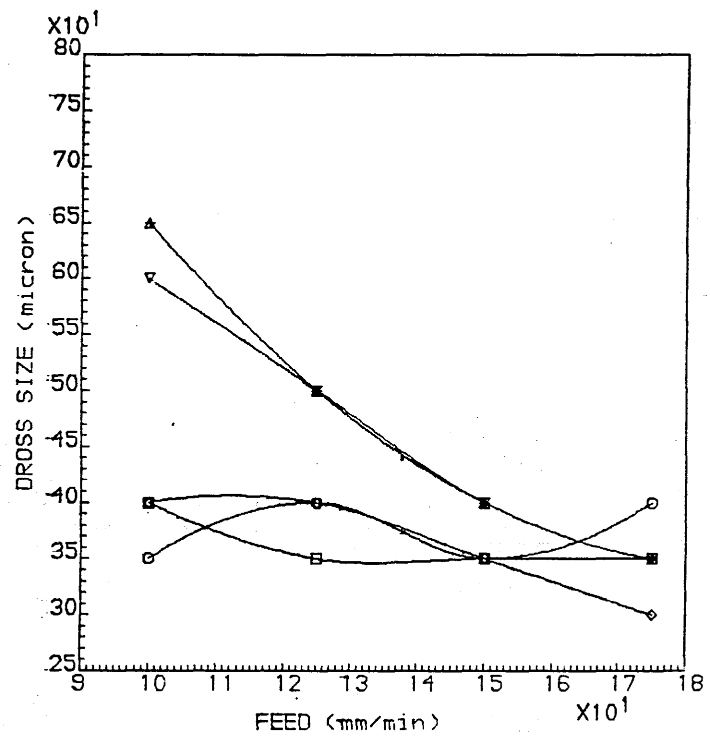


GRAPH 4.12 : DROSS SIZE Vs FEED FOR 1.6mm THICK TOOL STEEL



o 1000mb, 500mic ∇ 1000mb, 1000mic
◇ 2000mb, 500mic Δ 2000mb, 1000mic
□ 3000mb, 500mic + 3000mb, 1000mic

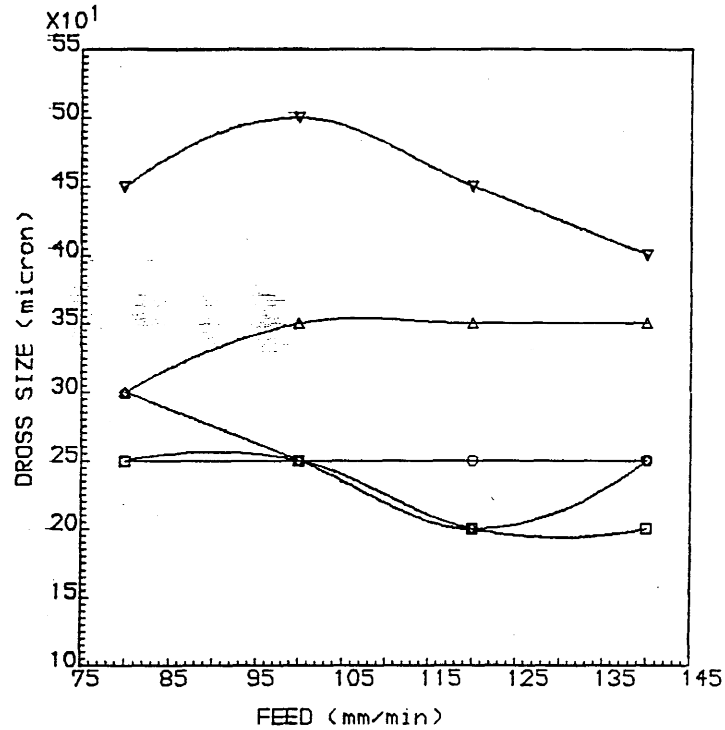
PRF = 15Hz , PULSE ENERGY = 3200mJ



o 1000mb, 500mic ∇ 1000mb, 1000mic
◇ 2000mb, 500mic Δ 2000mb, 1000mic
□ 3000mb, 500mic + 3000mb, 1000mic

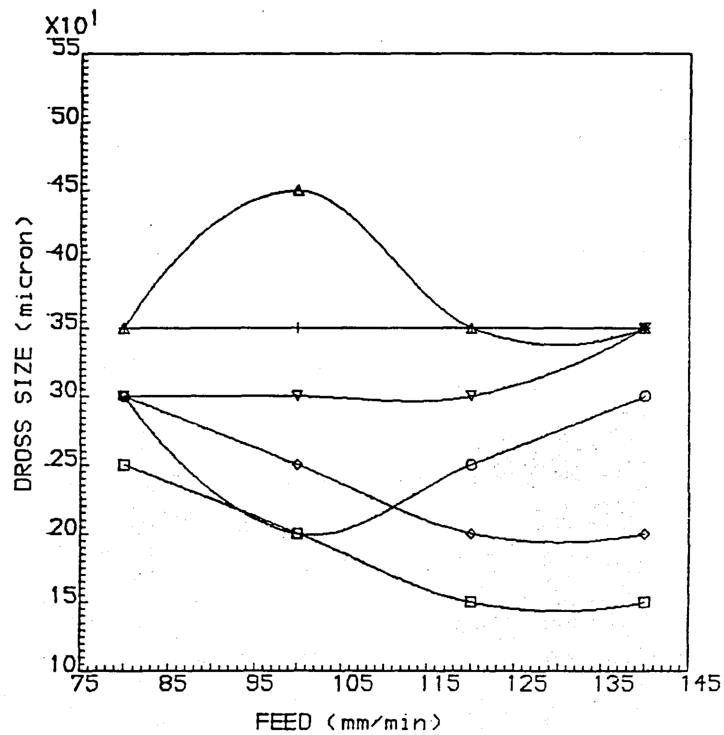
PRF = 10Hz , PULSE ENERGY = 4800mJ

GRAPH 4.13 : DROSS SIZE Vs FEED FOR 1.6mm THICK TITANIUM



○ 1000mb, 500mic
◇ 2000mb, 500mic
□ 3000mb, 500mic
▽ 2000mb, 1000mic
△ 3000mb, 1000mic

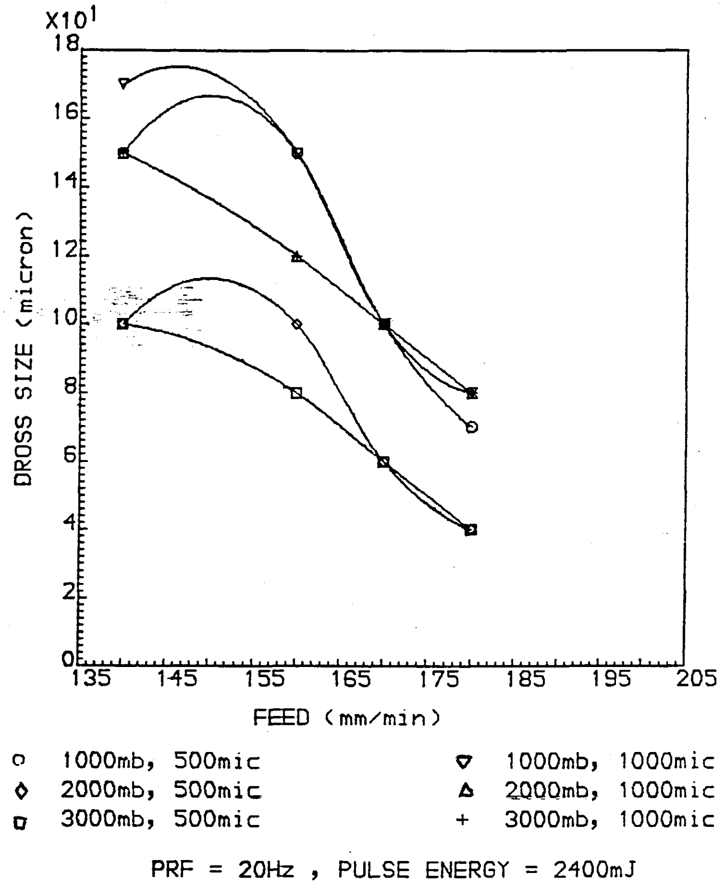
PRF = 15Hz , PULSE ENERGY = 3200mJ

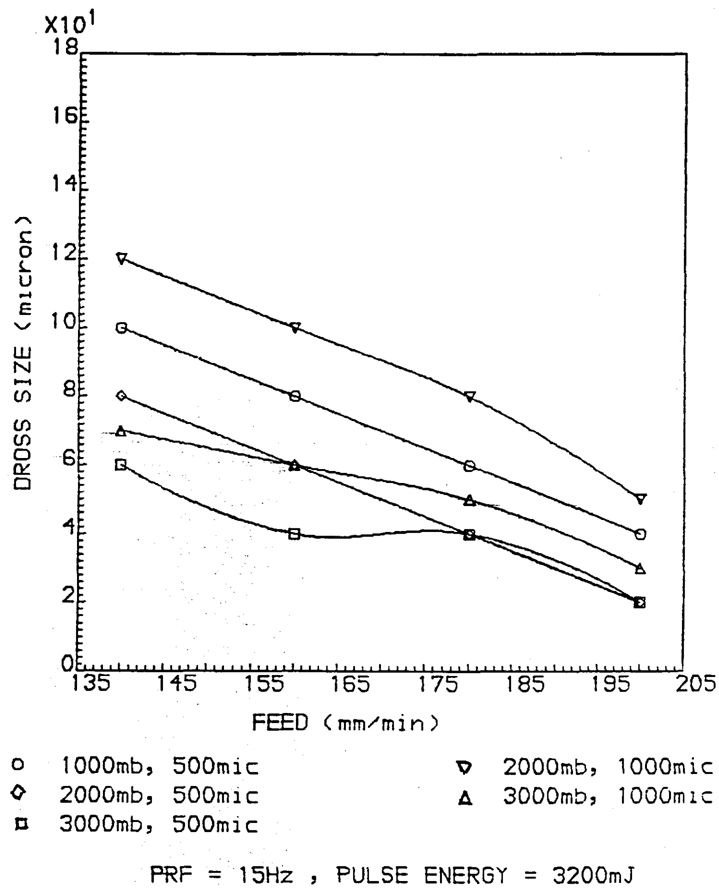


○ 1000mb, 500mic
◇ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
△ 2000mb, 1000mic
\+ 3000mb, 1000mic

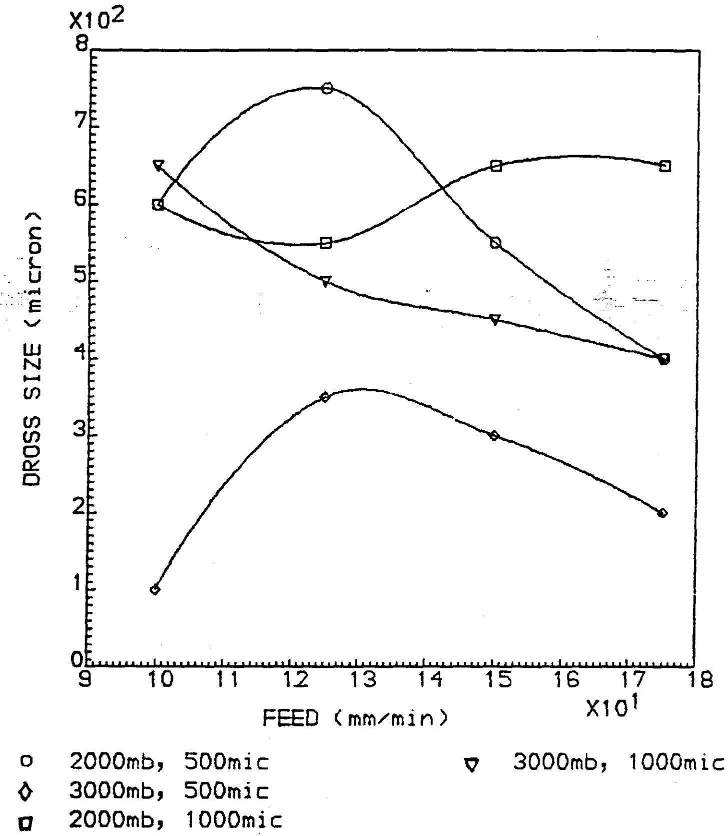
PRF = 10Hz , PULSE ENERGY = 5000mJ

GRAPH 4.14 : DROSS SIZE Vs FEED FOR 1.6mm THICK ALUMINIUM

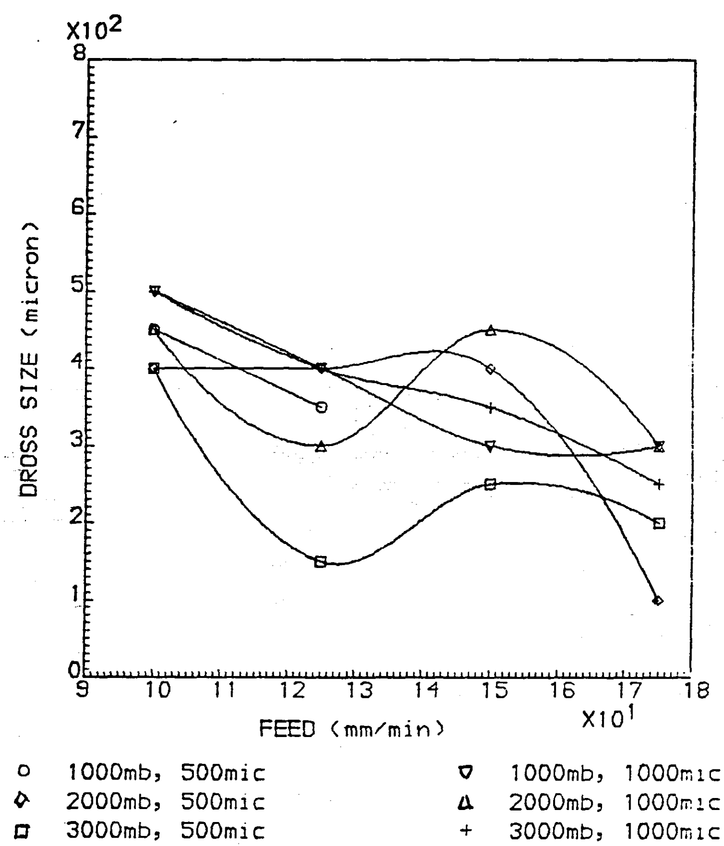




GRAPH 4.15 : DROSS SIZE Vs FEED FOR 0.7mm THICK ALUMINIUM

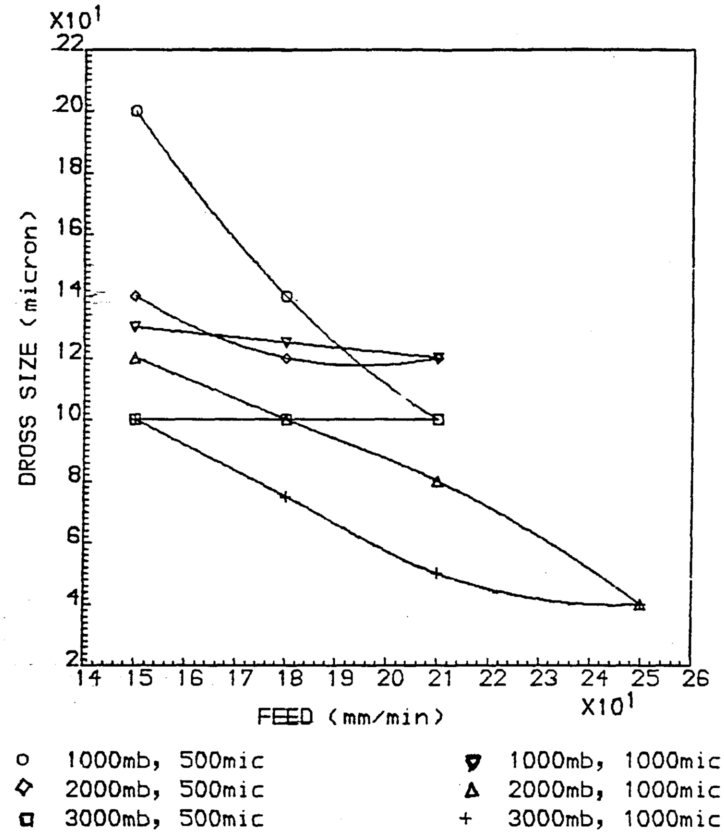


PRF = 15Hz , PULSE ENERGY = 3500mJ

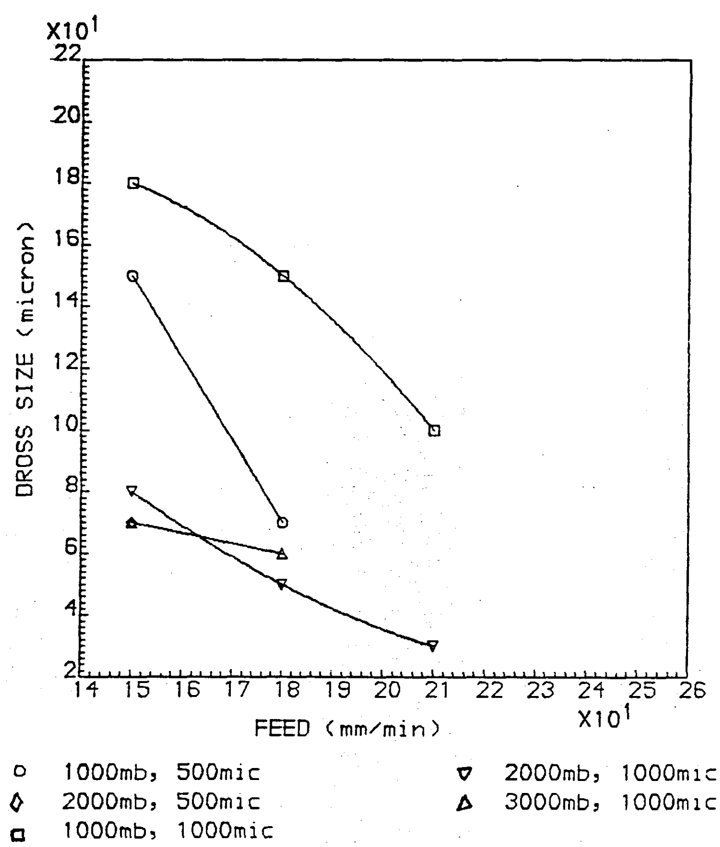


PRF = 11Hz , PULSE ENERGY = 5100mJ

GRAPH 4.16 : DROSS SIZE Vs FEED FOR 0.9mm THICK NICKEL

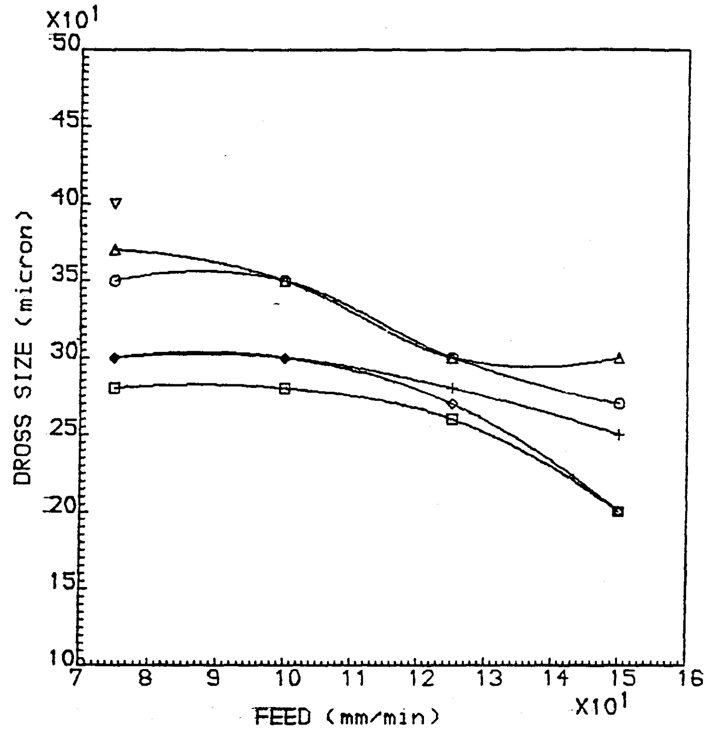


PRF = 15Hz , PULSE ENERGY = 3200mJ



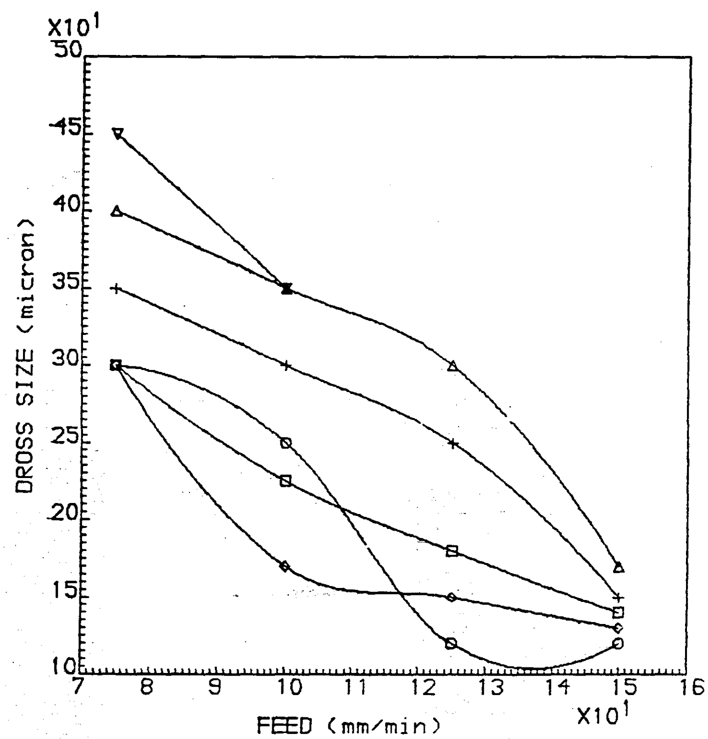
PRF = 10Hz , PULSE ENERGY =4800mJ

GRAPH 4.17 : DROSS SIZE Vs FEED FOR 0.75 THICK INCONEL



o 1000mb, 500mic ∇ 1000mb, 1000mic
◇ 2000mb, 500mic Δ 2000mb, 1000mic
□ 3000mb, 500mic + 3000mb, 1000mic

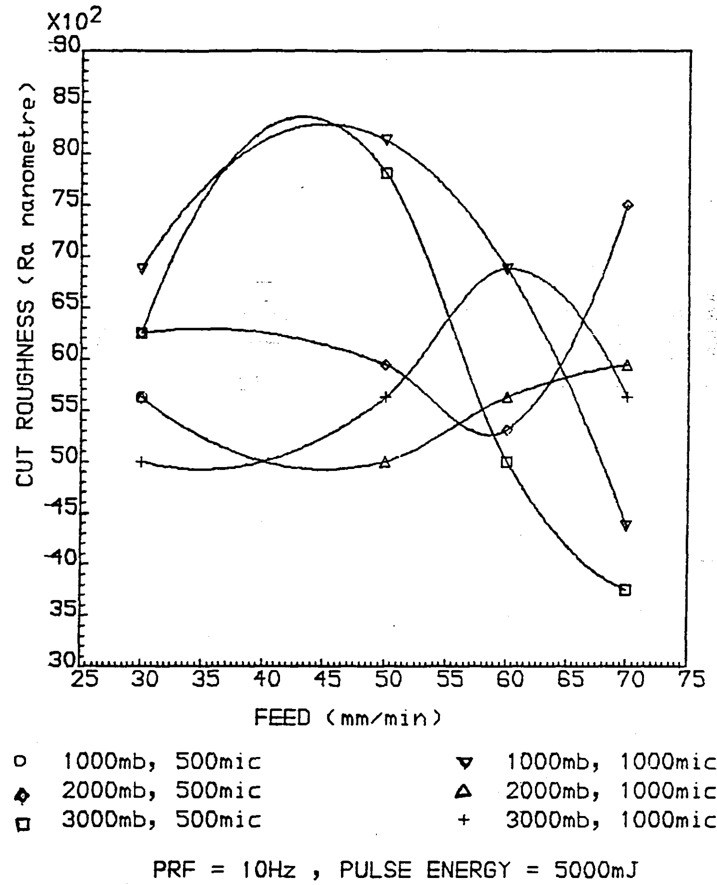
PRF = 15Hz , PULSE ENERGY = 3200mJ

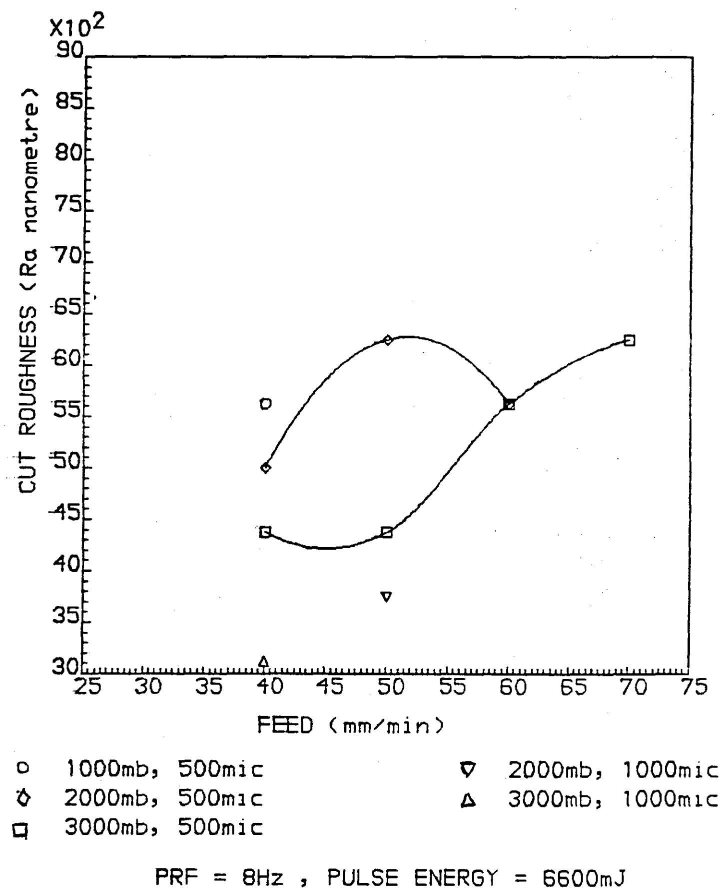


o 1000mb, 500mic ∇ 1000mb, 1000mic
◇ 2000mb, 500mic Δ 2000mb, 1000mic
□ 3000mb, 500mic + 3000mb, 1000mic

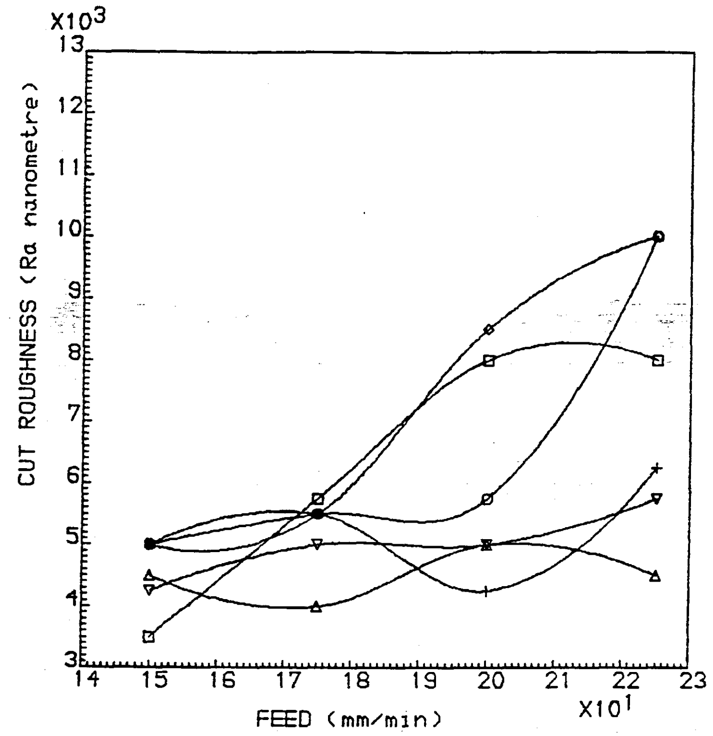
PRF = 10Hz , PULSE ENERGY =4800mJ

GRAPH 4.18 : DROSS SIZE Vs FEED FOR 1.6 THICK BRASS



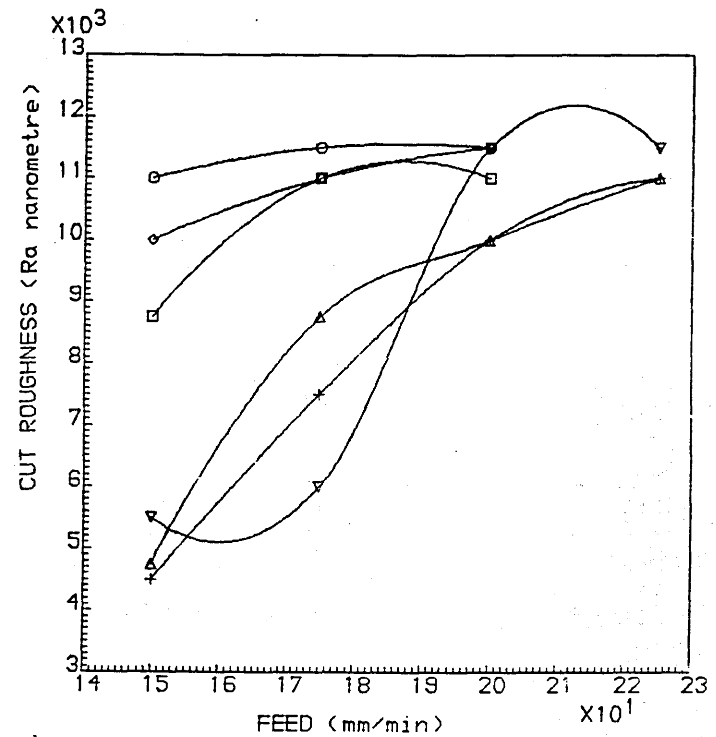


GRAPH 4.19 : CUT ROUGHNESS Vs FEED FOR 3.16mm THICK MILD STEEL



o 1000mb, 500mic
◊ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
△ 2000mb, 1000mic
+ 3000mb, 1000mic

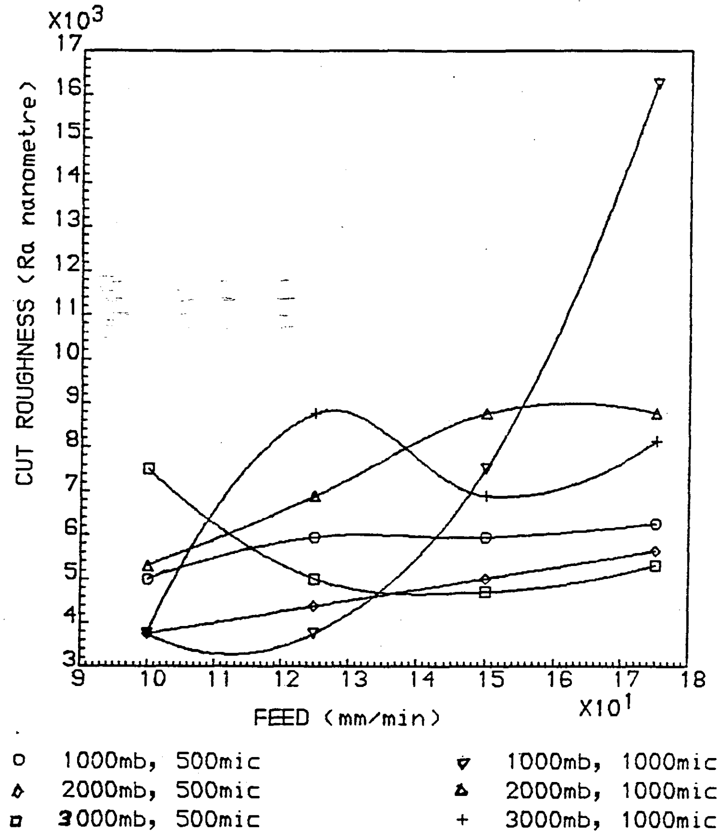
PRF = 15Hz , PULSE ENERGY = 3200mJ



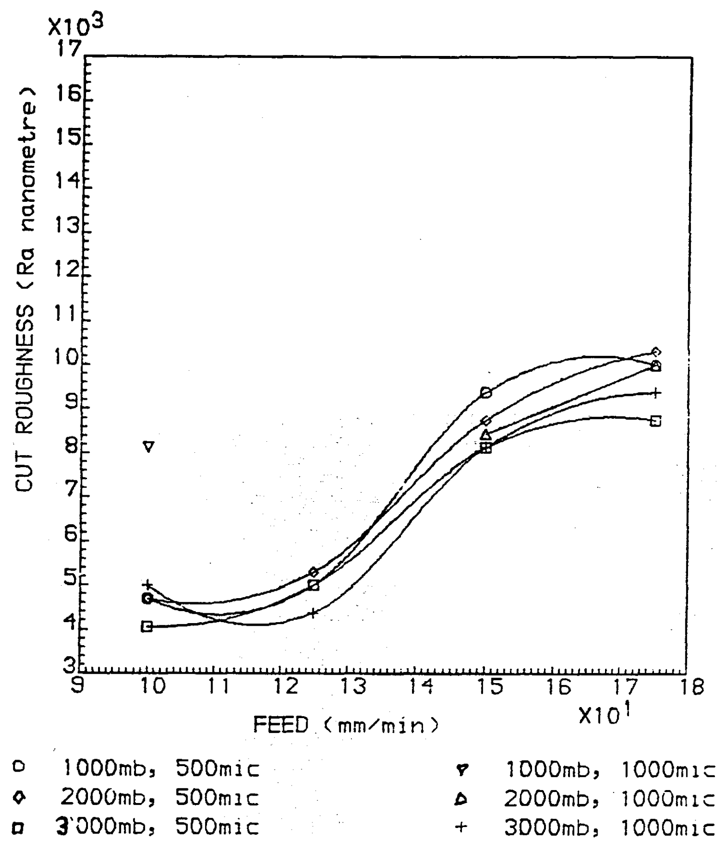
o 1000mb, 500mic
◊ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
△ 2000mb, 1000mic
+ 3000mb, 1000mic

PRF = 10Hz , PULSE ENERGY = 4800mJ

GRAPH 4.20 : CUT ROUGHNESS Vs FEED FOR 1.6mm THICK MILD STEEL

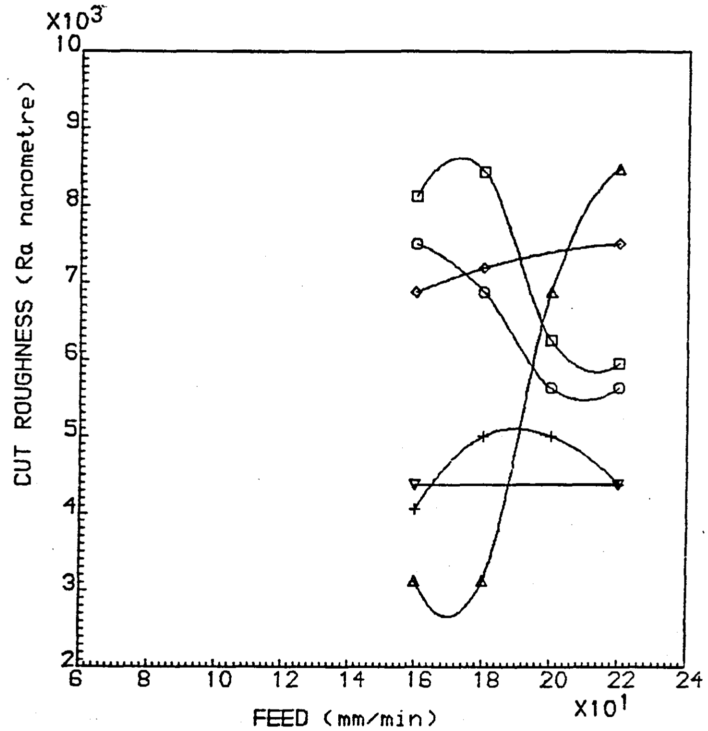


PRF = 15Hz , PULSE ENERGY = 3200mJ



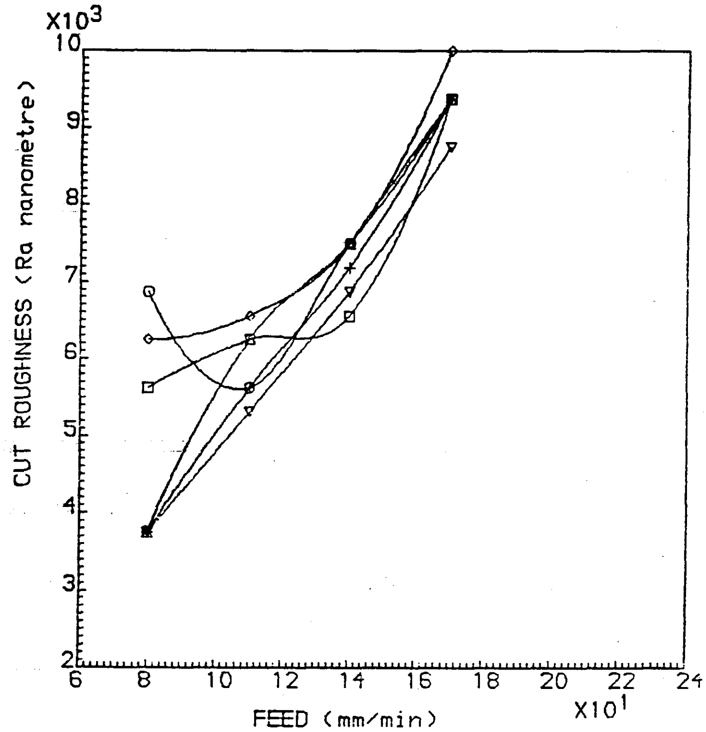
PRF = 10Hz , PULSE ENERGY = 4800mJ

GRAPH 4.21 : CUT ROUGHNESS Vs FEED FOR 1.6mm THICK STAINLESS STEEL



○ 1000mb, 500mic ▽ 1000mb, 1000mic
◇ 2000mb, 500mic △ 2000mb, 1000mic
□ 3000mb, 500mic + 3000mb, 1000mic

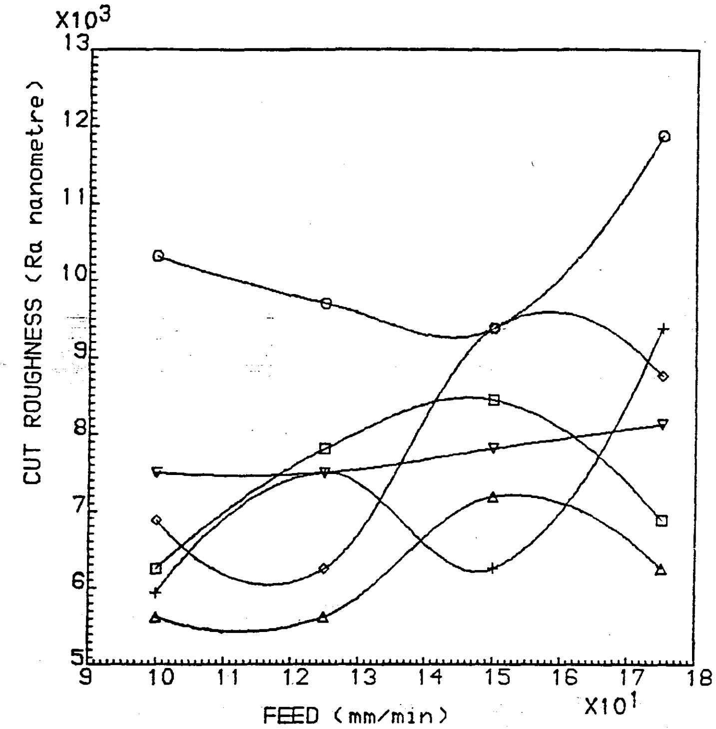
PRF = 15Hz , PULSE ENERGY = 3100mJ



○ 1000mb, 500mic ▽ 1000mb, 1000mic
◇ 2000mb, 500mic △ 2000mb, 1000mic
□ 3000mb, 500mic + 3000mb, 1000mic

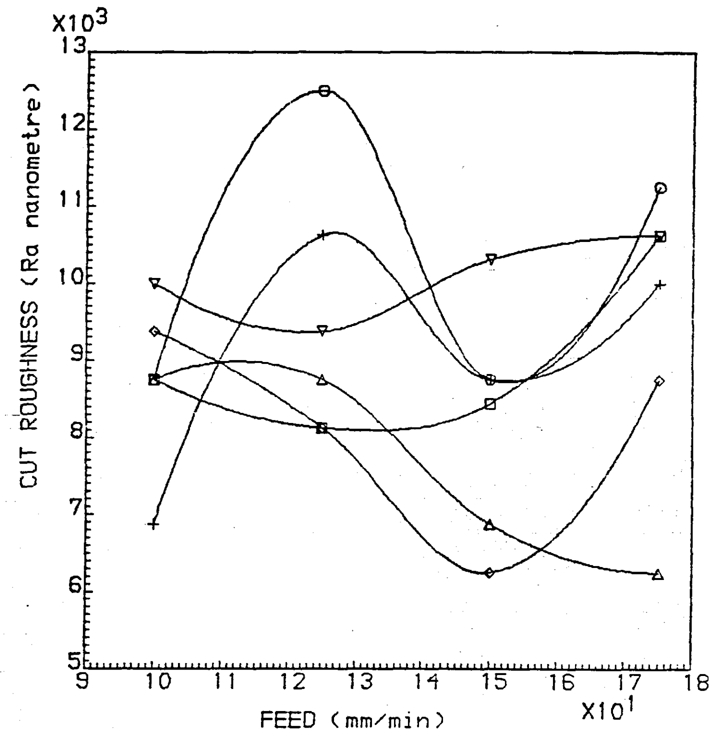
PRF = 10Hz , PULSE ENERGY = 5000mJ

GRAPH 4.22 : CUT ROUGHNESS Vs FEED FOR 1.6mm THICK TOOL STEEL



○ 1000mb, 500mic
◇ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
△ 2000mb, 1000mic
\+ 3000mb, 1000mic

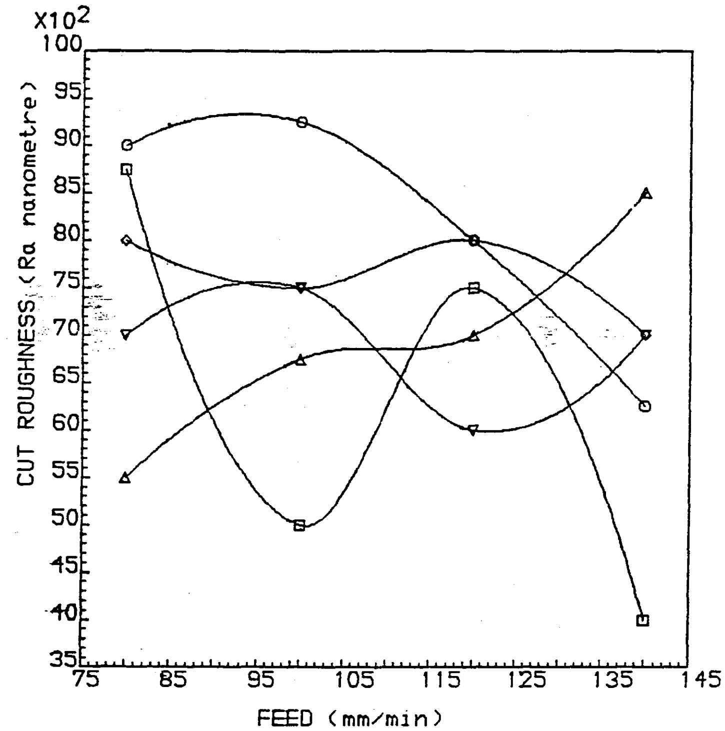
PRF = 15Hz , PULSE ENERGY = 3200mJ



○ 1000mb, 500mic
◇ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
△ 2000mb, 1000mic
\+ 3000mb, 1000mic

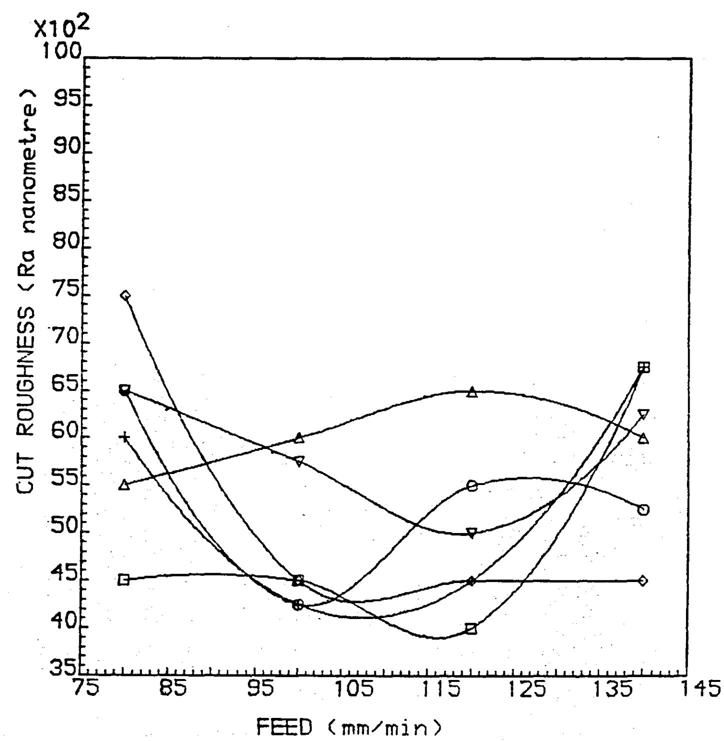
PRF = 10Hz , PULSE ENERGY = 4800mJ

GRAPH 4.23 : CUT ROUGHNESS Vs FEED FOR 1.6mm THICK TITANIUM



o 1000mb, 500mic
◊ 2000mb, 500mic
□ 3000mb, 500mic
▽ 2000mb, 1000mic
△ 3000mb, 1000mic

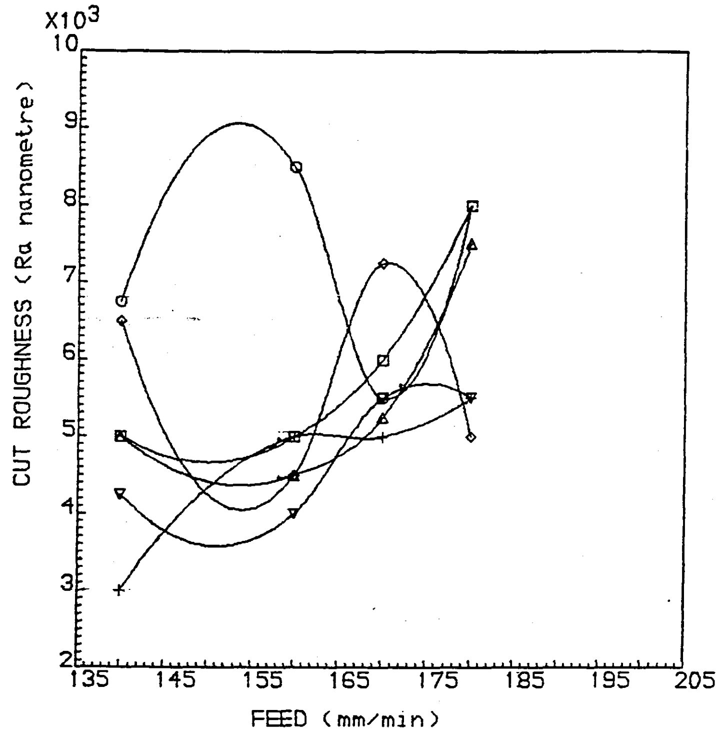
PRF = 15Hz , PULSE ENERGY = 3200mJ



o 1000mb, 500mic
◊ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
△ 2000mb, 1000mic
\+ 3000mb, 1000mic

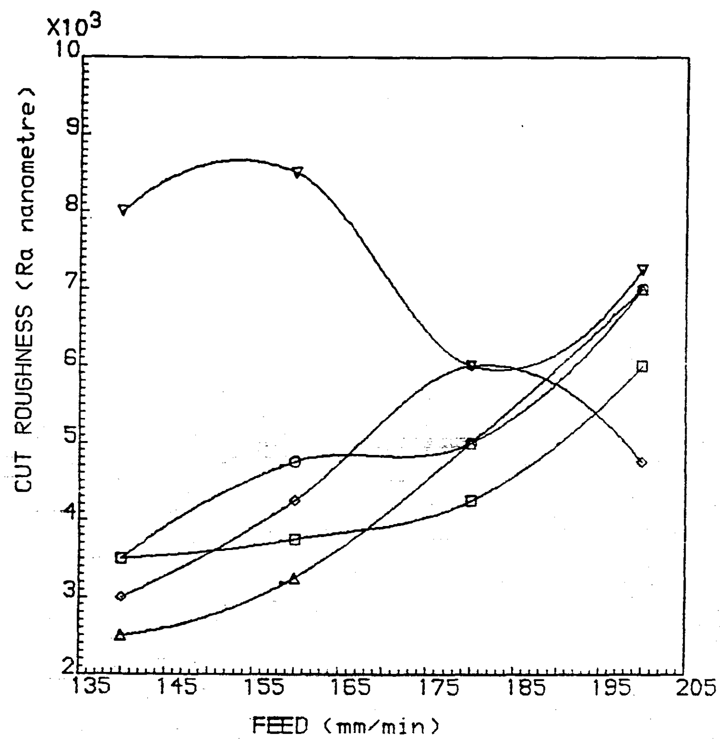
PRF = 10Hz , PULSE ENERGY = 5000mJ

GRAPH 4.24 : CUT ROUGHNESS Vs FEED FOR 1.6mm THICK ALUMINIUM



○ 1000mb, 500mic
◇ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
△ 2000mb, 1000mic
+ 3000mb, 1000mic

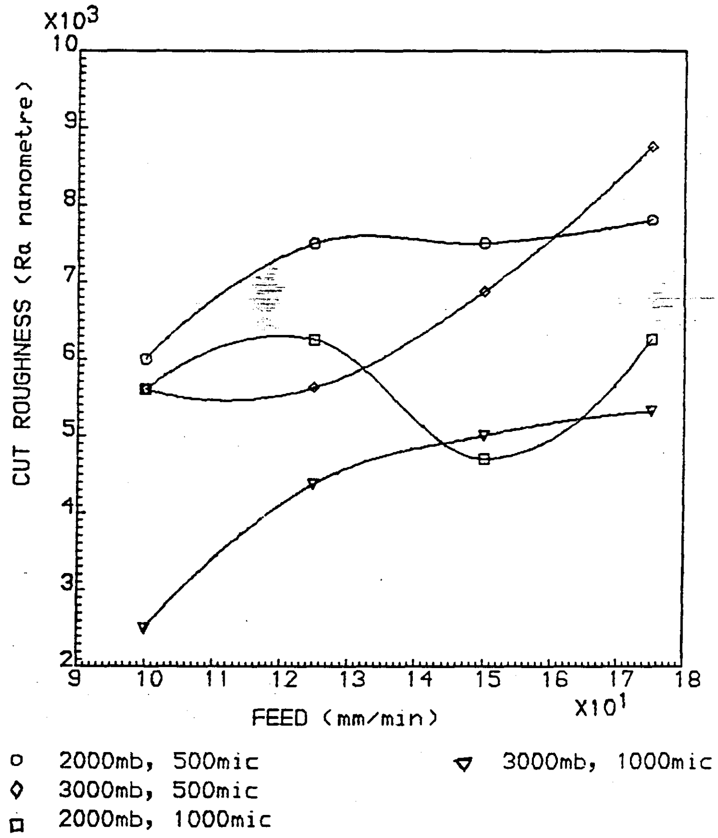
PRF = 20Hz , PULSE ENERGY = 2400mJ



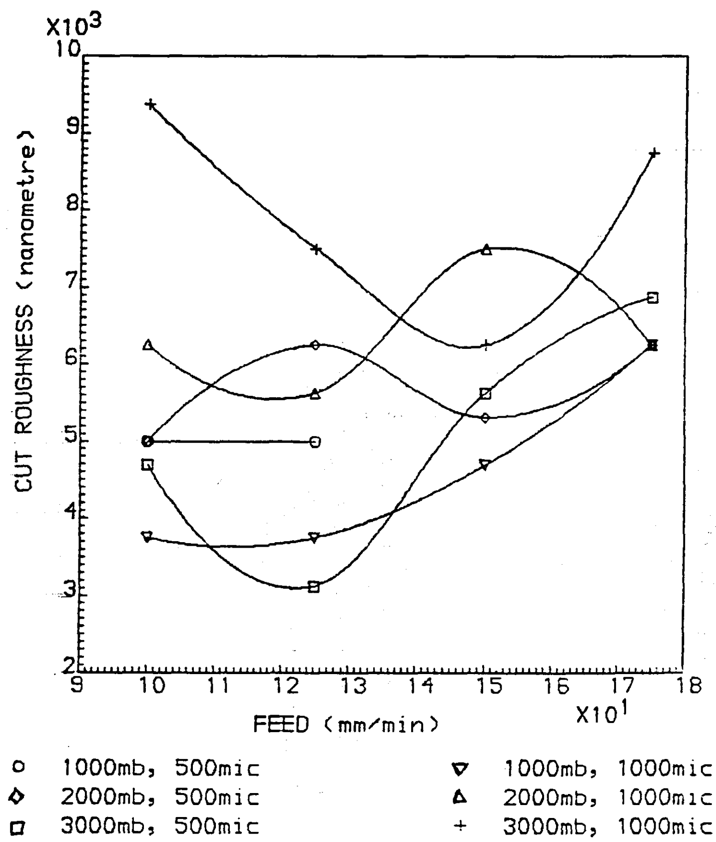
○ 1000mb, 500mic
◇ 2000mb, 500mic
□ 3000mb, 500mic
▽ 2000mb, 1000mic
△ 3000mb, 1000mic

PRF = 15Hz , PULSE ENERGY = 3200mJ

GRAPH 4.25 : CUT ROUGHNESS Vs FEED FOR 0.7mm THICK ALUMINIUM

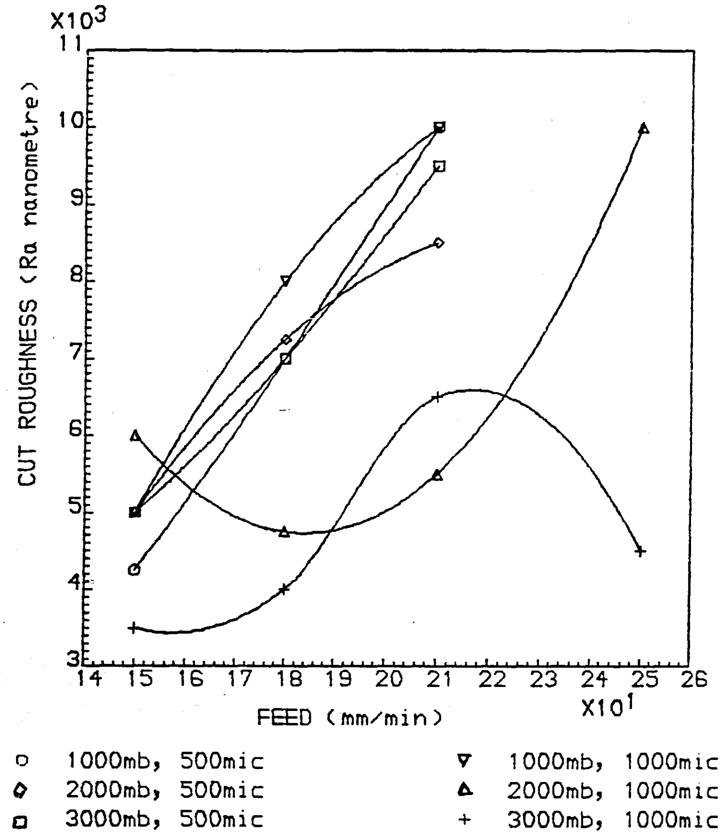


PRF = 15Hz , PULSE ENERGY = 3500mJ

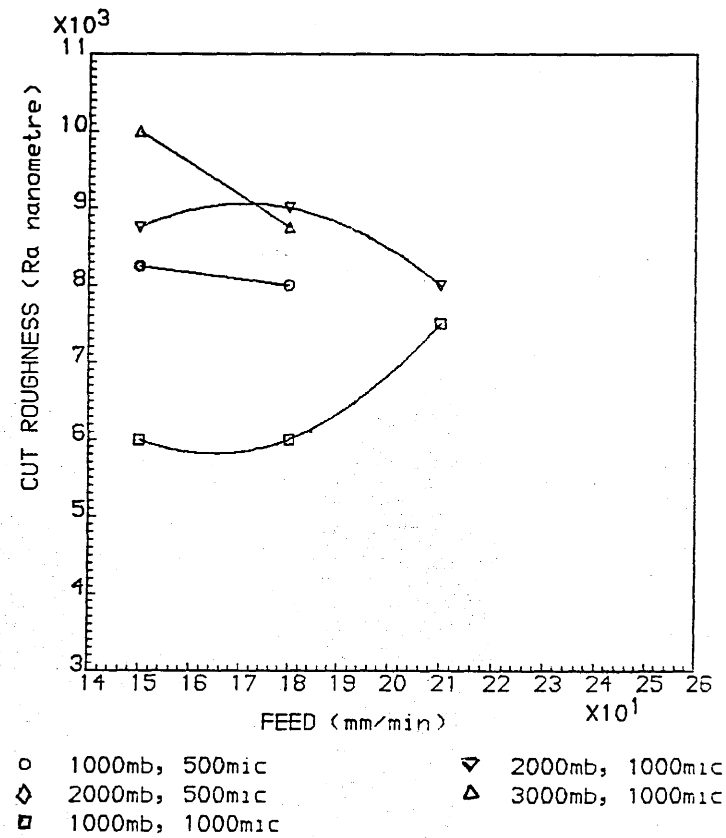


PRF = 11Hz , PULSE ENERGY = 5100mJ

GRAPH 4.26 : CUT ROUGHNESS Vs FEED FOR 0.9mm THICK NICKEL

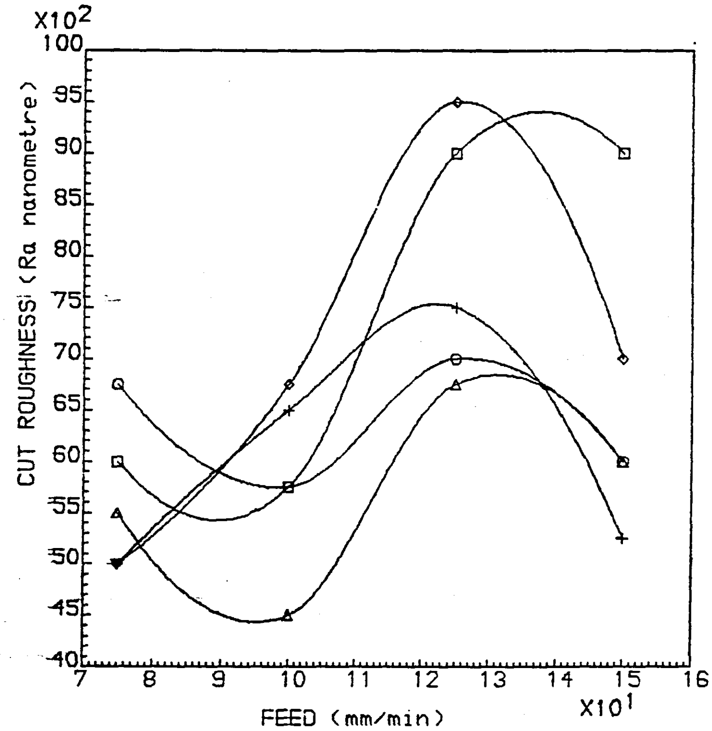


PRF = 15Hz , PULSE ENERGY = 3200mJ



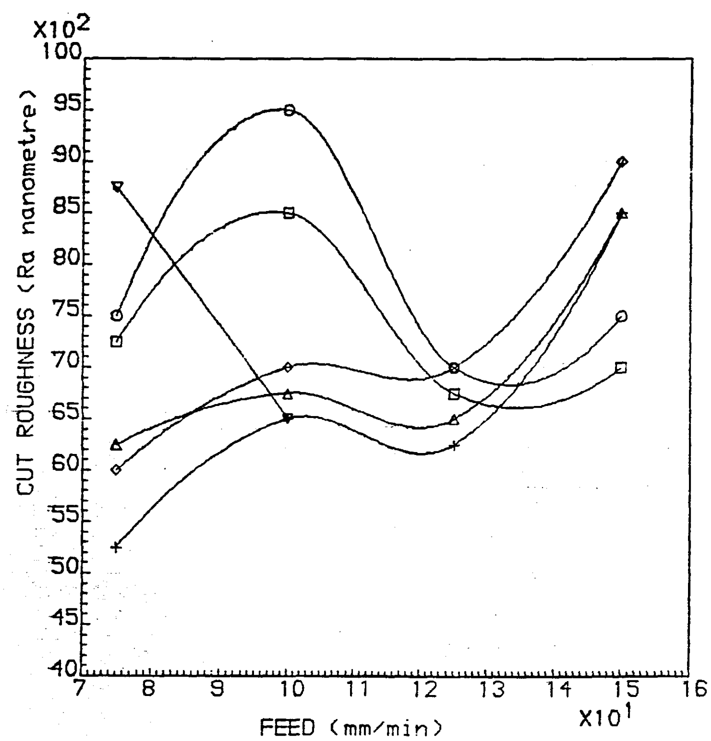
PRF = 10Hz , PULSE ENERGY =4800mJ

GRAPH 4.27 : CUT ROUGHNESS Vs FEED FOR 0.75 THICK INCONEL



o 1000mb, 500mic
◇ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
△ 2000mb, 1000mic
+ 3000mb, 1000mic

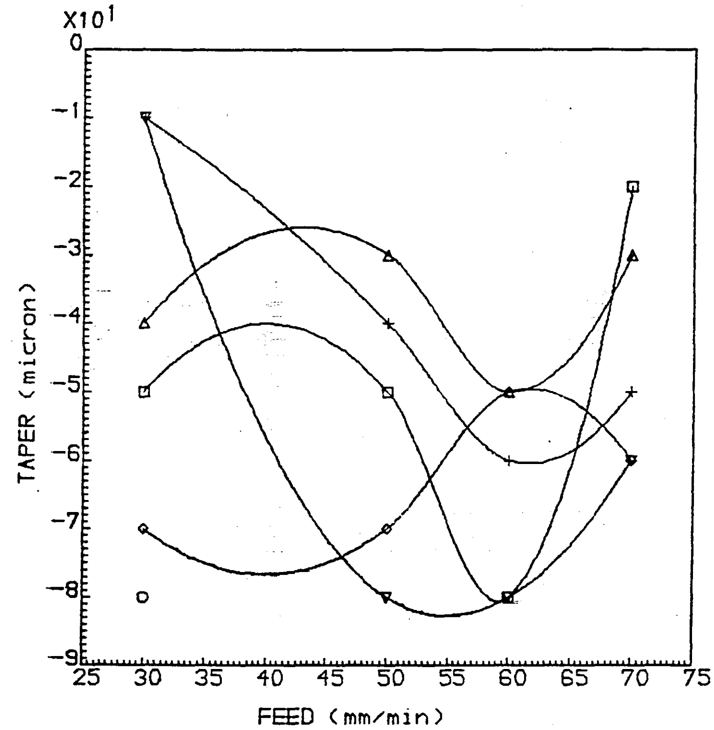
PRF = 15Hz , PULSE ENERGY = 3200mJ



o 1000mb, 500mic
◇ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
△ 2000mb, 1000mic
+ 3000mb, 1000mic

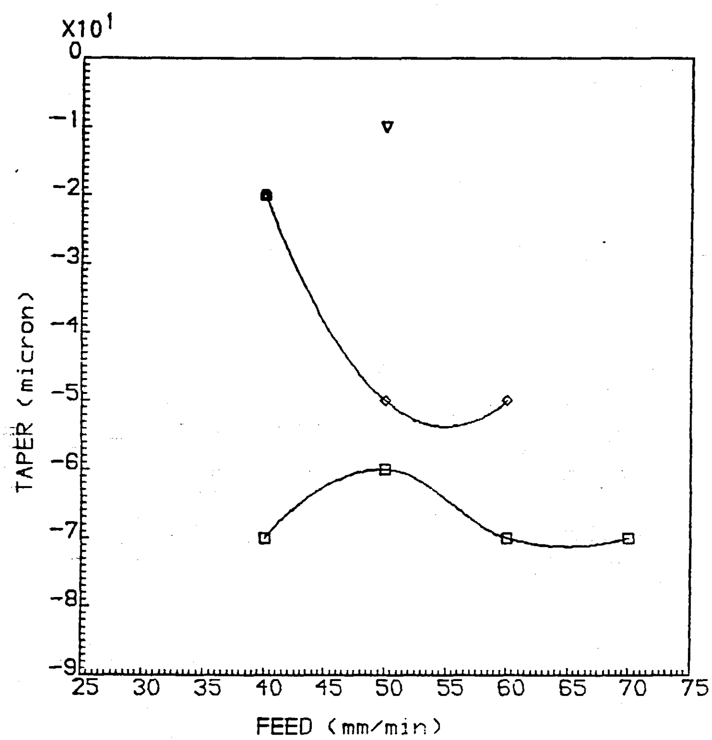
PRF = 10Hz , PULSE ENERGY =4800mJ

GRAPH 4.28 : CUT ROUGHNESS Vs FEED FOR 1.6 THICK BRASS



o 1000mb, 500mic
◇ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
△ 2000mb, 1000mic
+ 3000mb, 1000mic

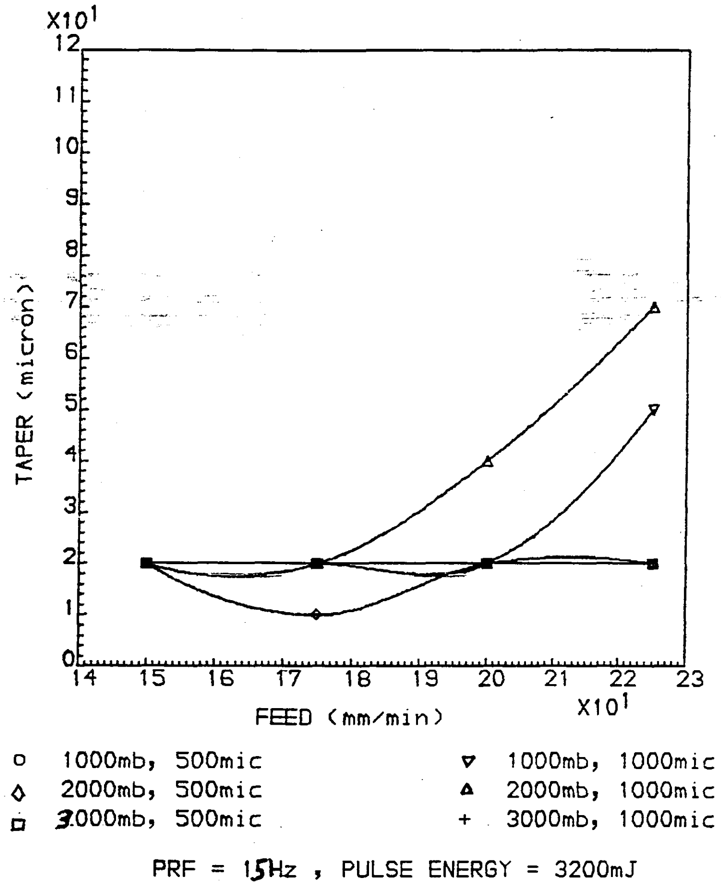
PRF = 10Hz , PULSE ENERGY = 5000mJ

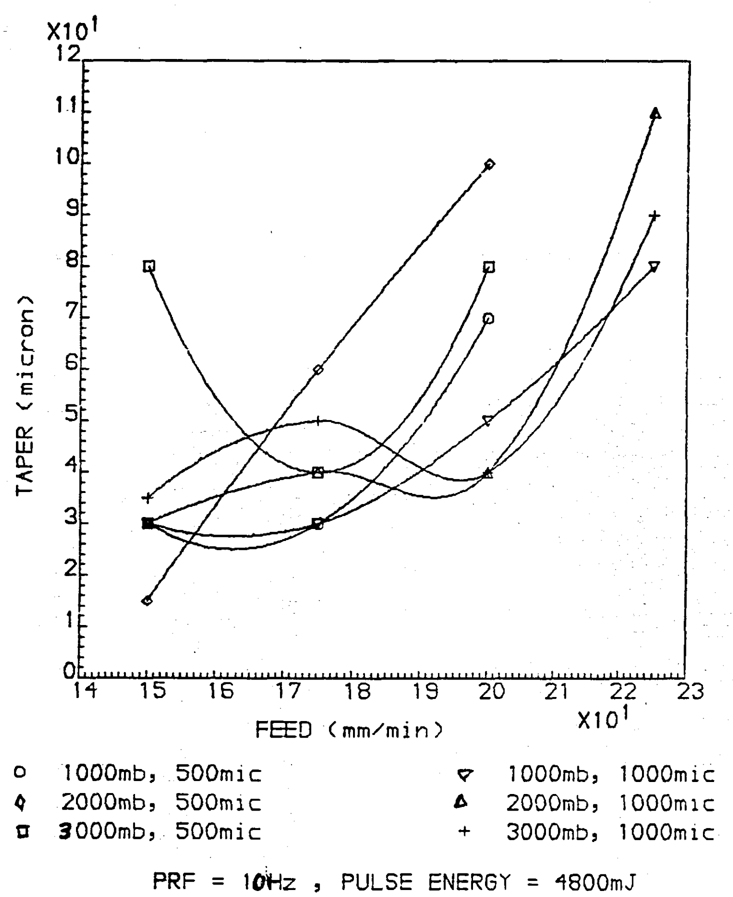


o 1000mb, 500mic
◇ 2000mb, 500mic
□ 3000mb, 500mic
▽ 2000mb, 1000mic
△ 3000mb, 1000mic

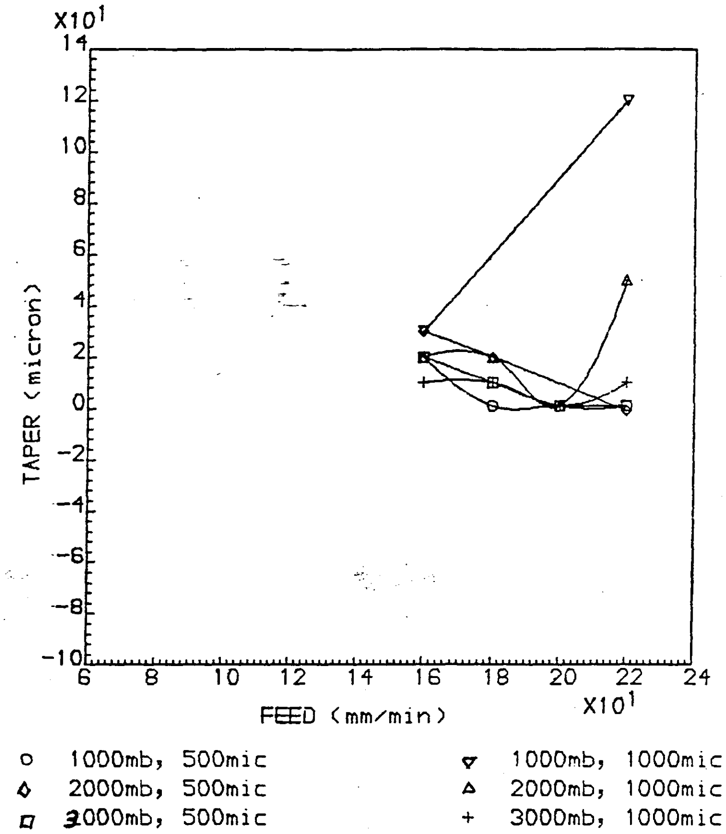
PRF = 8Hz , PULSE ENERGY = 6600mJ

GRAPH 4.29 : TAPER Vs FEED FOR 3.16mm THICK MILD STEEL

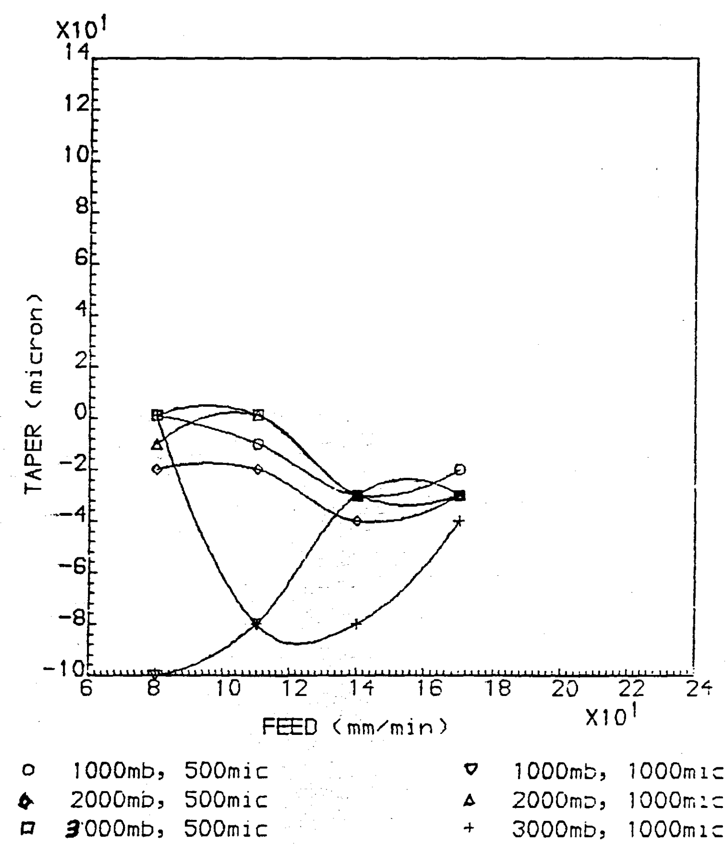




GRAPH 4.30 : TAPER Vs FEED FOR 1.6mm THICK MILD STEEL

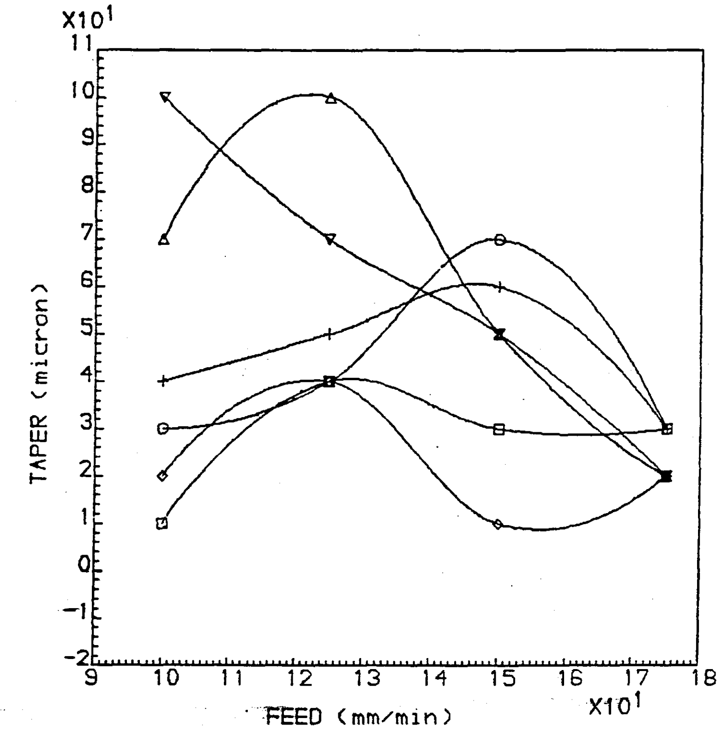


PRF = 15Hz , PULSE ENERGY = 3100mJ



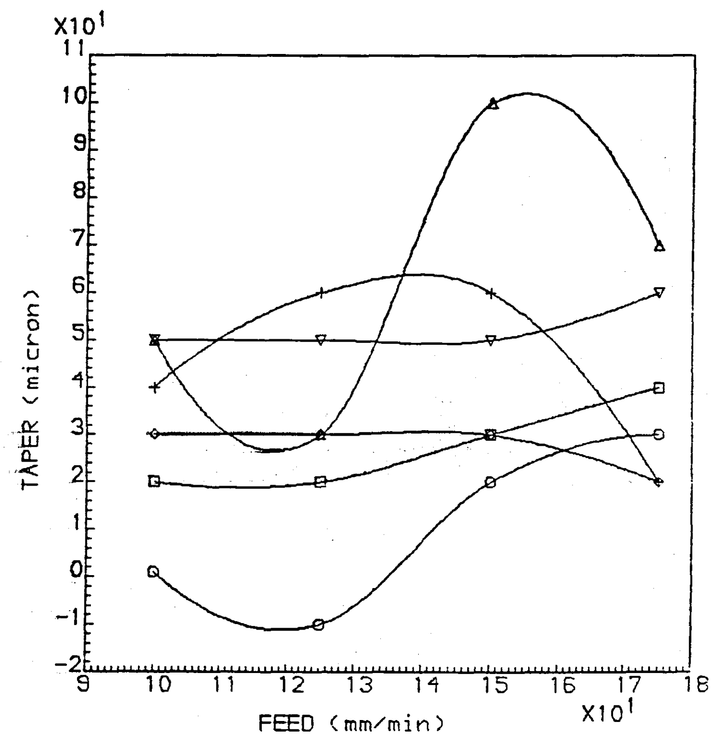
PRF = 10Hz , PULSE ENERGY = 5000mJ

GRAPH 4.31 : TAPER Vs FEED FOR 1.6mm THICK TOOL STEEL



○ 1000mb, 500mic ▽ 1000mb, 1000mic
◇ 2000mb, 500mic △ 2000mb, 1000mic
□ 3000mb, 500mic + 3000mb, 1000mic

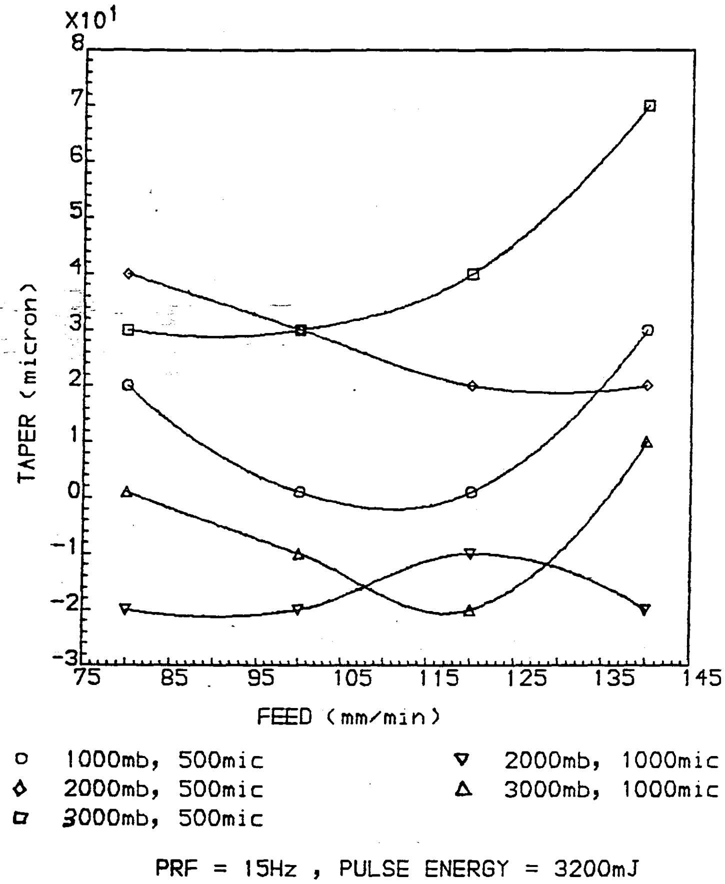
PRF = 15Hz , PULSE ENERGY = 3200mJ

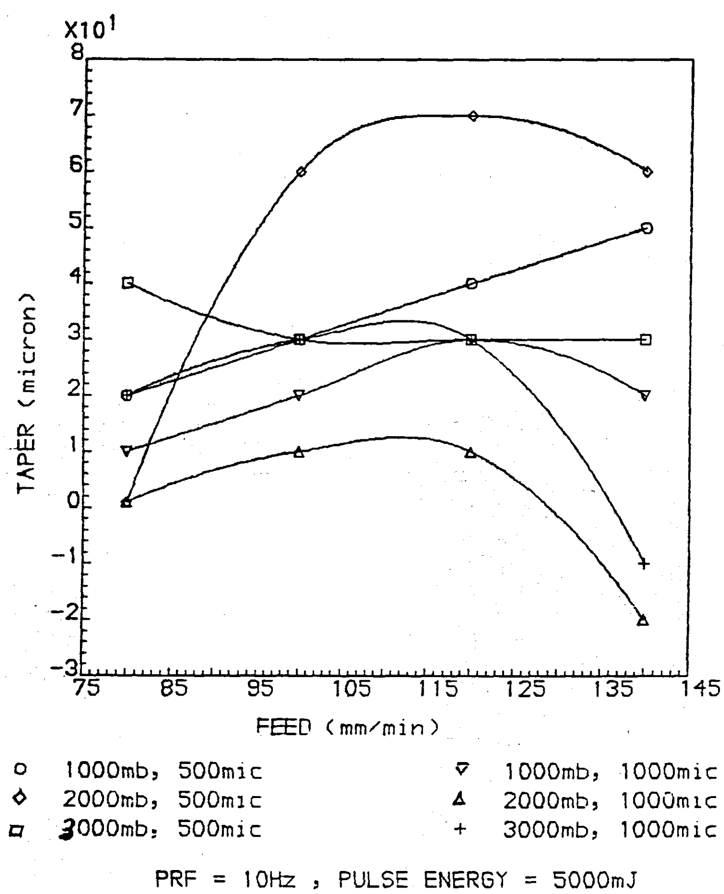


○ 1000mb, 500mic ▽ 1000mb, 1000mic
◇ 2000mb, 500mic △ 2000mb, 1000mic
□ 3000mb, 500mic + 3000mb, 1000mic

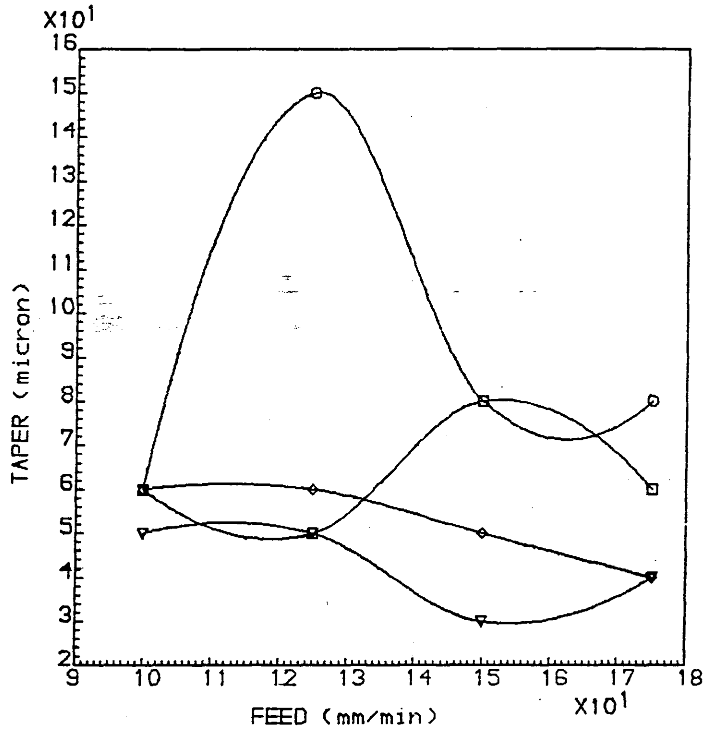
PRF = 10Hz , PULSE ENERGY = 4800mJ

GRAPH 4.32 : TAPER Vs FEED FOR 1.6mm THICK TITANIUM



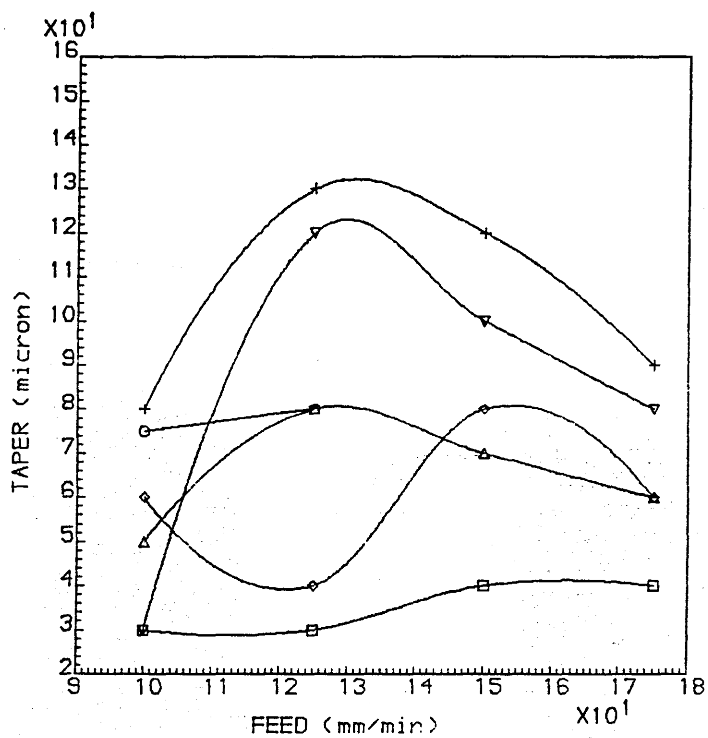


GRAPH 4.33 : TAPER Vs FEED FOR 1.6mm THICK ALUMINIUM



○ 2000mb, 500mic
◇ 3000mb, 500mic
□ 2000mb, 1000mic
▽ 3000mb, 1000mic

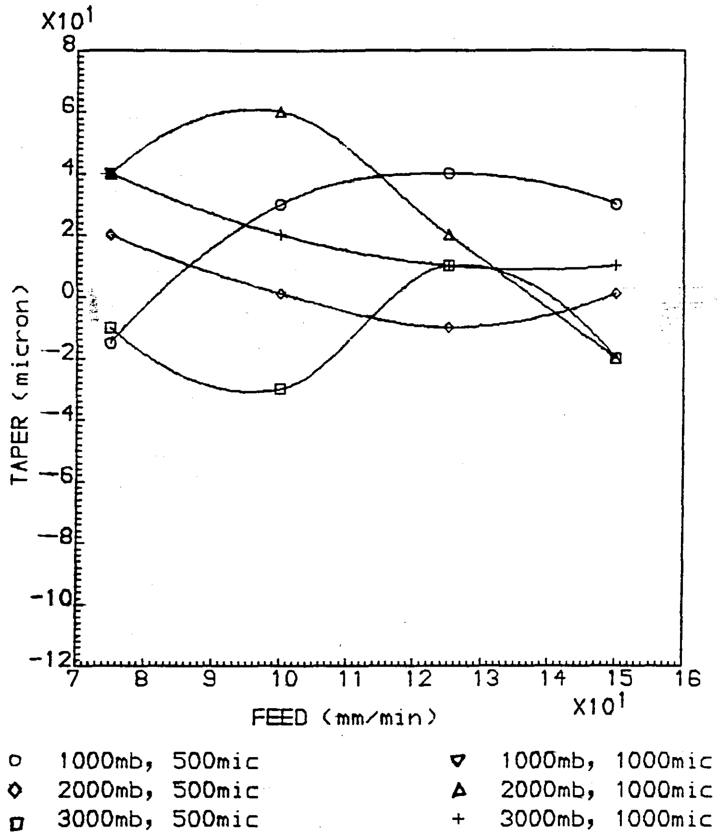
PRF = 15Hz , PULSE ENERGY = 3500mJ



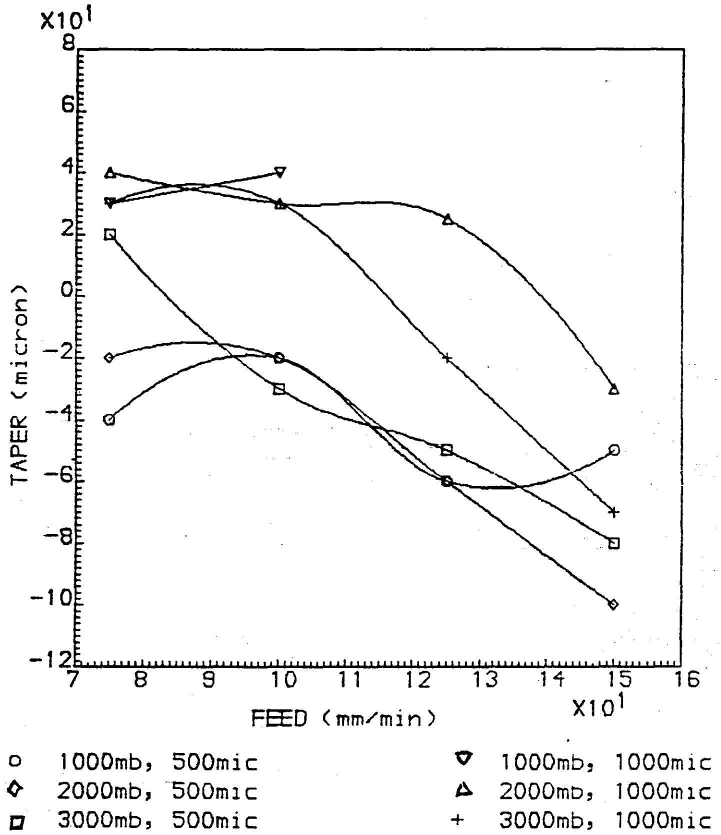
○ 1000mb, 500mic
◇ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
△ 2000mb, 1000mic
\+ 3000mb, 1000mic

PRF = 11Hz , PULSE ENERGY = 5100mJ

GRAPH 4.34 : TAPER Vs FEED FOR 0.9mm THICK NICKEL

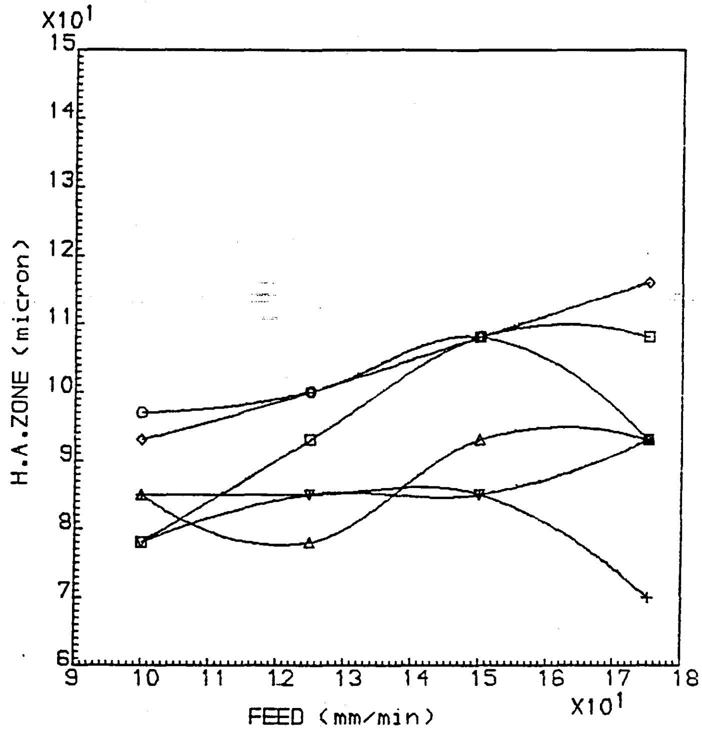


PRF = 15Hz , PULSE ENERGY = 3200mJ



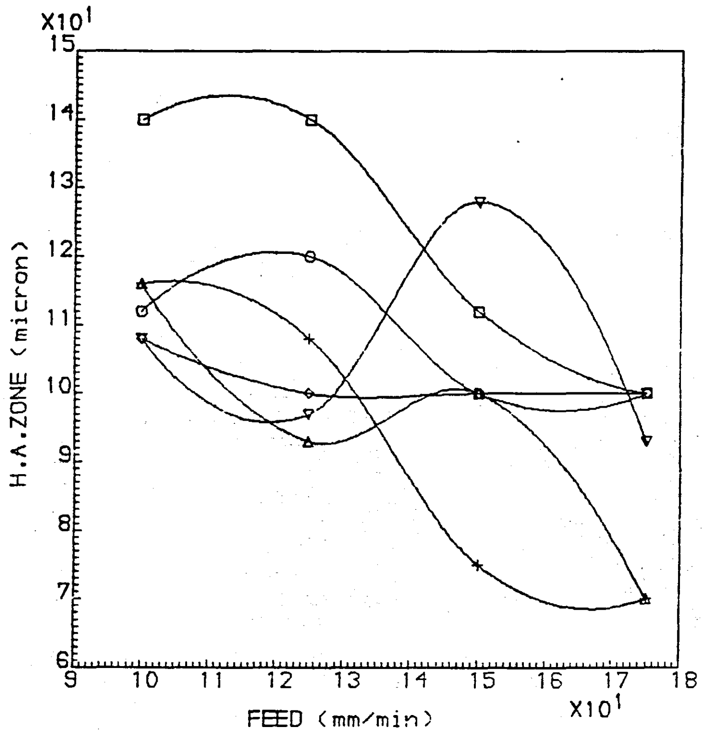
PRF = 10Hz , PULSE ENERGY =4800mJ

GRAPH 4.35 : TAPER Vs FEED FOR 1.6 THICK BRASS



o 1000mb, 500mic
◇ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
△ 2000mb, 1000mic
+ 3000mb, 1000mic

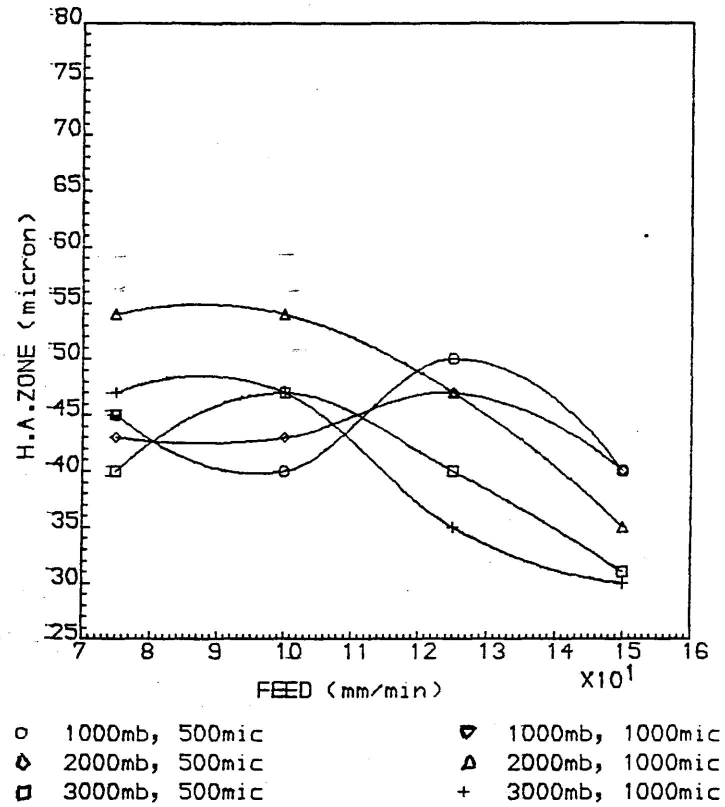
PRF = 15Hz , PULSE ENERGY = 3200mJ



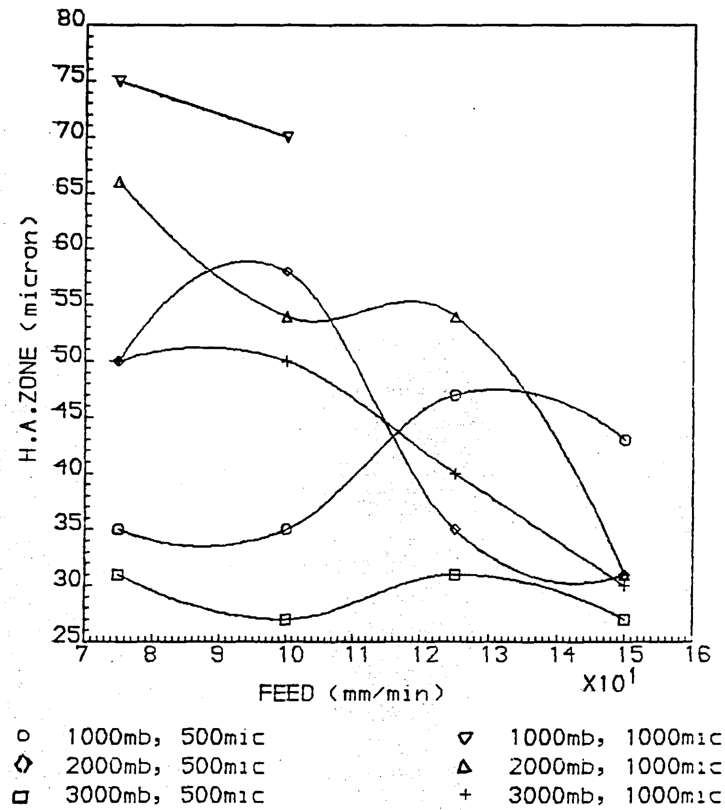
o 1000mb, 500mic
◇ 2000mb, 500mic
□ 3000mb, 500mic
▽ 1000mb, 1000mic
△ 2000mb, 1000mic
+ 3000mb, 1000mic

PRF = 10Hz , PULSE ENERGY = 4800mJ

GRAPH 4.36 : HEAT AFFECTED ZONE Vs FEED FOR 1.6mm THICK TITANIUM

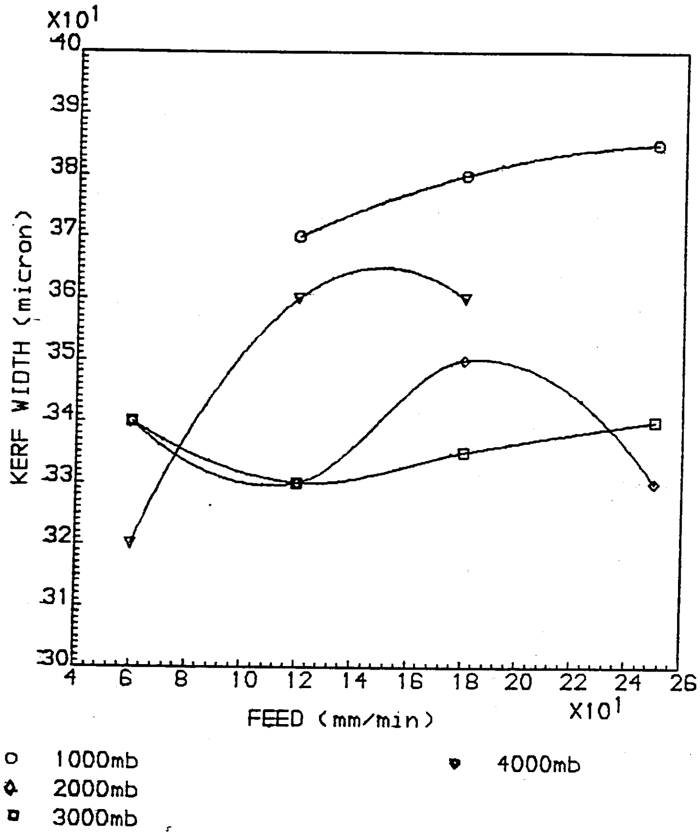


PRF = 15Hz , PULSE ENERGY = 3200mJ



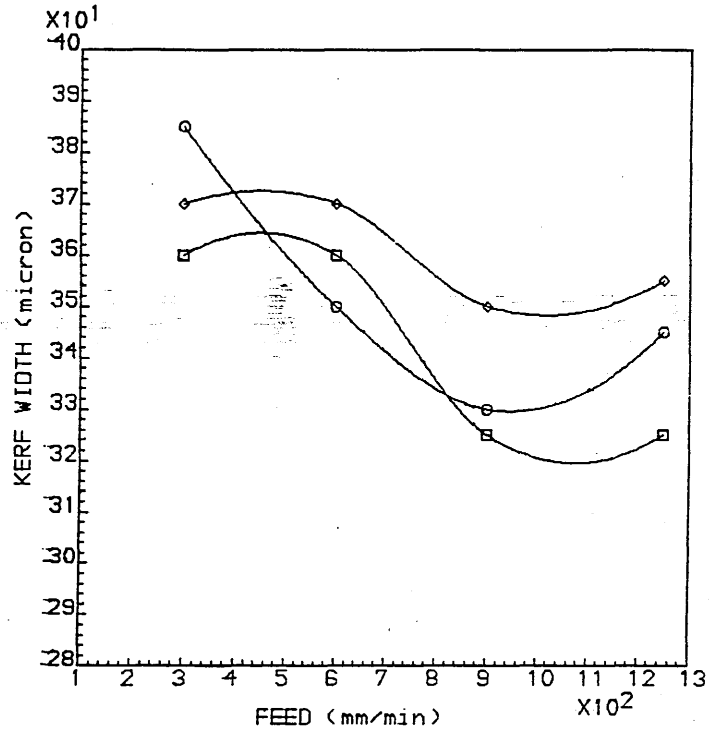
PRF = 10Hz , PULSE ENERGY =4800mJ

GRAPH 4.37 : HEAT AFFECTED ZONE Vs FEED FOR 1.6 THICK BRASS



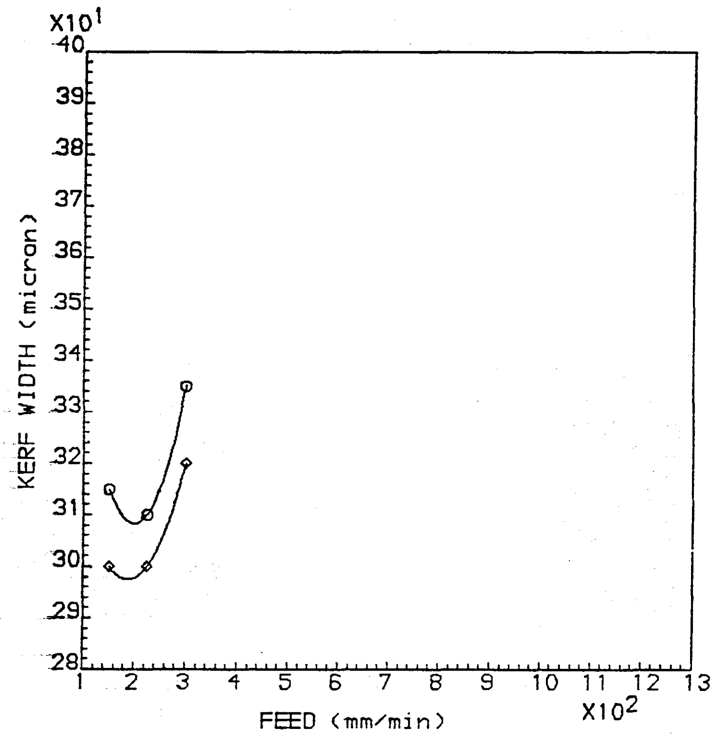
POWER = 210 W

GRAPH 4.38 :KERF WIDTH Vs FEED FOR 3.16mm MILD STEEL



o 1000mb
◊ 2000mb
□ 3000mb

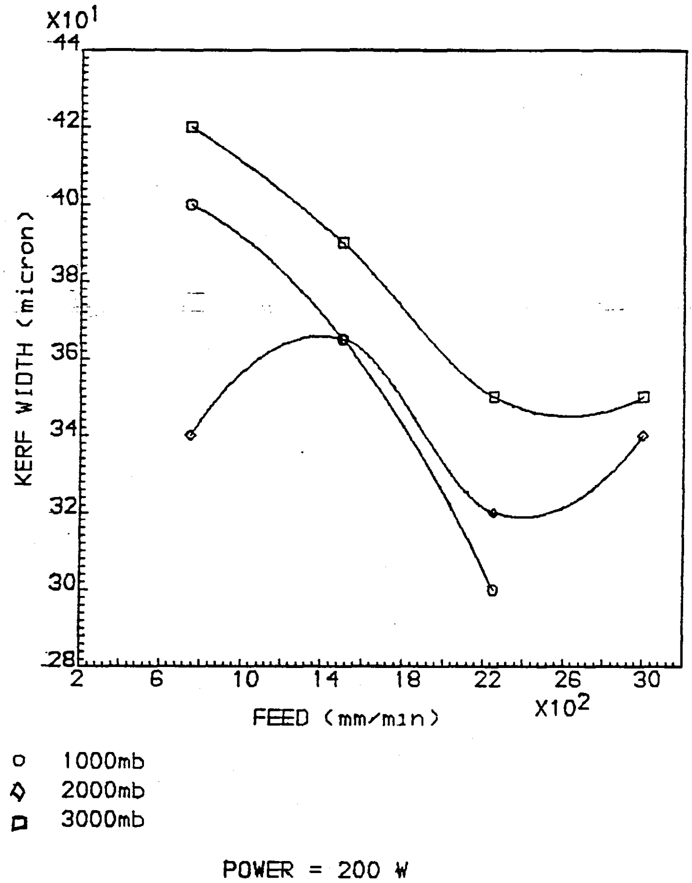
POWER = 200 W

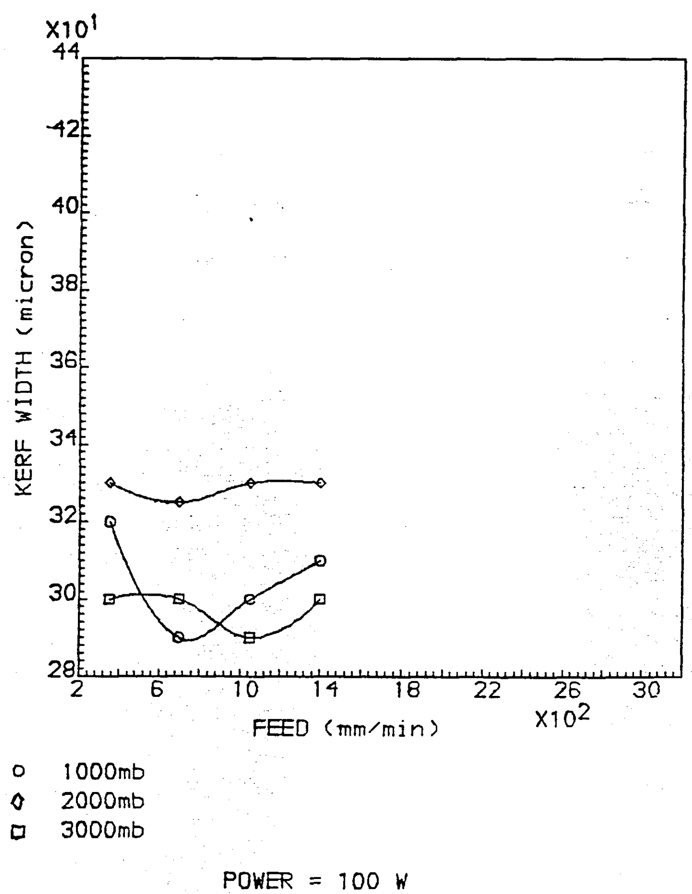


o 2000mb
◊ 3000mb

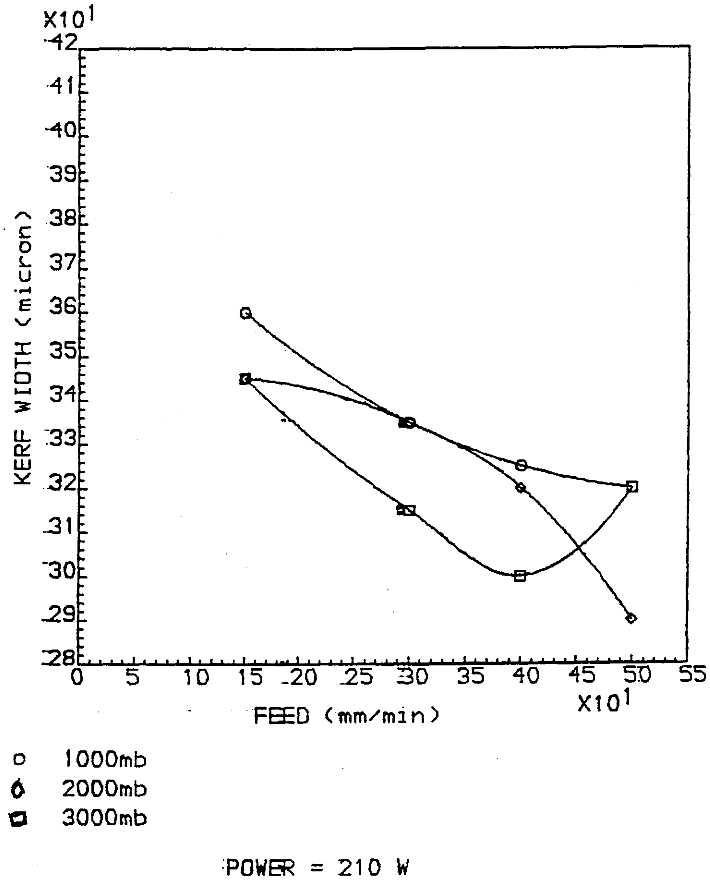
POWER = 120 W

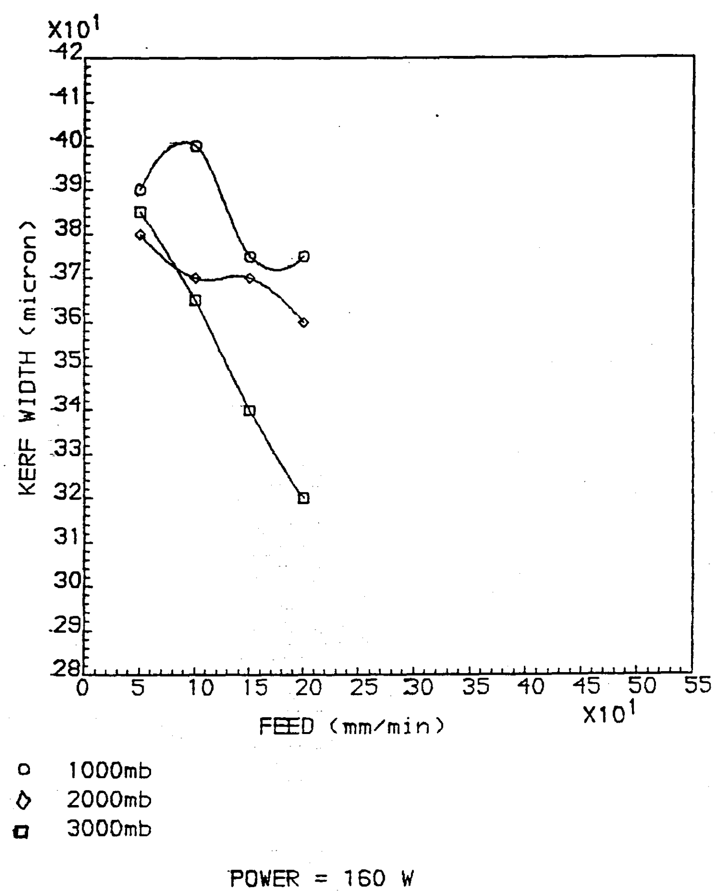
GRAPH 4.39 : KERF WIDTH Vs FEED FOR 1.6mm THICK MILD STEEL



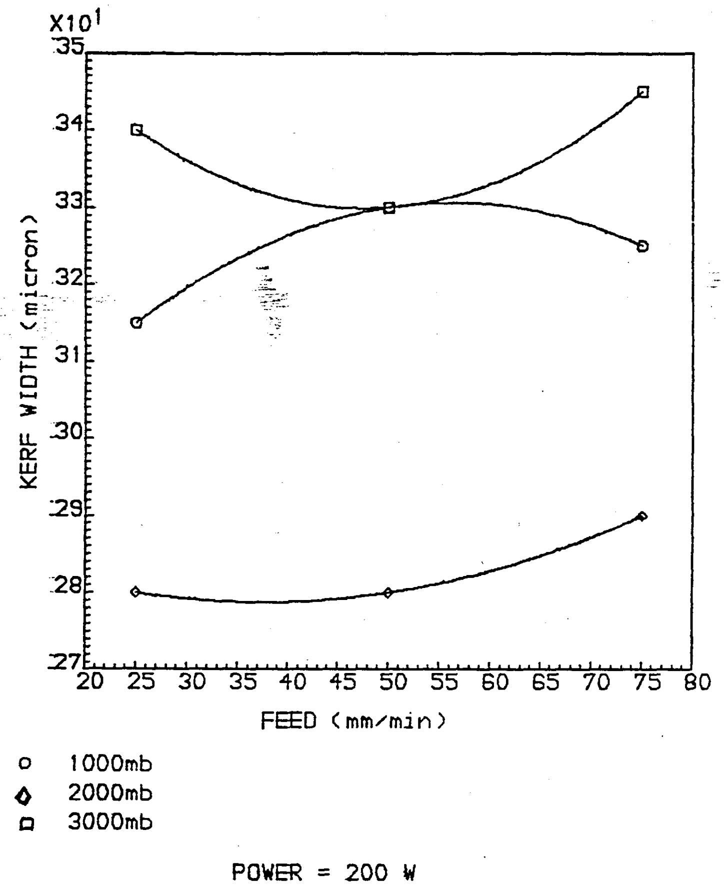


GRAPH 4.40 : KERF WIDTH Vs FEED FOR 0.8mm THICK MILD STEEL

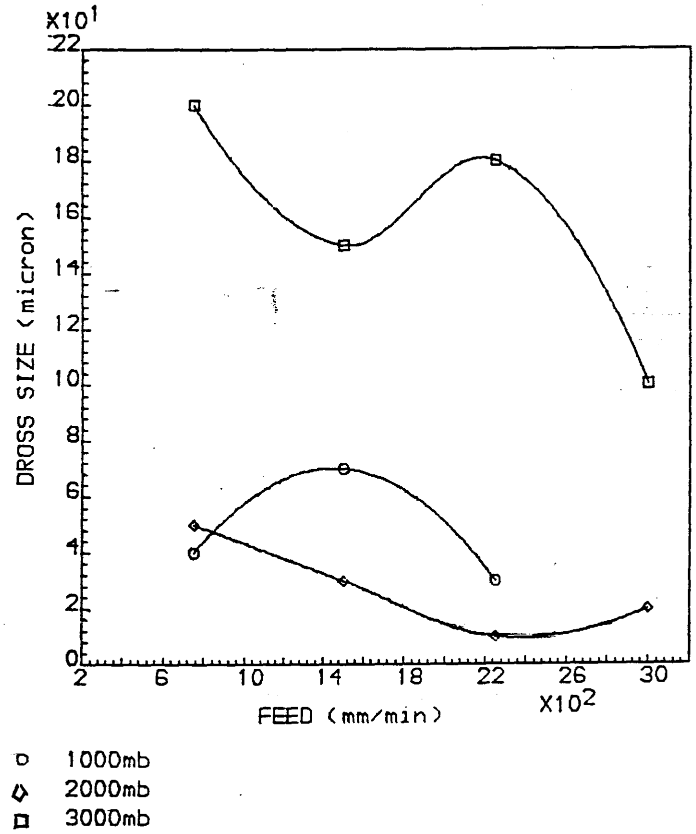




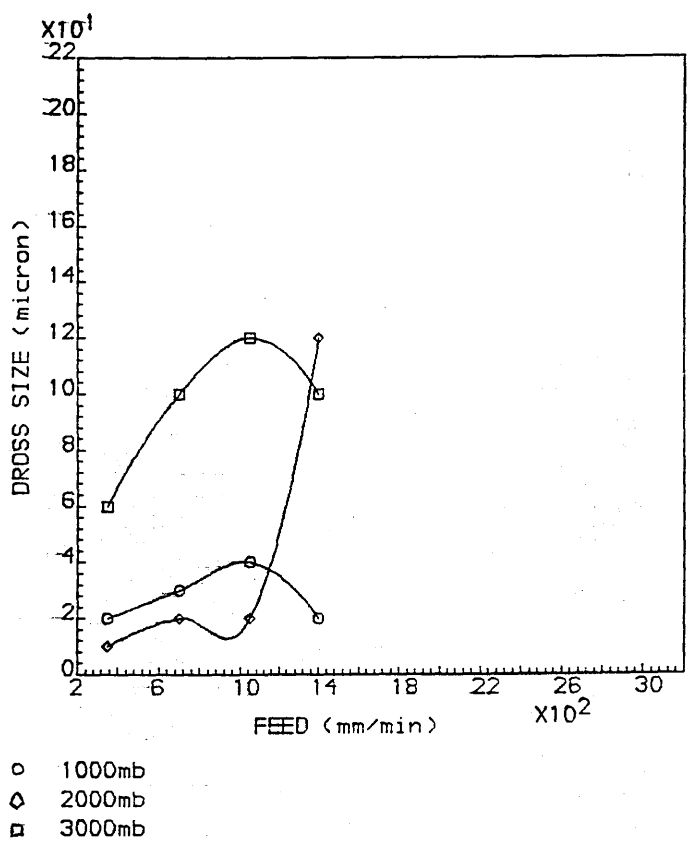
GRAPH 4.41 : KERF WIDTH Vs FEED FOR 1.6mm THICK STAINLESS STEEL



GRAPH 4.42 : KERF WIDTH Vs FEED FOR 0.75mm THICK INCONEL

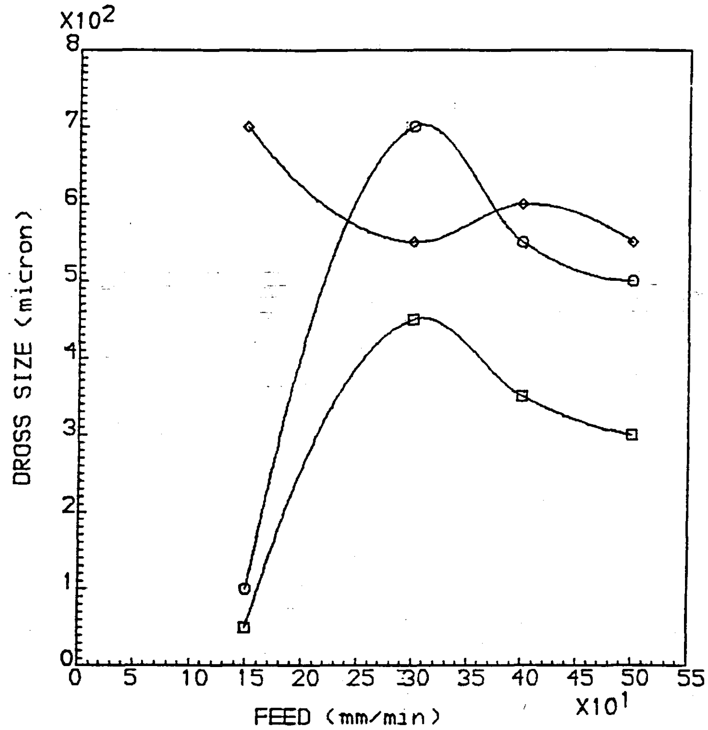


POWER = 200 W



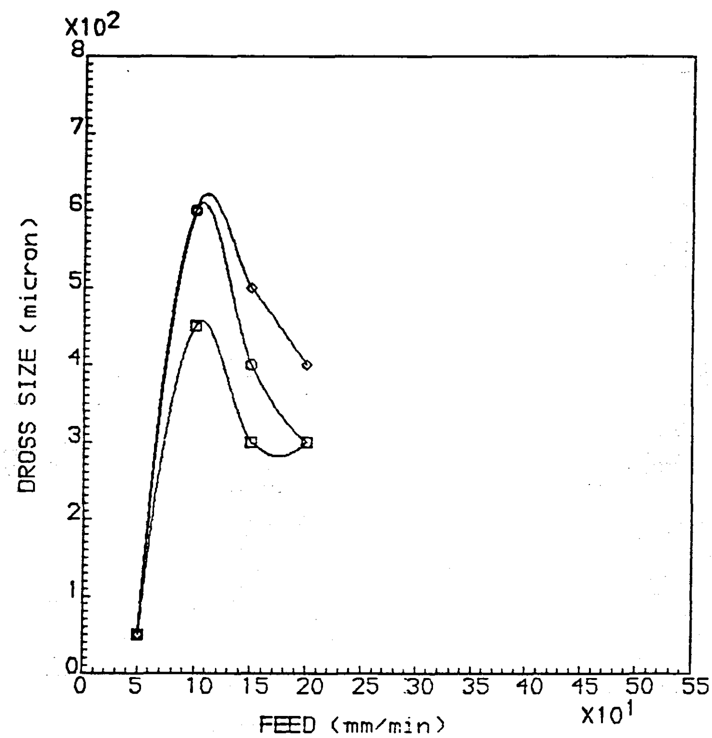
POWER = 100 W

GRAPH 4.43 : DROSS SIZE Vs FEED FOR 0.8mm THICK MILD STEEL



o 1000mb
◊ 2000mb
□ 3000mb

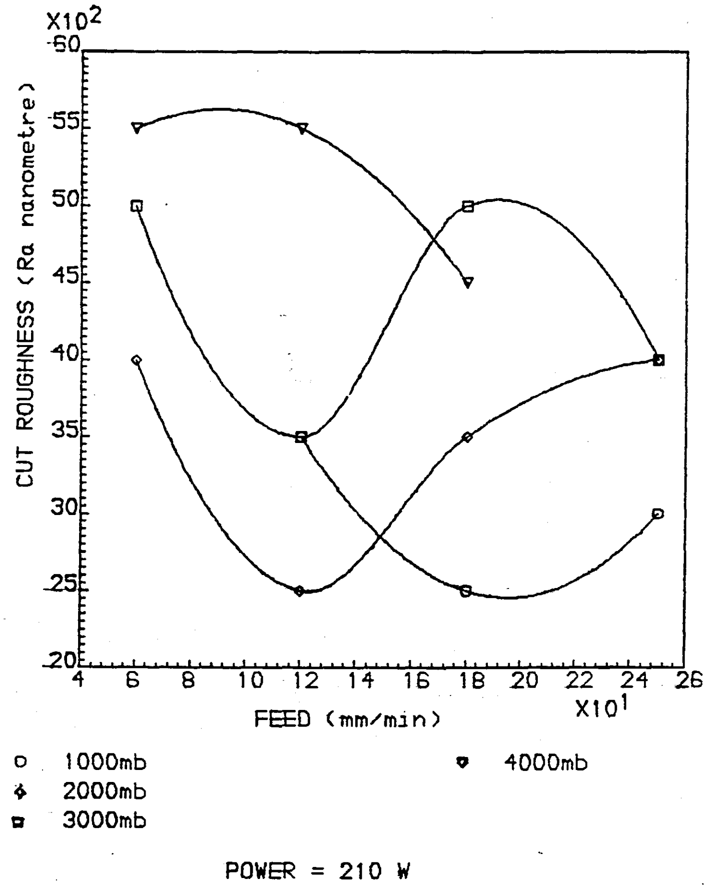
POWER = 210 W



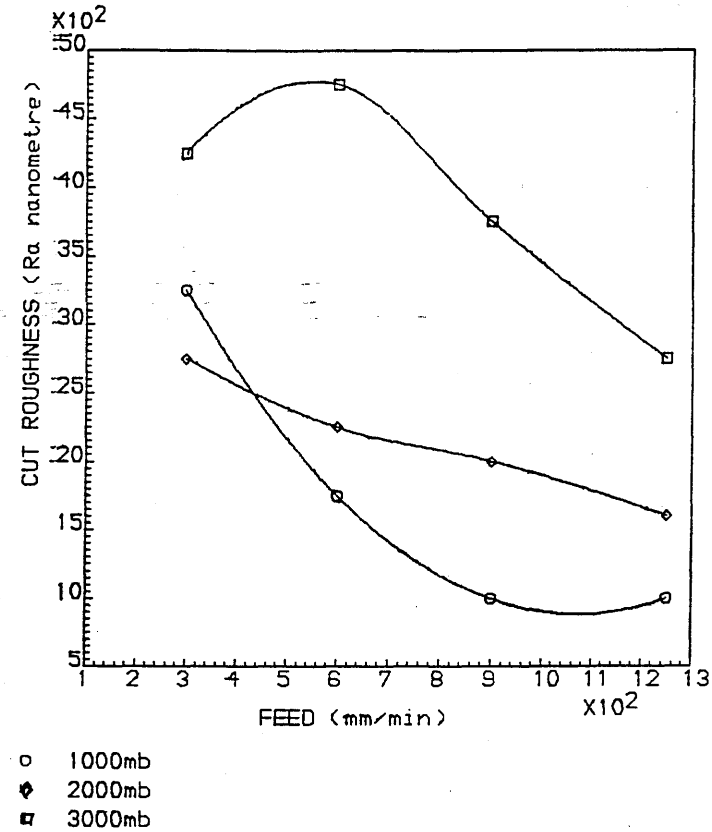
o 1000mb
◊ 2000mb
□ 3000mb

POWER = 160 W

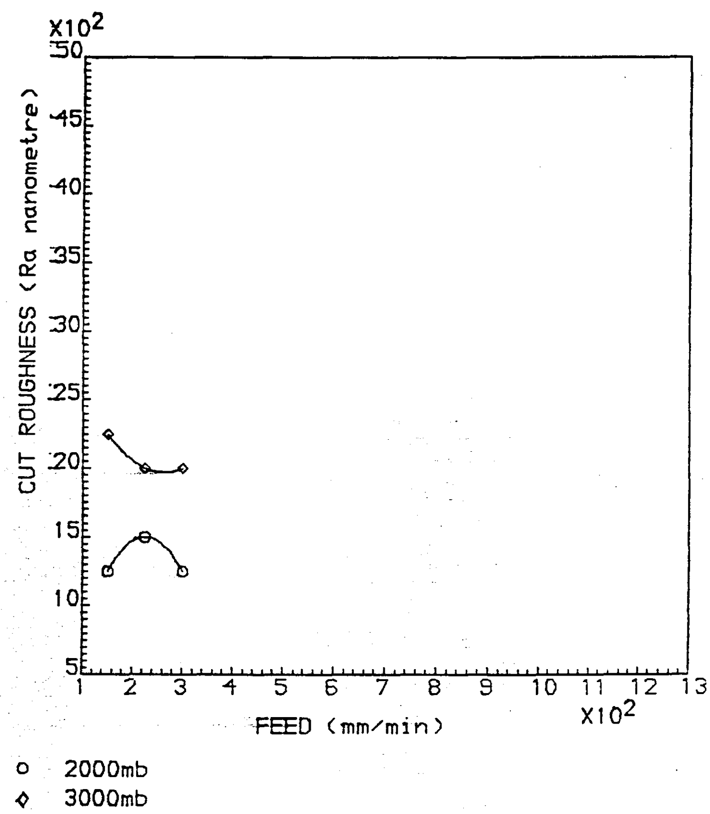
GRAPH 4.44 : DROSS SIZE Vs FEED FOR 1.6mm THICK STAINLESS STEEL



GRAPH 4.45 : CUT ROUGHNESS Vs FEED FOR 3.16mm MILD STEEL

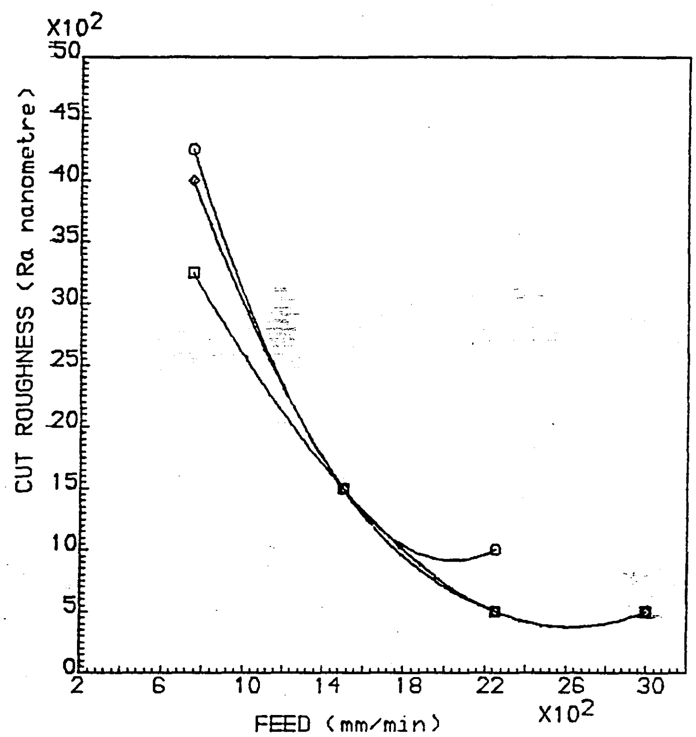


POWER = 200 W



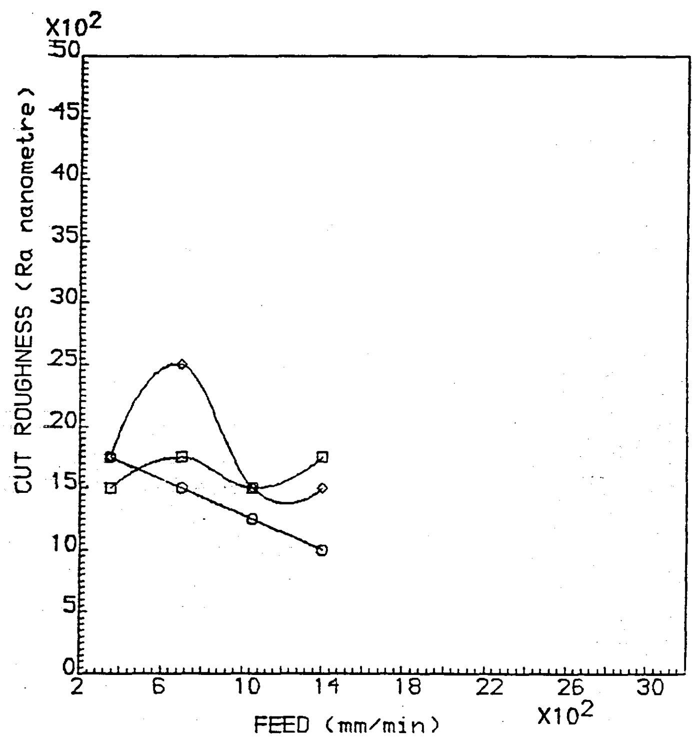
POWER = 120 W

GRAPH 4.46 : CUT ROUGHNESS Vs FEED FOR 1.6mm THICK MILD STEEL



o 1000mb
◊ 2000mb
□ 3000mb

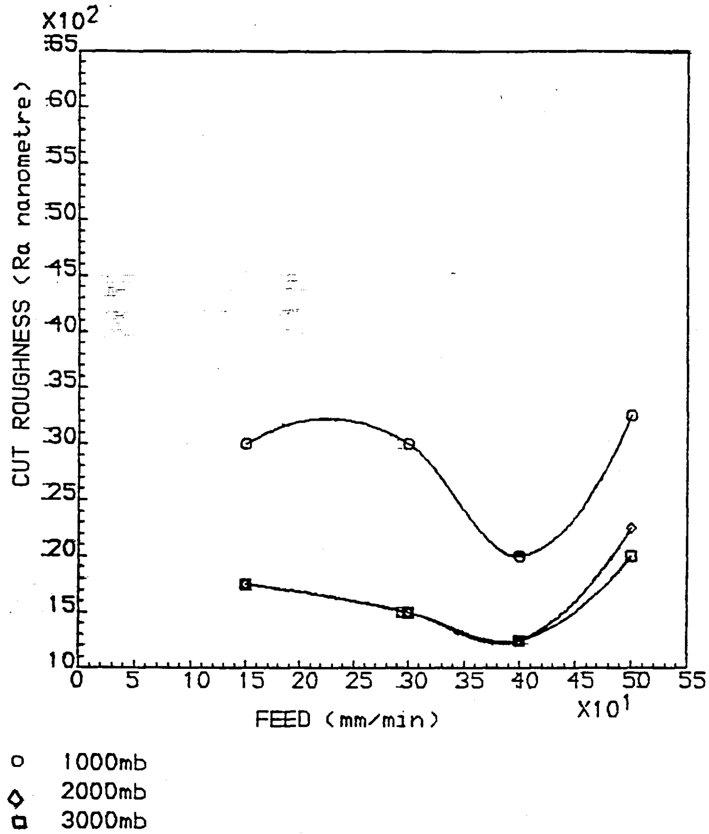
POWER = 200 W



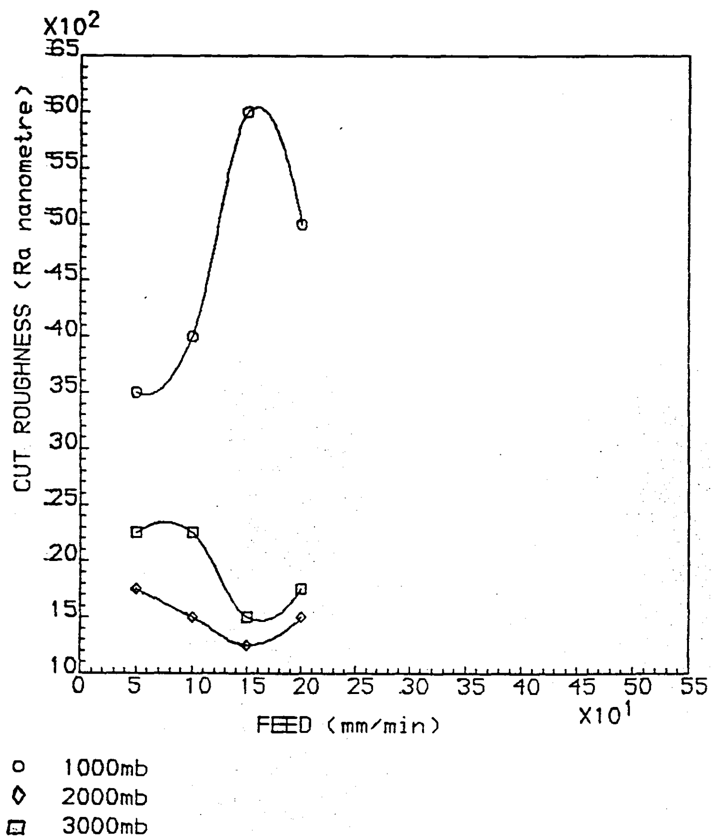
o 1000mb
◊ 2000mb
□ 3000mb

POWER = 100 W

GRAPH 4.47 : CUT ROUGHNESS Vs FEED FOR 0.8mm THICK MILD STEEL

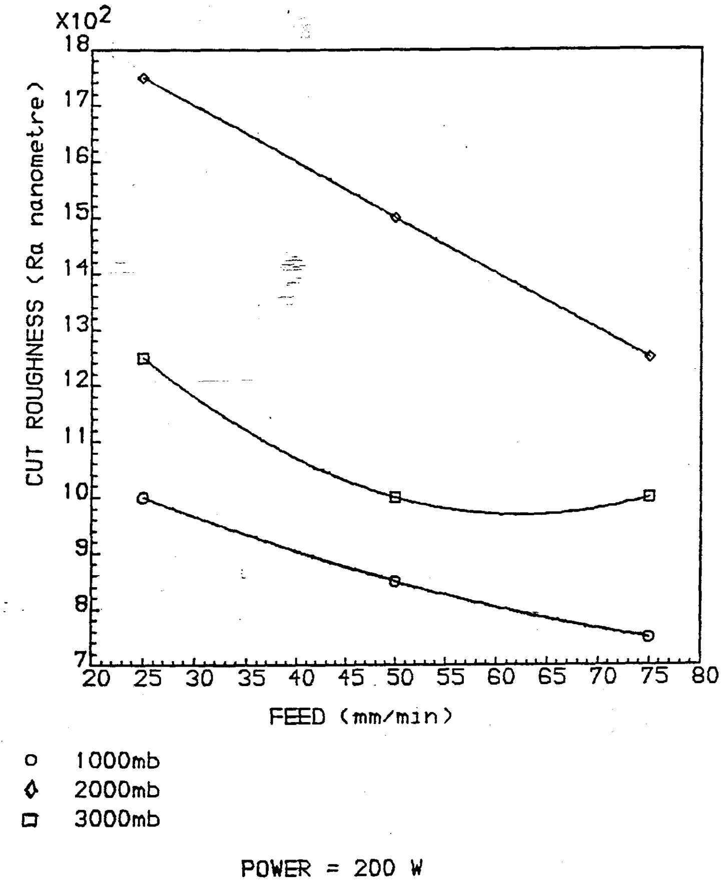


POWER = 210 W

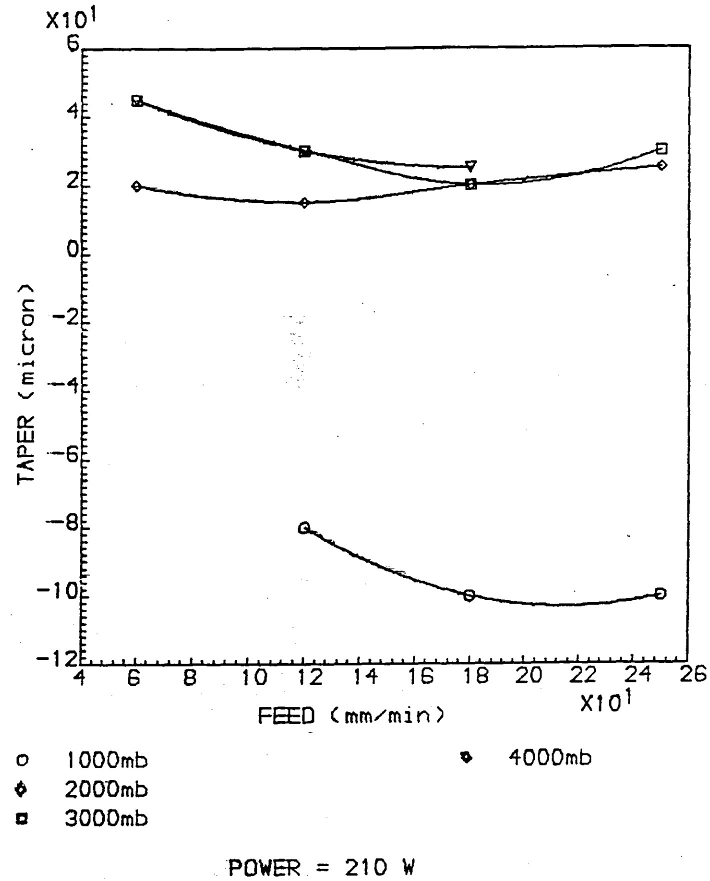


POWER = 160 W

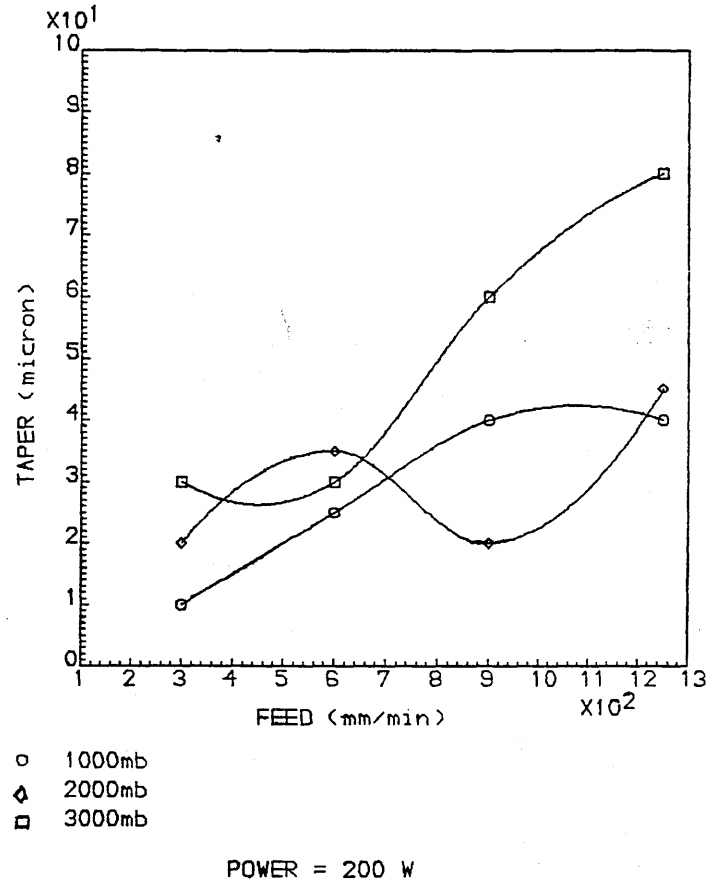
GRAPH 4.48 : CUT ROUGHNESS Vs FEED FOR 1.6mm THICK STAINLESS STEEL

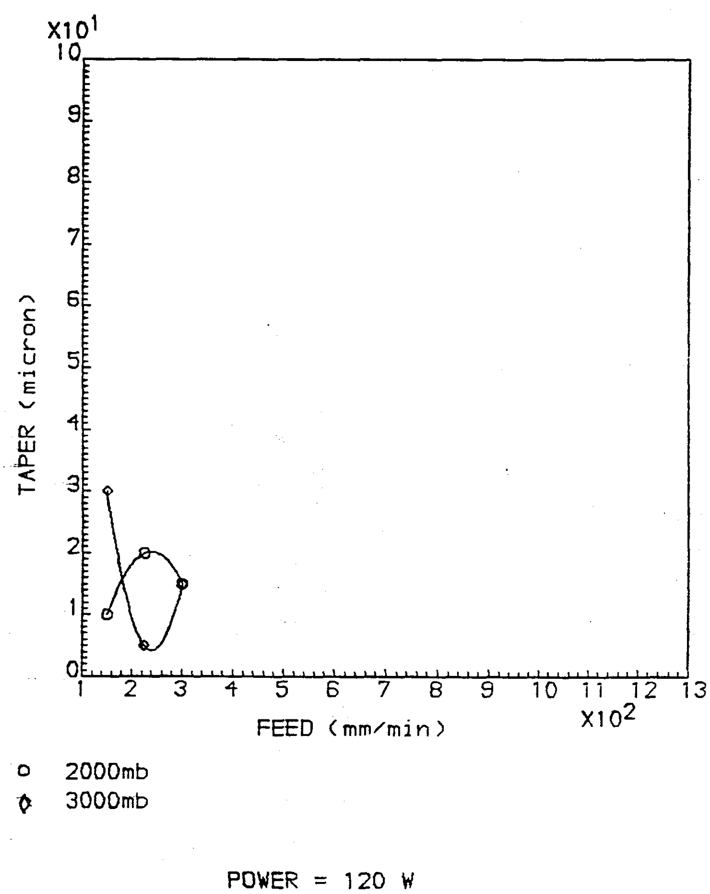


GRAPH 4.49 : CUT ROUGHNESS Vs FEED FOR 0.75mm THICK INCONEL

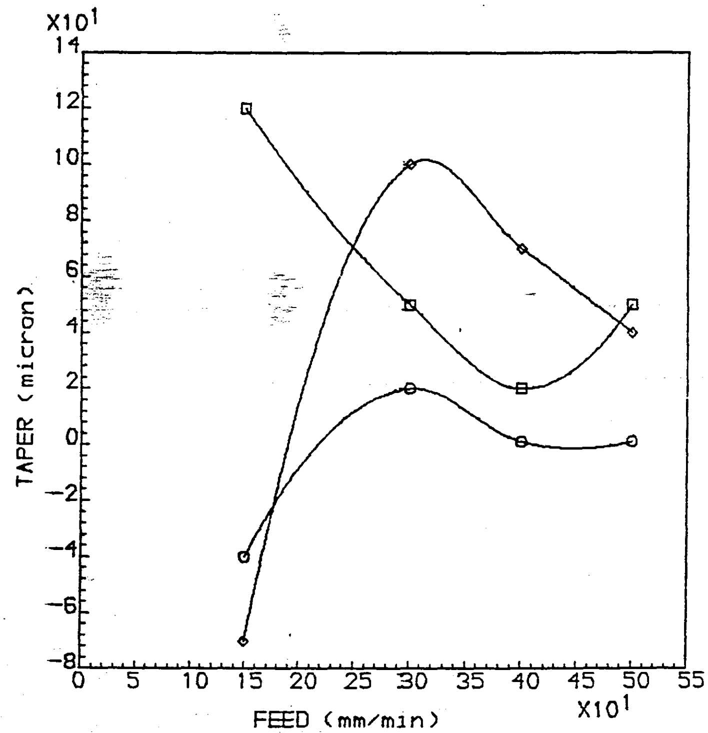


GRAPH 4.50 :TAPER Vs FEED FOR 3.16mm MILD STEEL



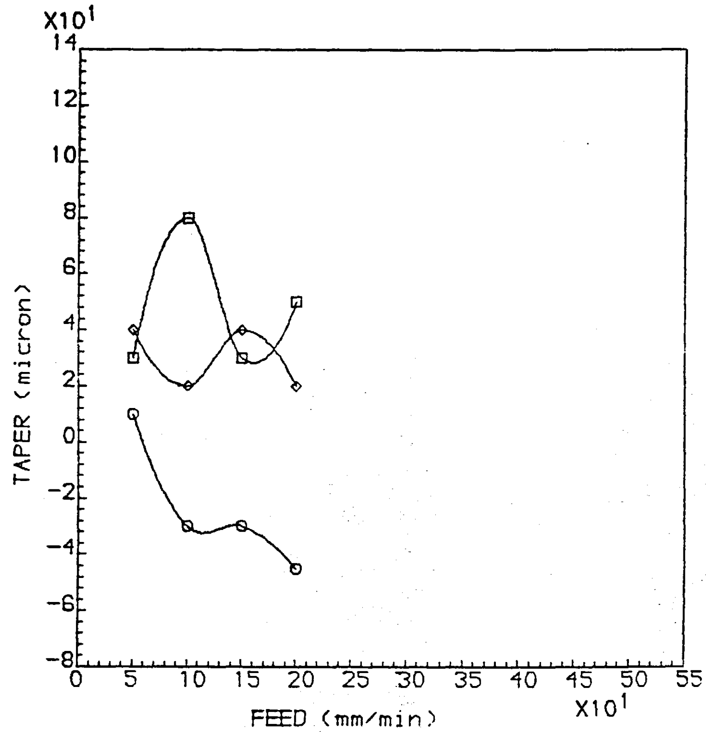


GRAPH 4.51 : TAPER Vs FEED FOR 1.6mm THICK MILD STEEL



○ 1000mb
◇ 2000mb
□ 3000mb

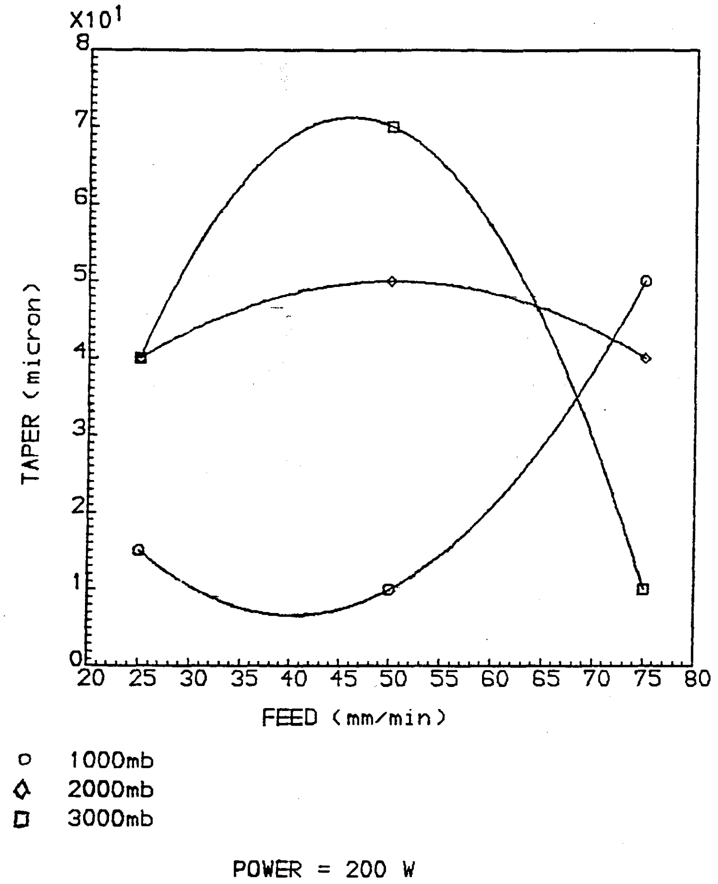
POWER = 210 W



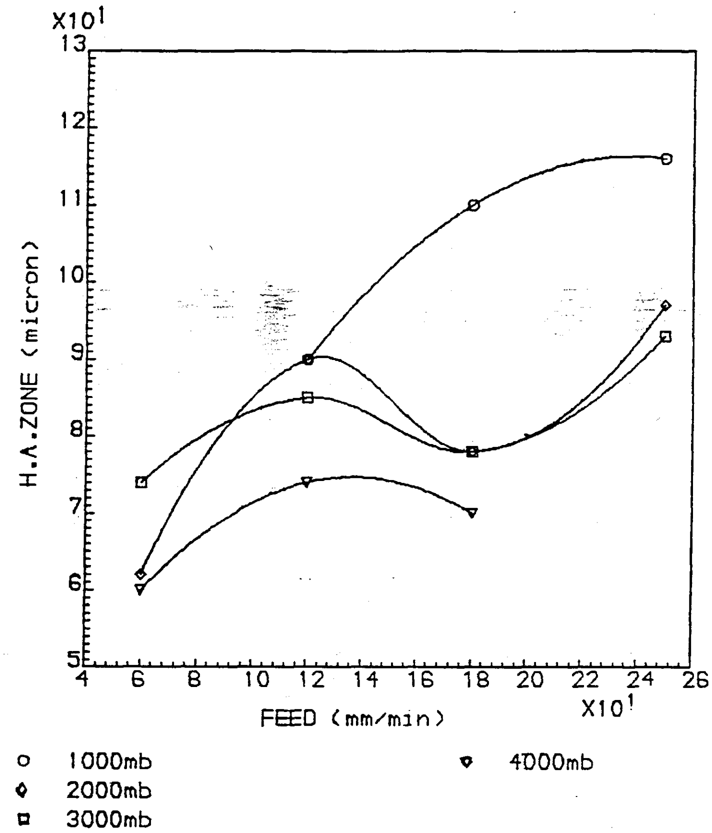
○ 1000mb
◇ 2000mb
□ 3000mb

POWER = 160 W

GRAPH 4.52 : TAPER Vs FEED FOR 1.6mm THICK STAINLESS STEEL

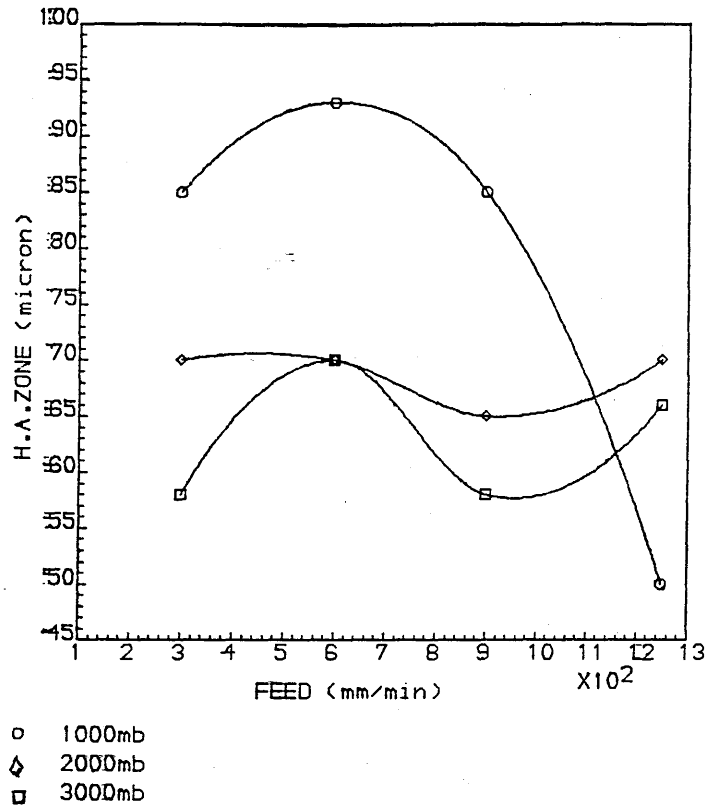


GRAPH 4.53 : TAPER Vs FEED FOR 0.75mm THICK INCONEL

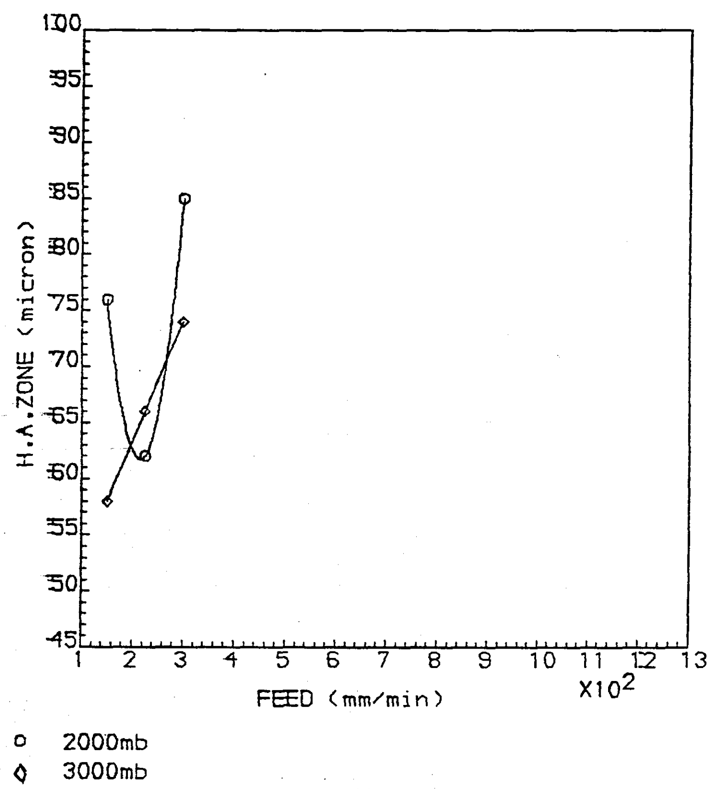


POWER = 210 W

GRAPH 4.54 : HEAT AFFECTED ZONE Vs FEED FOR 3.16mm MILD STEEL

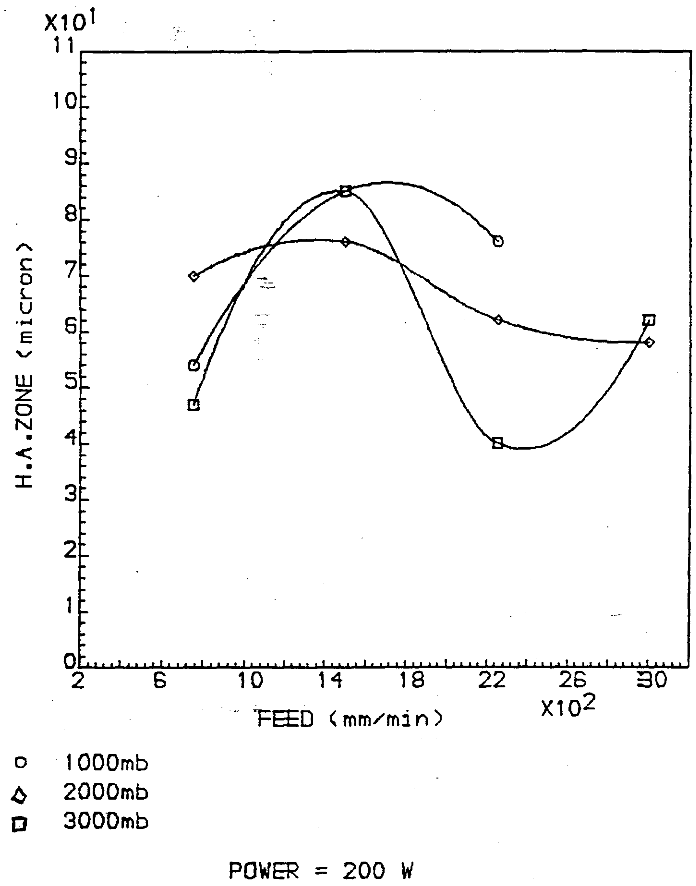


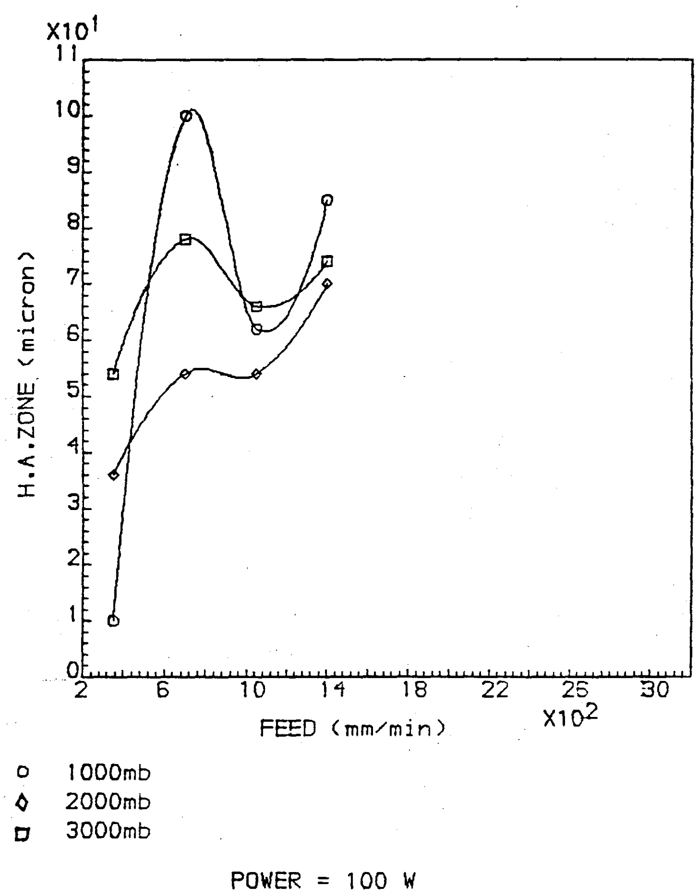
POWER = 200 W



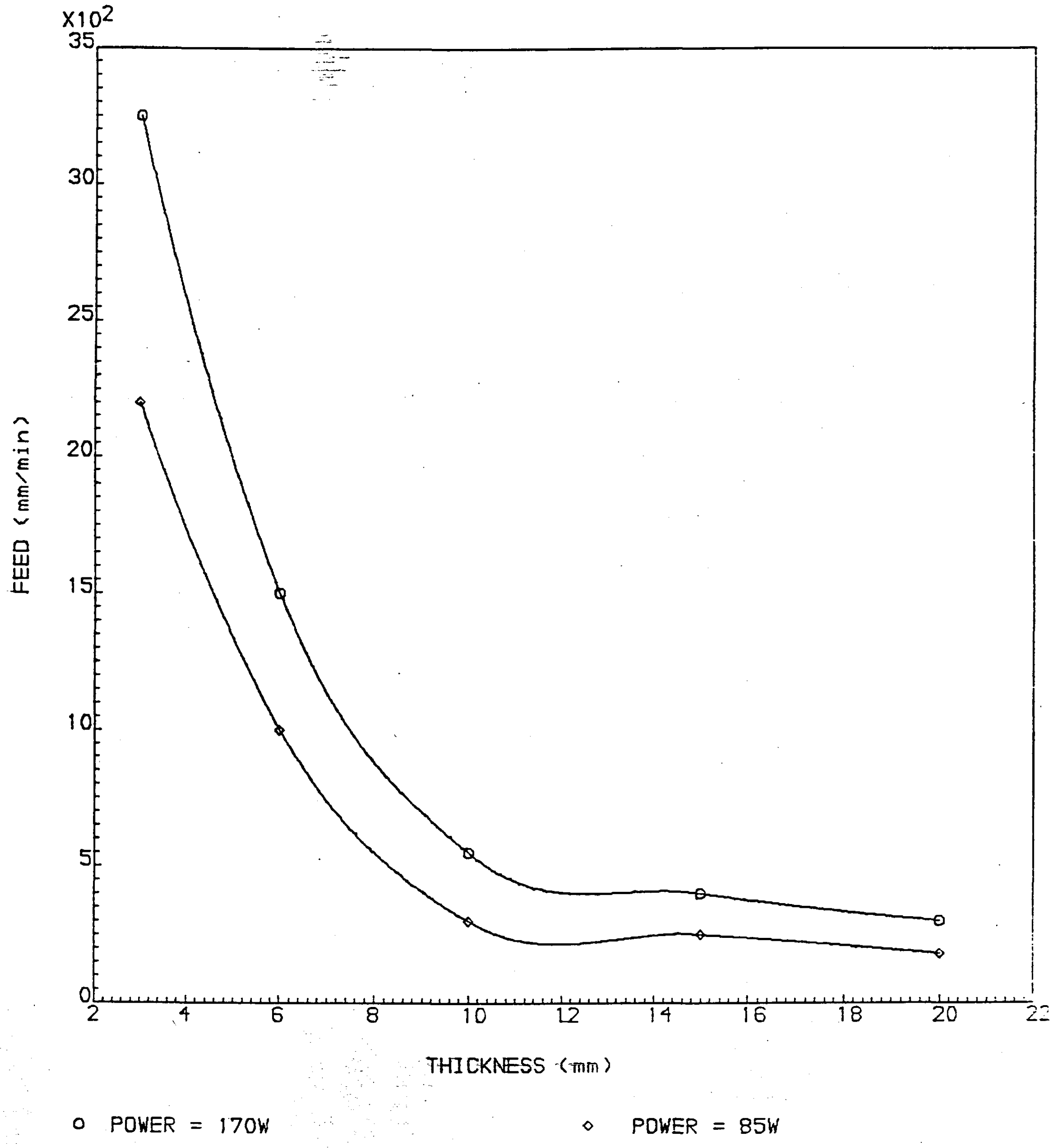
POWER = 120 W

GRAPH 4.55 : HEAT AFFECTED ZONE Vs FEED FOR 1.6mm THICK MILD STEEL





GRAPH 4.56 : HEAT AFFECTED ZONE Vs FEED FOR 0.8mm THICK MILD STEEL



GRAPH 4.57 : MAXIMUM FEED Vs THICKNESS FOR PLASTIC CUTTING

CHAPTER 5

THE DATA COLLECTION MODULE

Chapter 2 showed that a module is needed to search the database to collect data relevant for the target application. This module together with the prediction module, constitutes a Task Planning Executive and Control Strategy Generator. A two module approach was adopted as the requisite software resides on two separate machines: the database on an IBM 6150 minicomputer, and the algorithmic and graphical software required for the prediction program on a VAX 11/750 machine. The final system would reside on a single machine.

The following main aspects are discussed in this chapter: The conceptual structure of the collection module, the database search strategy and the implementation of this module in a computer program.

5.1 OBJECTIVE AND TASKS

The main objective of this module can be defined as: The provision of a set of data, that is both relevant to the target application and satisfactory for the operation of the prediction module. The data which is submitted to the prediction program is composed of two types: data supplied by the user, and data retrieved from the database system.

This objective was fulfilled by accomplishing the following tasks:

1. User input.
2. Database search.
3. Data output.

The objective of user input is the acquisition of all the data necessary for the operation of the search procedure and the prediction module. The input process is guided by a system of menu-driven inquiry. Information from the database is shown during the inquiry to help the user to select feasible answers.

The database search for a satisfactory set of data (relevant to the target application) is guided and controlled by a system of search rules, that organises the search according to the user input, and evaluates the adequacy of the collected data.

The module output is via an output file which contains all data needed by the prediction program to operate properly.

5.2 REQUIREMENTS AND DESIRED CHARACTERISTICS

A number of characteristics are required for this module, to ensure a reasonable level of performance and accuracy, these requirements include:

1) Flexibility: To provide ease of alteration to the program, required for future development.

2) Expandability: To allow the inclusion of more processes (i.e search rules), and the inclusion of more parameters (variables) in existing and new processes.

3) Data integrity: This is a prime requirement for both the collection and prediction modules. The success of the search for data depends on the integrity of the user supplied information (provided that there are relevant data in the database). While the success of the prediction program, depends on the volume and integrity of the data collected by the collection module.

Flexibility and expandability were achieved by designing a modular structure for the program. The collection module (collector) is divided into two main submodules. Each of which is divided into several independent divisions (procedures), then into subdivisions (blocks) and down to individual processes. In this way, data processing can be manipulated at a number of levels, by the substitution of submodules or divisions. These two characteristics are supported by ensuring that the main flow of the program processing is unidirectional.

Integrity was enforced by using the following techniques:

- Menu-driven user interrogation, embedded with informative messages.

- Rollback of the user input operation and provision of warning messages, when the user response contains a detectable error (i.e an error that can be detected by the test power of the program host language).

- If a user input data value is mandatory, the program will cease at this point, until the user responds correctly.

- Termination of the program, and echo of error messages, when an error is encountered during the process of communication with SQL/RT database.

5.3 THE CONCEPTUAL STRUCTURE OF THE COLLECTION MODULE

The first prototype of the collection module was built prior to the development of the prediction module, and after the database module prototype. The cutting experiments had not yet commenced, due to the reasons explained in section 3.4.1, and the prototype was designed and implemented on the basis of conceived theoretical knowledge of laser cutting, and experience gained from working with the SQL/RT database. The model was revised subsequently, according to observations made during the

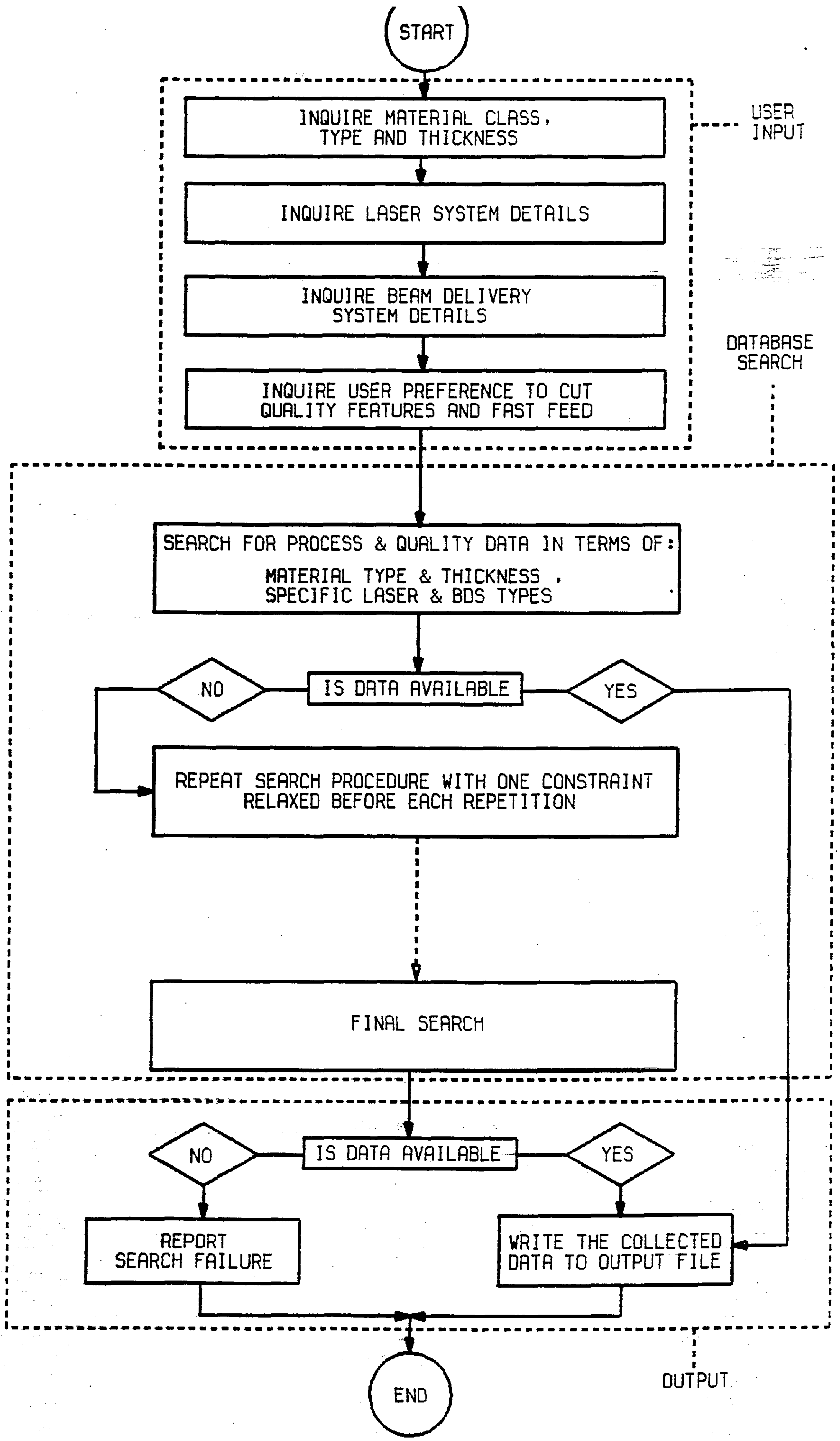


CHART 5.1 : OVERALL FLOW CHART OF THE DATA COLLECTION MODULE

experimental programme and to accomodate the requirements ot the prediction module.

The ultimate function of the collector, is to search the database for a specific set of data based on the requirements (constraints) given by the user, which defines his application. Consequently, user-interrogation becomes a major design issue.

This would suggest that the conceptual model of the data collector consists of two major submodules: User interrogation and database search. Chart 5.1 illustrates these two units schematically, together with the main underlying procedures within them. Each procedure is independent of the others (modular), and the flow of processing is unidirectional. These two features are necessarily dominant in all of the subsequent divisions of this module. These submodules are described in more detail.

5.3.1 THE USER-INPUT SUBMODULE

The data acquired from the user falls into the following categories:

1. Information required for both the search unit and prediction module, e.g material thickness and laser type.

2.Information necessary for the search unit only, e.g type of material.

3.Information necessary for the prediction module only, e.g preference weight of cut quality features.

4.Provisional information that may be needed for future development, e.g material constituents and focal length of the focusing lens.

User supplied information is classified as mandatory or optional. Mandatory data is that which failure to provide would prevent any further processing. This is because either or both the search and prediction procedures requires this data for operation. Material thickness is an example of this type of data. Optional data is either that of secondary importance (to the present design), or provisional for later development.

The overall flow of the user input procedure is shown in chart 5.1. This queries the user about: The material, laser system, beam delivery system and the preference of each of cut quality features. These inquiries are explained below.

The following nomenclature is used in the flow charts of this chapter:

- The term INQUIRE requires the user to provide a value or an answer.

- The term INTERROGATE is a search action upon the database.

- An oval box denotes a user action or response.

a)MATERIAL DETAILS INQUIRY: This is responsible for the acquisition of data related to the material to be slit (chart 5.2).

The first inquiry block determines if user input is to be given interactively or through an input data file. Data acquisition from a data file has not been implemented, as the interactive mode leads to a better perception of the data input process, and to the importance of each of the supplied details of information.

The second block displays the classes of materials that are available in the database module. Each class represents an individual table of material (as per chapter 3). The user selects the class to which the target material belongs.

The third step uses a program action to interrogate the database as to the distinct types of materials included in the selected table (class). These types are then displayed to the user, and an inquiry is made to determine if any of them matches his type. Then, it proceeds according to the

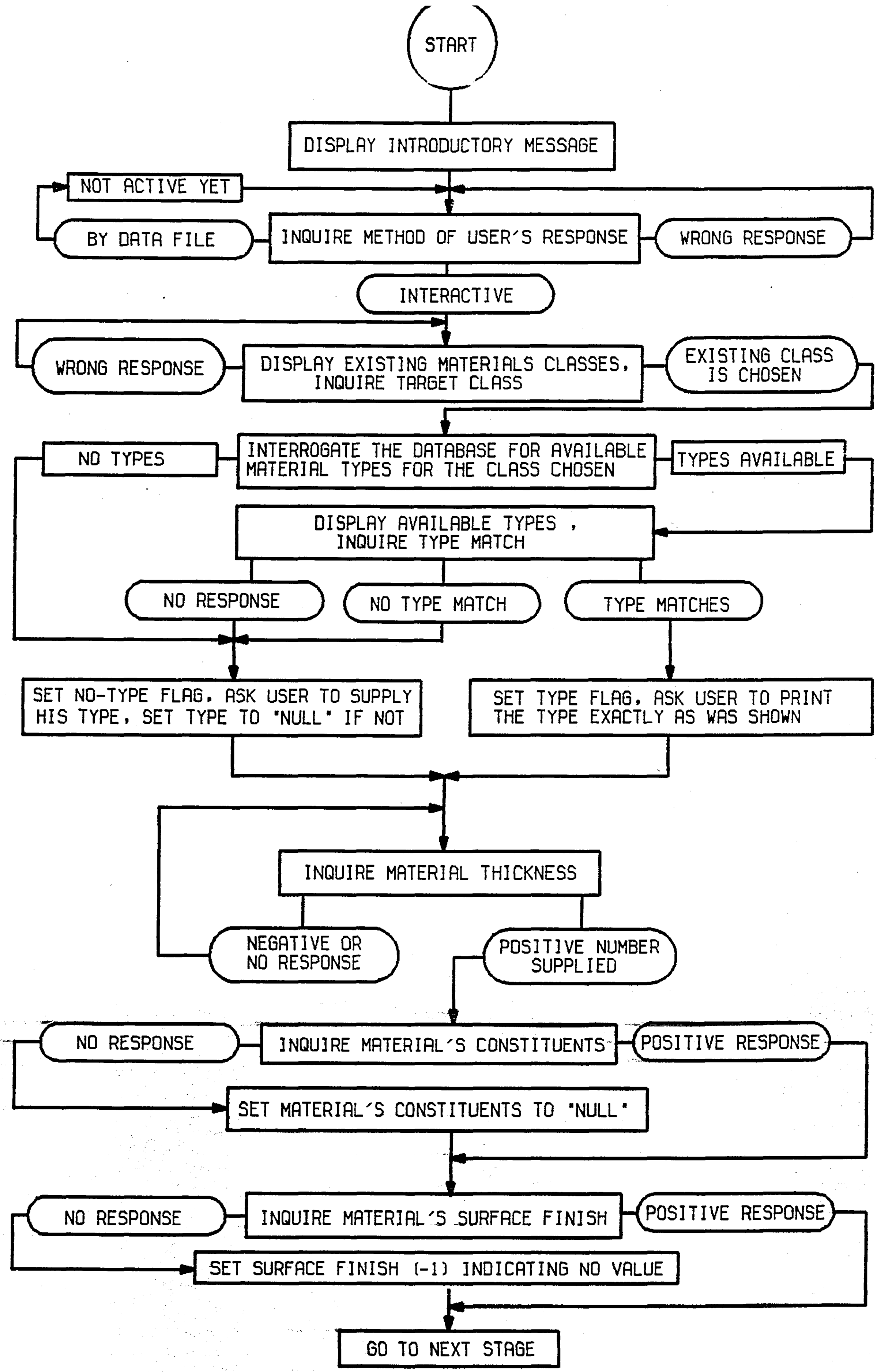


CHART 5.2 : MATERIAL DETAILS INQUIRY PROCEDURE

user response, and as indicated in the chart.

b)INQUIRY OF THE LASER SYSTEM DETAILS: This division is used to specify details of the user's particulars laser (chart 5.3).

The first block inquires the user to select a method of inputing laser details from three options. The first option is the direct selection of a laser system known to be in the database, by its identification code "CL". The second option causes an interrogation of the database as to the laser systems already detailed, and a table containing these information to be displayed to the user. If the user reckons that any of the shown systems matches his own, he responds by "YES" to the next query, and types the laser system code at the following query.

The third option is to supply laser details individually. The module is designed to insist on acquiring three data entries, these are: Type of the laser, its maximum power and whether it is pulsed or continuous wave (CW).

The other four queries are optional as they are not used in the current search strategy.

c)INQUIRY OF THE BEAM DELIVERY SYSTEM DETAILS: This division is used to specify details of the user's beam delivery system. The procedure of this division is

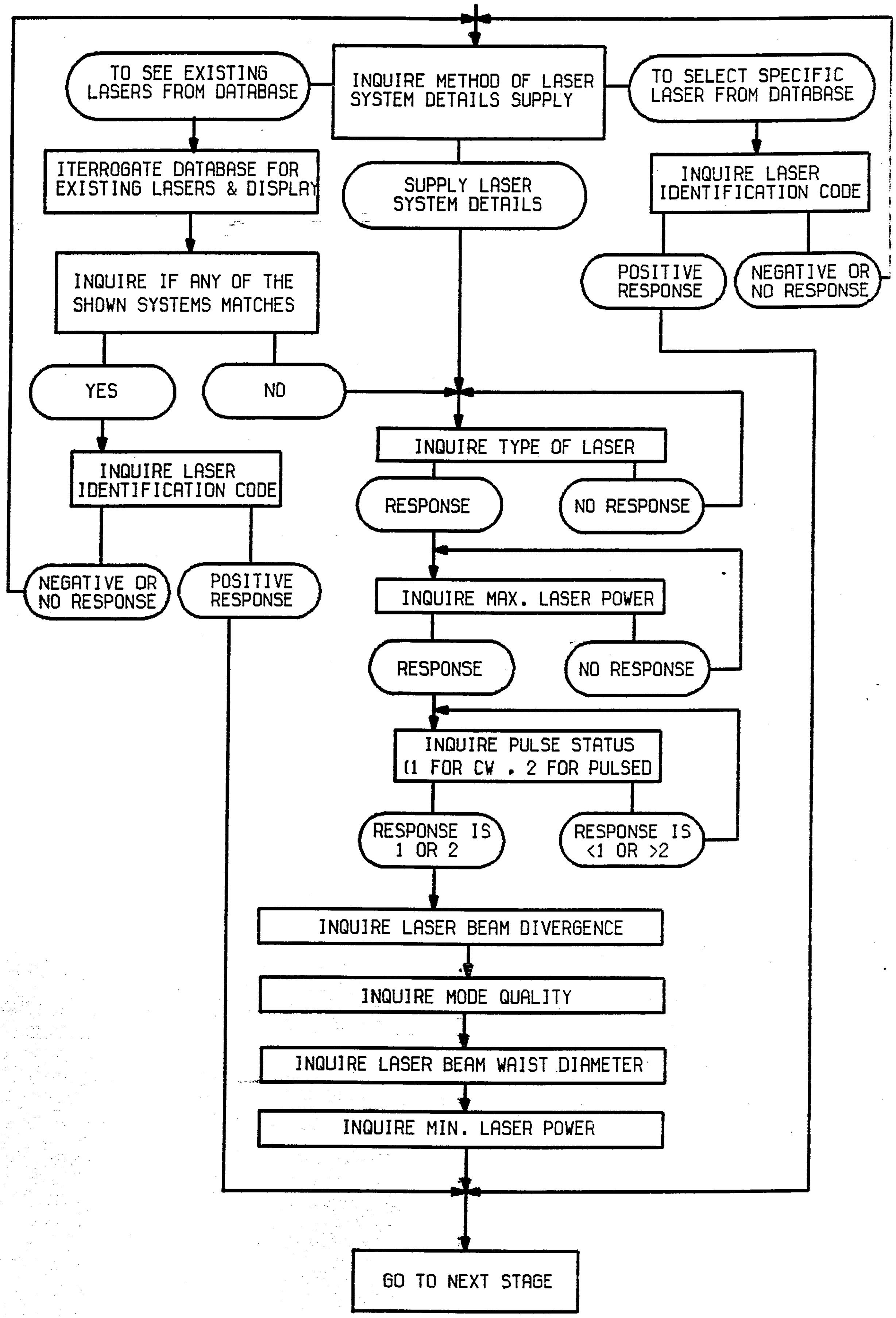


CHART 5.3 : INQUIRY PROCEDURE OF LASER SYSTEM DETAILS

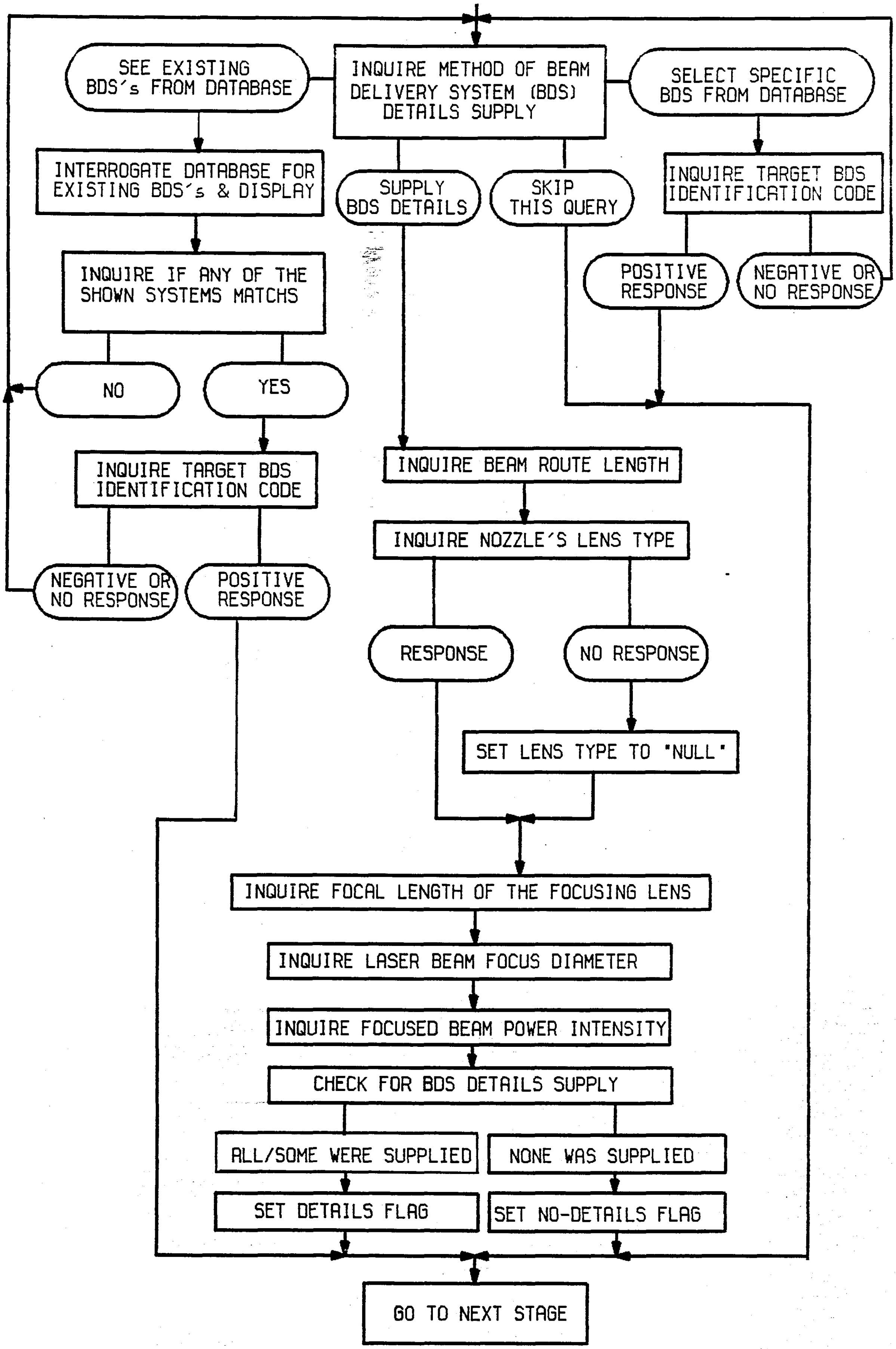


CHART 5.4 : INQUIRY PROCEDURE OF BEAM DELIVERY SYSTEM DETAILS

presented in chart 5.4, and is similar to that of laser system details.

d)INQUIRY OF PREFERENCE OF CUT QUALITY FEATURES: This inquires from the user information as to the importance to be placed in optimising each of the five cut quality features (kerf width, HAZ, cut roughness, dross formation and taper), and to the importance of attaining maximum cutting speed (chart 5.5). These so called preference weights are used by the prediction program to determine suitable laser processing parameters, by compromising between the six preferences above. The use of these weights is discussed in more detail in chapter 6.

The preference weights represent relative weights, and not absolute values, thus each of them can be any positive integer, regardless of their total sum. However, it is suggested to the user that this integer should lie between 0 and 99, only as a matter of formality and common sense.

There is no insistence on supplying any of the data values. When the user skips this procedure, or does not propose any preference, this is taken as an implicit indication of not being interested in quality. Consequently, economy is considered to be the default goal, and a weight of 99 is assigned to feed preference, and zero to all of the others.

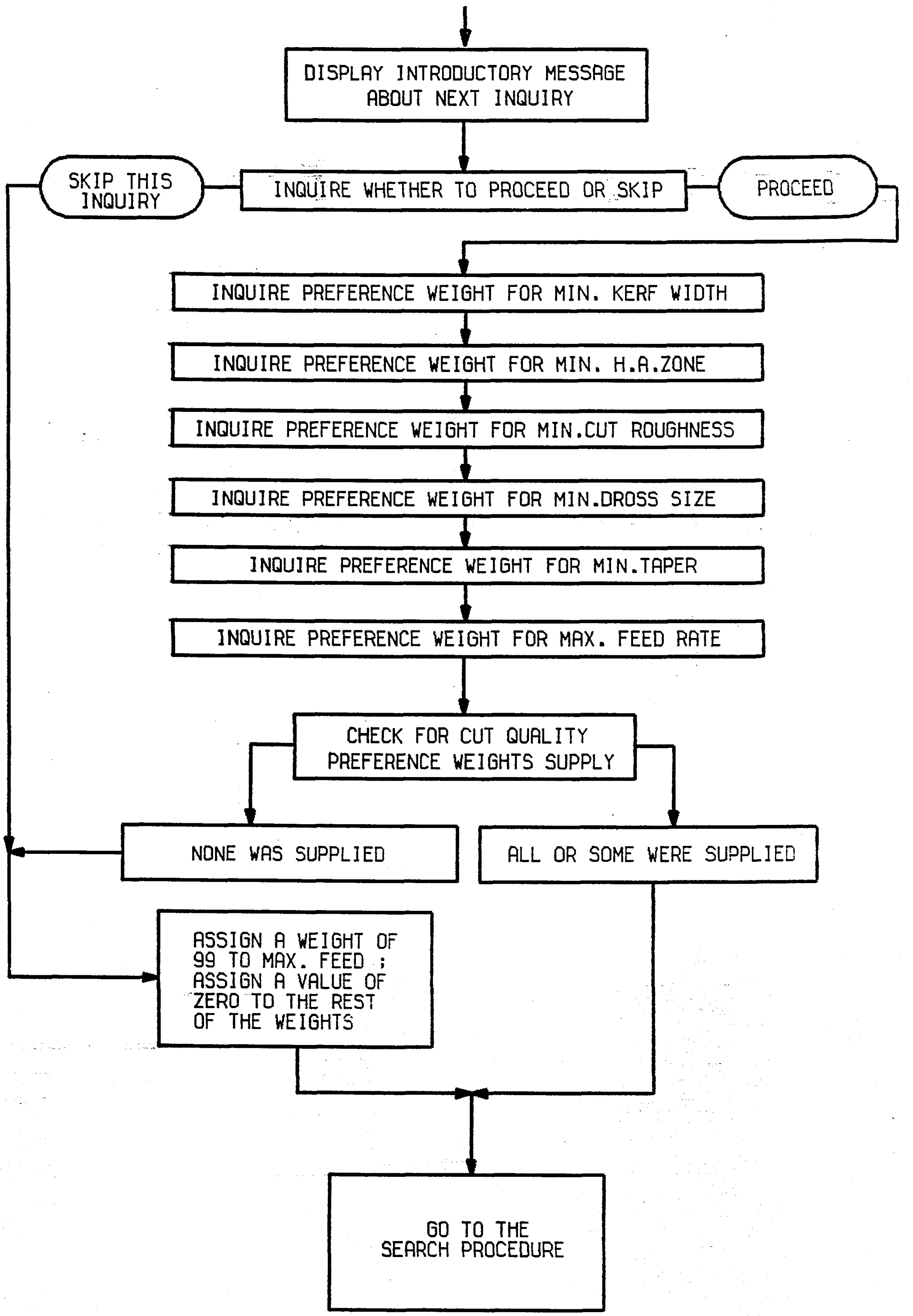


CHART 5.5 : INQUIRY PROCEDURE OF THE PREFERENCE WEIGHTS

5.3.2 THE DATABASE SEARCH PROCEDURE

The database search submodule is responsible for searching the tables of the database for relevant data. The search is made in a multi-stage fashion and directed according to user requirements.

The search unit is a system of rules that examines the implications of the user constraints, and based on these creates and executes a search operation. If this operation fails due to insufficient data being found, a user constraint is relaxed and an alternative search operation is executed. These steps are based on the arguement that search failure was not caused by total absence of data, but due to the constraints exerted. This method of searching the database is detailed in the next section.

5.3.2.1 THE SEARCH STRATEGY

The working principle of the search strategy is shown schematically in chart 5.6, and is based upon the following principles:

1) Compilation of the search constraints in terms of their relative importance. This is done on two phases. In the first phase, the constraints are categorised as:

- Category 1: Constraints that cannot be dropped or relaxed.

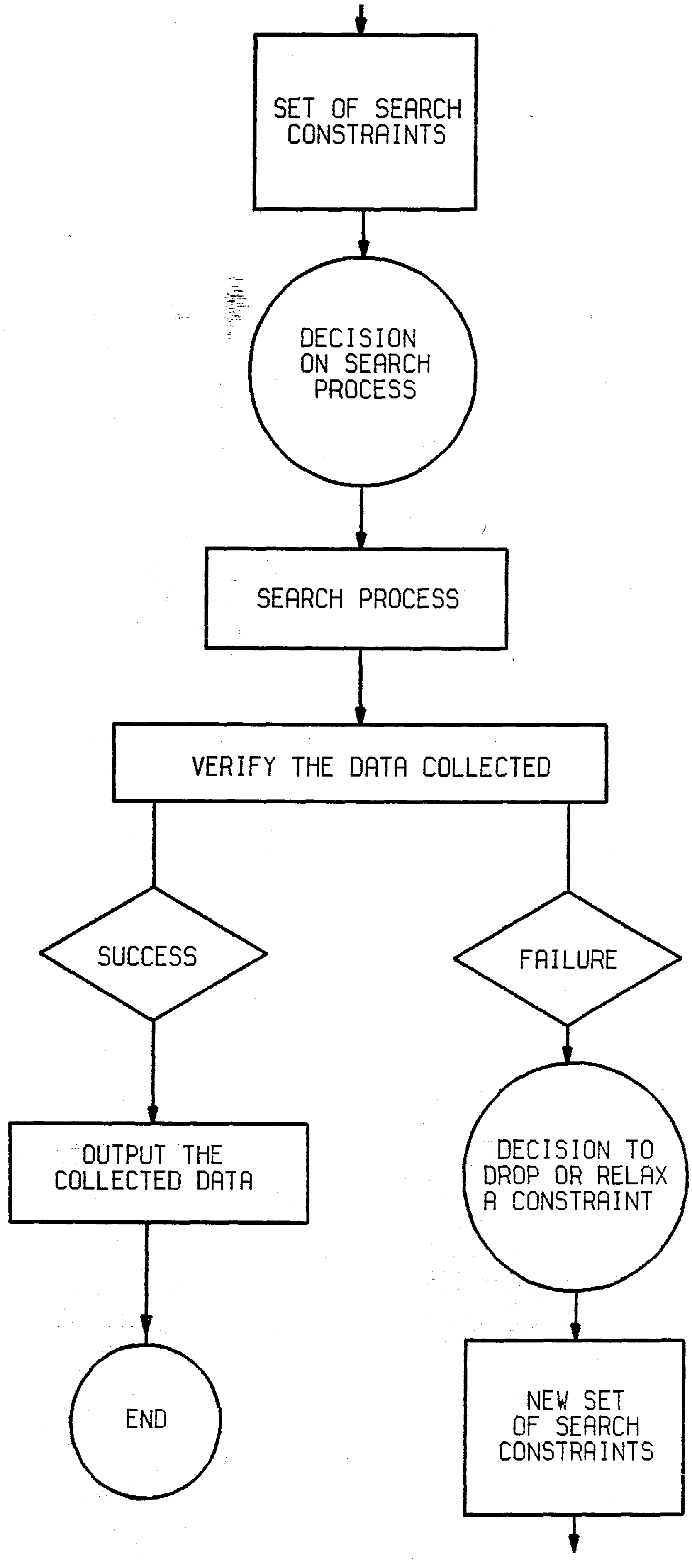


CHART 5.6 : THE WORKING PRINCIPLE OF THE SEARCH STRATEGY

- Category 2: Constraints that can be relaxed (to a limit).
- Category 3: Constraints that can be dropped.

In the second phase, the relative priorities are assigned to the constraints of the second and third categories above.

2) Identification of the initial set of user imposed constraints.

3) identification of an initial search operation according to the initial set of constraints.

4) If the search operation fails, then the constraint that has the least relative priority among the current set is dropped, and a new search is created in terms of the remaining constraints.

5) Repeat the procedure of 4 after every search failure, to a predefined limit, where no further concession can be given. If no (adequate) data can be found, the search procedure must be abandoned, and a search failure reported.

The classification of the search constraints (item 1) is of distinct importance, as the deduction or relaxation of these constraints needs to be performed systematically in the scope of the priorities given by this classification.

5.3.2.2 CLASSIFICATION OF THE SEARCH CONSTRAINTS

The search constraints were classified in terms of the compilation categories (section 5.3.2.1) as below:

CONSTRAINT	CATEGORY
Material class	1
Material type	3
Material thickness	2
Laser type	2
Laser pulsing	1
Beam delivery system	3

The material class (group) is a category 1, as materials are classified according to the relative similarity of their cutting characteristics implied by their chemical and physical properties (chapter 1 and chapter 3). Accordingly, the search procedure begins and proceeds to the end, only in term of one class of material as identified by the user.

The specific material type, when identified as matching one of those types that already exist in the database, will not be dropped unless all other constraints have been dropped or relaxed. The importance of this constraint is appreciated, due to the belief that different material types within a class may have widely differing cutting characteristics. On the other hand, this phenomena will become less important as data becomes more substantial, where it can be compiled into more material tables, with fewer and closer types in each table.

Material thickness has a comparable importance to material type. The thickness constraint is maintained as far as type is maintained, and relaxed after ensuring that no data could be found for the target thickness. This is relaxed by searching in terms of a range of thickness instead of a single thickness. This range is expressed as a specific percentage above and below the target thickness.

The laser system, specified by its identification code (CL), is relaxed by making the search to be performed in term of its pulse status (i.e whether it is pulsed or CW). The pulse status is an important characteristic, due to the difference in cutting mechanism and performance between these two types of laser, as highlighted in chapters 1 and 4. Therefore, laser pulse type is maintained up to the final step in the search procedure.

The constraint of beam delivery system is dropped when it is recognized that no data could be found for that specific system.

5.3.2.3 THE SEARCH ALGORITHM

An eleven stage search algorithm was developed in accordance with the principles of the search strategy. A schematic layout of the this algorithm is drawn in parts (A) and (B) of chart 5.7. Part (B) is a continuation to "A", and contains five search stages which complement the six stages of part (A).

In chart 5.7, "SPECIFIC LASER" denotes a specific laser system that already exists in the database, and is represented by its identification code. While "LASER TYPE" denotes whether the laser is pulsed or CW (i.e pulsation status). The term "SPECIFIC BDS" in this chart, denotes specific beam delivery system represented by its identification code. Additionally, when the user identifies a specific laser by its identification code, the pulsation status of that laser is retrieved automatically from the database, by the program and retained for later use. This is needed for two reasons: The first is when "SPECIFIC LASER" constraint is relaxed to "LASER TYPE". The second reason is that it must be output to the output data file as it is needed by the prediction module.

Chart 5.7(A) indicates that the first process is to identify the material class, in order to specify the material table to be searched. Then, the initial set of user constraints is compiled into one of the six categories shown in the chart, and the first search is executed.

Stage 1 of the search procedure creates a search with a full set of constraints (the first set).

Stage 2 allows a search to be made without the beam delivery system constraint, or for that constraint to be dropped if the search is redirected upon failure at stage 1.

In stage 3 "SPECIFIC LASER" is relaxed to "LASER TYPE", while the last opportunity is given to "SPECIFIC BDS" constraint.

Stage 4 accepts the redirected search from stages 2 and 3, and the initial search from the fourth set.

In stage 5 one last opportunity is given to SPECIFIC LASER constraint. This stage performes the initial search directed from the fifth set, and the search redirected from stage 4.

Stage 6 is the last stage which performs an initially directed search. It is constrained by two basic constraints, i.e thickness and laser type, and also provides for any other combination of constraints not matching the other five sets. An example of that is when the user specifies the mandatory information only like thickness and laser pulsation status. This stage also accepts the search redirected from stages 4 and 5.

Stage 7 provides the last opportunity for material type constraint to be imposed, while the thickness constraint is

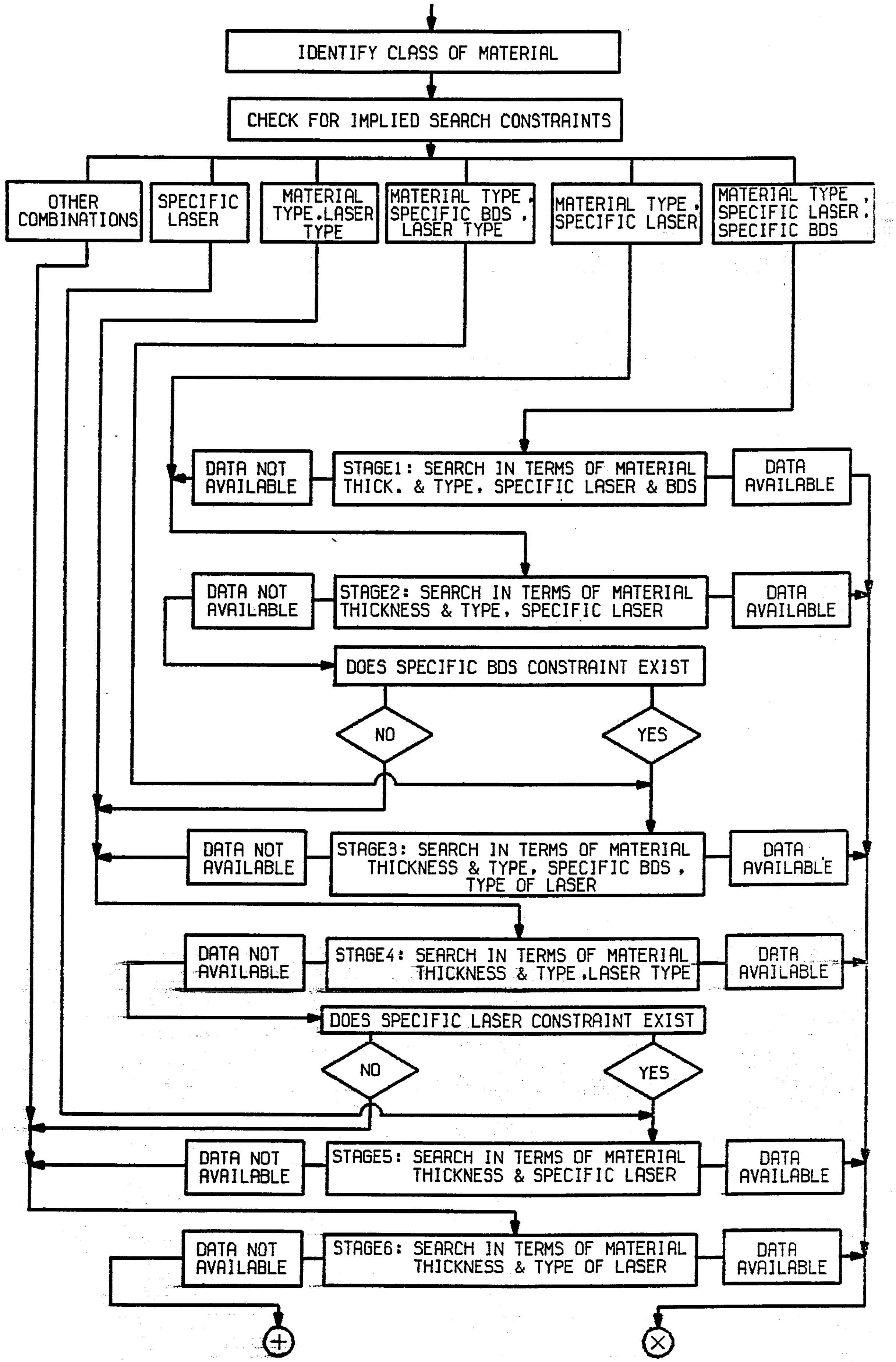


CHART 5.7 (A) : THE DATABASE SEARCH PROCEDURE

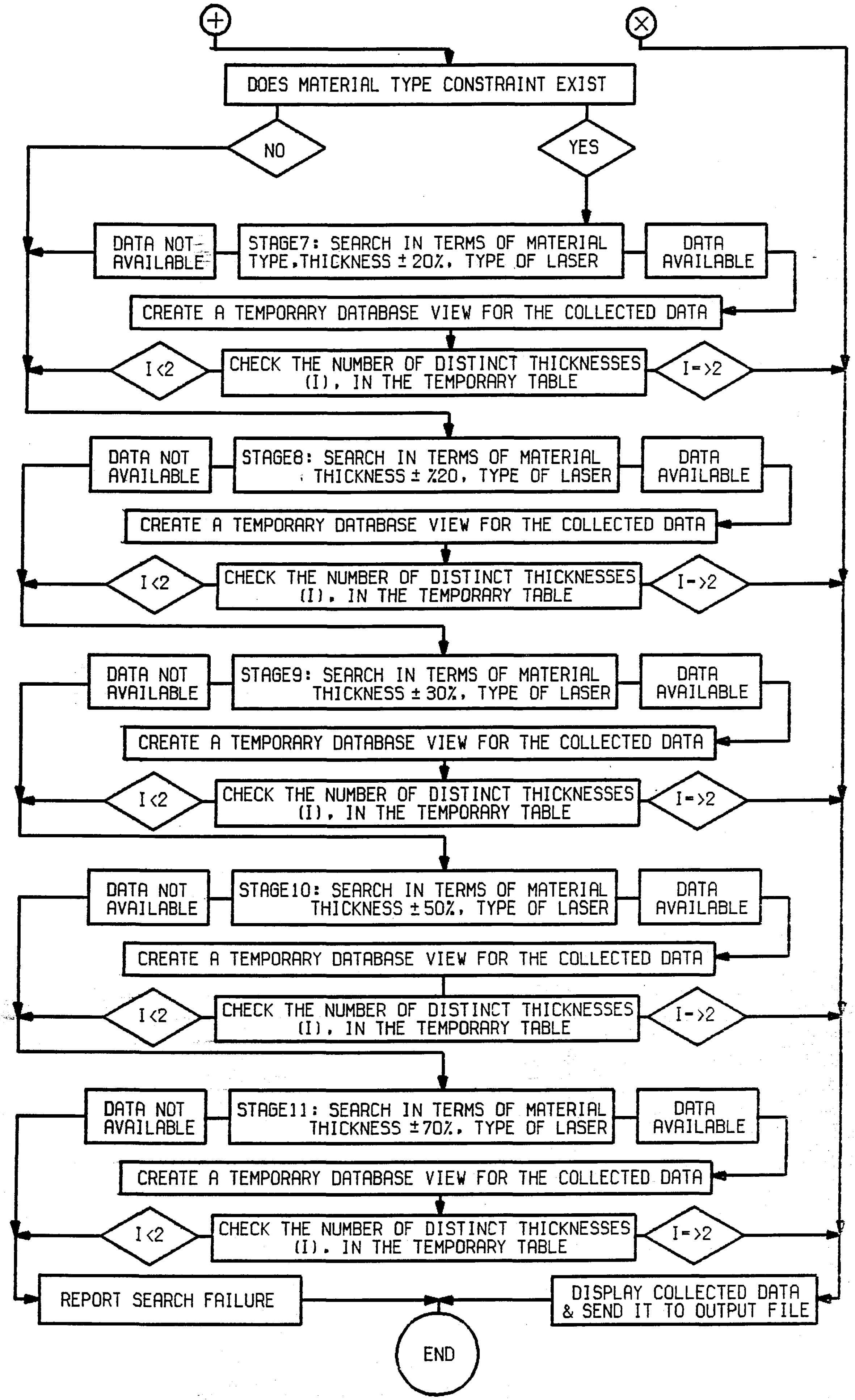


CHART 5.7 (B) : THE DATADASE SEARCH PROCEDURE (CONTINUED)

relaxed by 40% of the target thickness (±20%).

stage 8 and the subsequent stages are all alike except in broadening the thickness range after every stage, up to a predefined limit.

Stages 7 to 11 use a different method of searching which includes the creation of a temporary database VIEW of the collected data. This was necessary as the prediction program generates process parameters on the basis of interpolation or extrapolation of data of, necessarily, more than one thickness. Accordingly, the collected data must be checked for the existence of two or more thicknesses. The most efficient way to do this is to use features available within the SQL/RT RDBMS, rather than imposing additional external program codes and processing. This is achieved by gathering the data into a view, which is then searched for distinct thickness values.

This method is considerably faster than direct searching (checking) of the original database tables, as a two search operation would be required (one for check only and one for collection).

5.4 IMPLEMENTATION OF THE DATA COLLECTION MODULE

The data collection program was implemented as a computer program written in the "C" programming language as this had the necessary interface to the SQL/RT RDBMS. Specially formulated SQL statements were embedded within the program host language, to interact with the SQL/RT database system. This program is called a "C Precompiler program". APPENDIX 5A is a listing of the developed pc program. This program has to be compiled in a two stage process before it becomes executable. The first stage is a pre-compilation process which expands the embeded statements into "C" code, for internal communication with the database management system. The second process is a normal compilation of the "C" code into an executable module, that runs under the AIX operating system of the IBM 6150 computer system.

The previously mentioned special statements are of two types: The first type is ordinary SQL statements from the SQL data sublanguage. The second type is called Precompiler statements, they establish and organise communication channels with the SQL/RT database system. Appendix 5B lists these two types with a brief description of each statement.

The formulation of the "C" program to interface with SQL/RT database had some implications on the database and on the program itself, these are detailed in Appendix 5C.

The database search processes were based on the use of "Dynamically Defined SQL Statements", this technique is detailed in Appendix 5D.

5.5 THE COLLECTION MODULE OUTPUT DATA

The output data is formulated in a data file that satisfies all the data requirements of the prediction module. It is also, organised and commented in a way well enough to be man-readable and understandable. Fig 5.1 reveals two different output samples. Fig 5.1(A) represents output from one of the first six search stages, i.e for specific thickness. while the data of fig 5.1(B) is from one of the last five stages, i.e thickness range.

The output data falls into three categories:

1) Information data: It includes the first two line of the file, which define the material class and type.

2) Control data: This data is used to guide the processes of the prediction program. It occupies the next six lines of the file, and as follows:

- The target material thickness , in microns.

- Two codes for thickness range indication. The first can be 0 or 1, to indicate whether the collected data is for one thickness, or for a range of thickness, respectively. The second is the number of distinct thickness values.

- The maximum achievable laser power and cutting speed of the user cutting system.

- A code indicating fast feed dominance. It can be either 1 or -1 , to indicate whether preference weights of cut quality features have been assigned or not, respectively.

- The preference weights of fast feed and of the five cut quality features, as in the following sequence: feed, kerf, roughness, dross, taper and HAZ.

- The type of laser, i.e PULSED or CW.

3) Operational data: This is the data collected by one of the stages of database search. Fig 5.1(B) shows an extra, five column table of data. This contains the distinct values of the collected thickness range, and the maximum values of feed, power, PRF and pulse energy, that was found for each distinct thickness. The main table contains all the data, against the parameters that appear at the heading of the table. The two tables were organised in a fixed format, which allows data for both types of lasers, i.e pulsed and CW, to be included.

MILD STEEL	Material class
BS970EN3B	Material type
1600	Thickness
0 1	Codes of thickness range
60 8000	Max.power & max.feed
1	Fast feed code
50 40 00 00 99 30	Preference weights
PULSED	pulse status

thick	power	feed	gaspr	stdof	prf	plswd	plsen	kerf	haz	taper	rfnes	drsiz
1600	48	150	1000	500	15	1000	3200	470	55	20	5000	80
1600	48	175	1000	500	15	1000	3200	480	50	20	5500	80
1600	48	200	1000	500	15	1000	3200	450	58	20	5750	70
1600	48	225	1000	500	15	1000	3200	430	50	20	10000	60
1600	48	150	2000	500	15	1000	3200	420	50	20	5000	50
1600	48	175	2000	500	15	1000	3200	460	47	10	5500	80
1600	48	200	2000	500	15	1000	3200	420	35	20	8500	100
1600	48	225	2000	500	15	1000	3200	450	47	20	10000	25
1600	48	150	3000	500	15	1000	3200	440	40	20	3500	25
1600	48	175	3000	500	15	1000	3200	460	43	20	5750	25
1600	48	200	3000	500	15	1000	3200	410	45	20	8000	25

A : SINGLE THICKNESS AS THE TARGET THICKNESS

MILD STEEL	Material class
BS970EN3B	Material type
1250	Thickness
1 3	Codes of thickness range
210 12000	Max.power & max.feed
1	Fast feed code
30 20 80 0 0 50	Preference weights
CW	Pulse status

thick	feed	power	prf	enrgy
800	2250	200	0	0
1600	1250	200	0	0
3160	250	210	0	0

thick	power	feed	gaspr	stdof	prf	plswd	plsen	kerf	haz	taper	rfnes	drsiz
800	200	750	1000	1000	0	0	0	400	54	20	4250	40
800	200	1500	1000	1000	0	0	0	365	85	25	1500	70
800	200	2250	1000	1000	0	0	0	300	76	1	1000	30
1600	120	150	3000	1000	0	0	0	300	58	30	2250	10
1600	120	225	3000	1000	0	0	0	300	66	5	2000	10
1600	120	300	3000	1000	0	0	0	320	74	15	2000	10
3160	210	120	1000	1000	0	0	0	370	90	-80	3500	20
3160	210	180	1000	1000	0	0	0	380	110	-100	2500	40
3160	210	250	1000	1000	0	0	0	385	116	-100	3000	30
3160	210	60	2000	1000	0	0	0	340	62	20	4000	10
3160	210	120	2000	1000	0	0	0	330	90	15	2500	10
3160	210	180	2000	1000	0	0	0	350	78	20	3500	10
3160	210	250	2000	1000	0	0	0	330	97	25	4000	10
3160	210	60	3000	1000	0	0	0	340	74	45	5000	10
3160	210	120	3000	1000	0	0	0	330	85	30	3500	20
3160	210	180	3000	1000	0	0	0	335	78	20	5000	20
3160	210	250	3000	1000	0	0	0	340	93	30	4000	20
3160	210	60	4000	1000	0	0	0	320	60	45	5500	10

B : RANGE OF THICKNESS

FIG 5.1 : THE COLLECTION PROGRAM OUTPUT DATA FILE

CHAPTER 6

THE PREDICTION MODULE

The prediction module forms the second part of the planning system, and complements the data collection module. The module predicts a set of process parameters suitable for a prospective laser cutting application, using data supplied by the collection module.

This approach uses a generalised set of rules for the qualitative and quantitative analysis of supplied data. Consequently, the analysis is highly influenced by the integrity of the supplied data.

This chapter adresses: the main requirements of the module, analysis of the prediction strategy, the mathematical methods used in the analysis, and the method of implementation.

6.1 OBJECTIVE AND TASKS

The objective of the prediction module can be stated as: the prediction of a set of laser cutting process parameters, that will perform a successful cut featuring the desired cut quality specified by the user, for a defined application.

This approximation is made, in mathematical terms, by interpolation or extrapolation between the various output parameters of the data supplied by the collection module, to provide the required process input parameters.

The second task of this module is the graphical presentation of the predicted data together with original data points, so that the user can see the distribution of the collected data, the condition of interpolation and the effect of the various parameters on each other.

This module will also be used to formulate a matrix (or matrices) of control data to be used in real-time. The implementation of this task was allocated to later stage of development.

6.2 REQUIREMENTS AND IMPLICATIONS

The characteristics required by the data collection module, were also required by this module (see section 5.2). Flexibility and expandability were achieved by program modularity. The program is designed so that no fitting (interpolation) operation takes place before the data is tested as to its suitability.

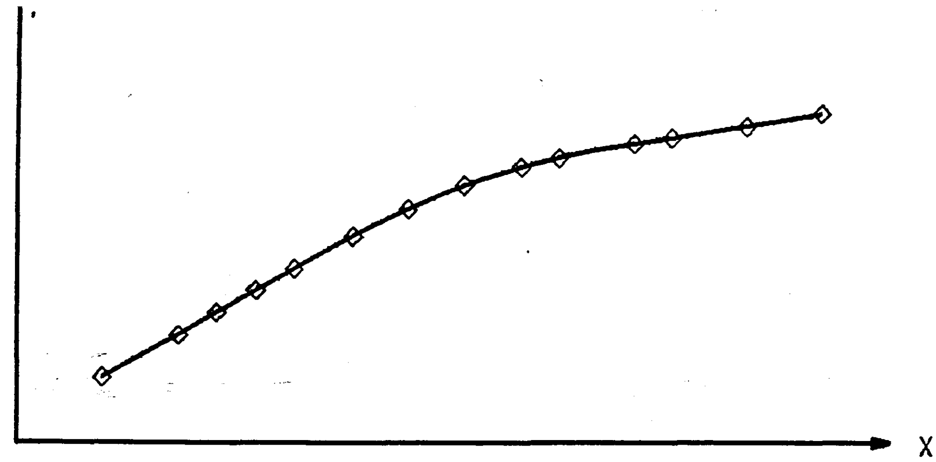
Program operation is influenced by two factors: the first is that the data provided may not be amenable to fitting.

The second is the stability and accuracy of the mathematical method of approximation.

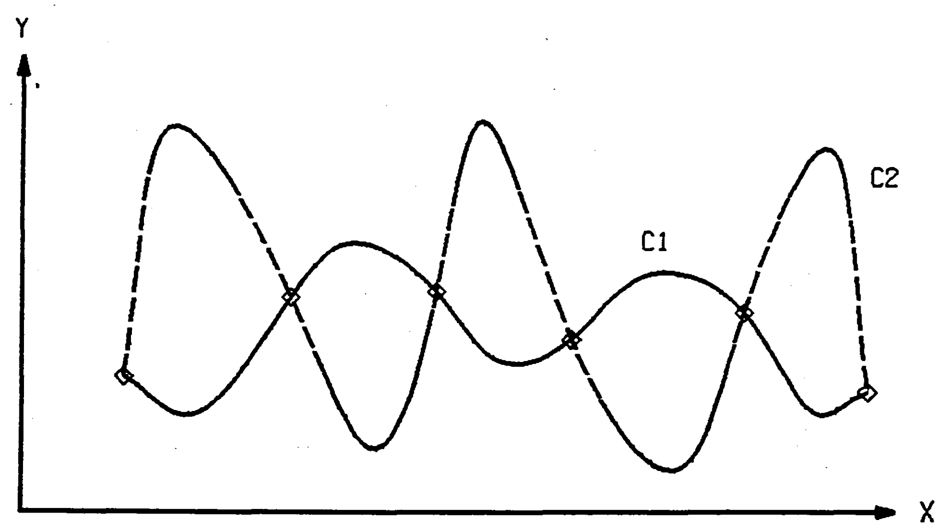
Fig 6.1 helps to clarify the influence of these two factors on the process of interpolation. Graph A shows an ideal set of data for exact interpolation, where the points are nearly continuous and show little scatter. The two curves in graph B can also be considered as exact interpolations as they pass through all points (ref 6.1). However, they exhibit different degrees of stability. This is caused by the interpolation method used and the wide separation interval between the points. Graph C shows that by using approximate rather than exact interpolation, a more average fit can be achieved. This type of interpolation is called curve fitting or smoothing (ref 6.2).

The three examples of fig 6.1 are based on the assumption that the data points interpolated occur consecutively, i.e each of the points has a distinct value on the independent variable (abscissa). The prediction program must be able to manipulate arbitrary data as shown in fig 6.2. These samples show that approximate interpolation can be found to the points in graphs A and B, but no sensible solution can be found for C. This phonemena highlights the importance of the selection of reliable approximation methods.

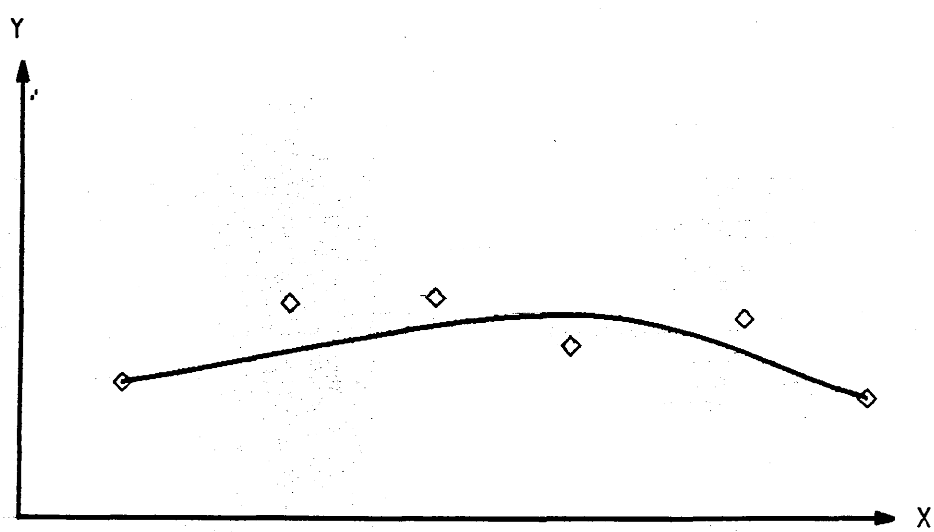
A requirement emerges at this stage, that is the need to inform the user of the reliability of the method used, the



GRAPH A : EXACT INTERPOLATION



GRAPH B : EXACT INTERPOLATION



GRAPH C : APPROXIMATE INTERPOLATION

FIG 6.1 : EFFECTS OF DATA POINTS DISTRIBUTION AND INTERPOLATION METHOD

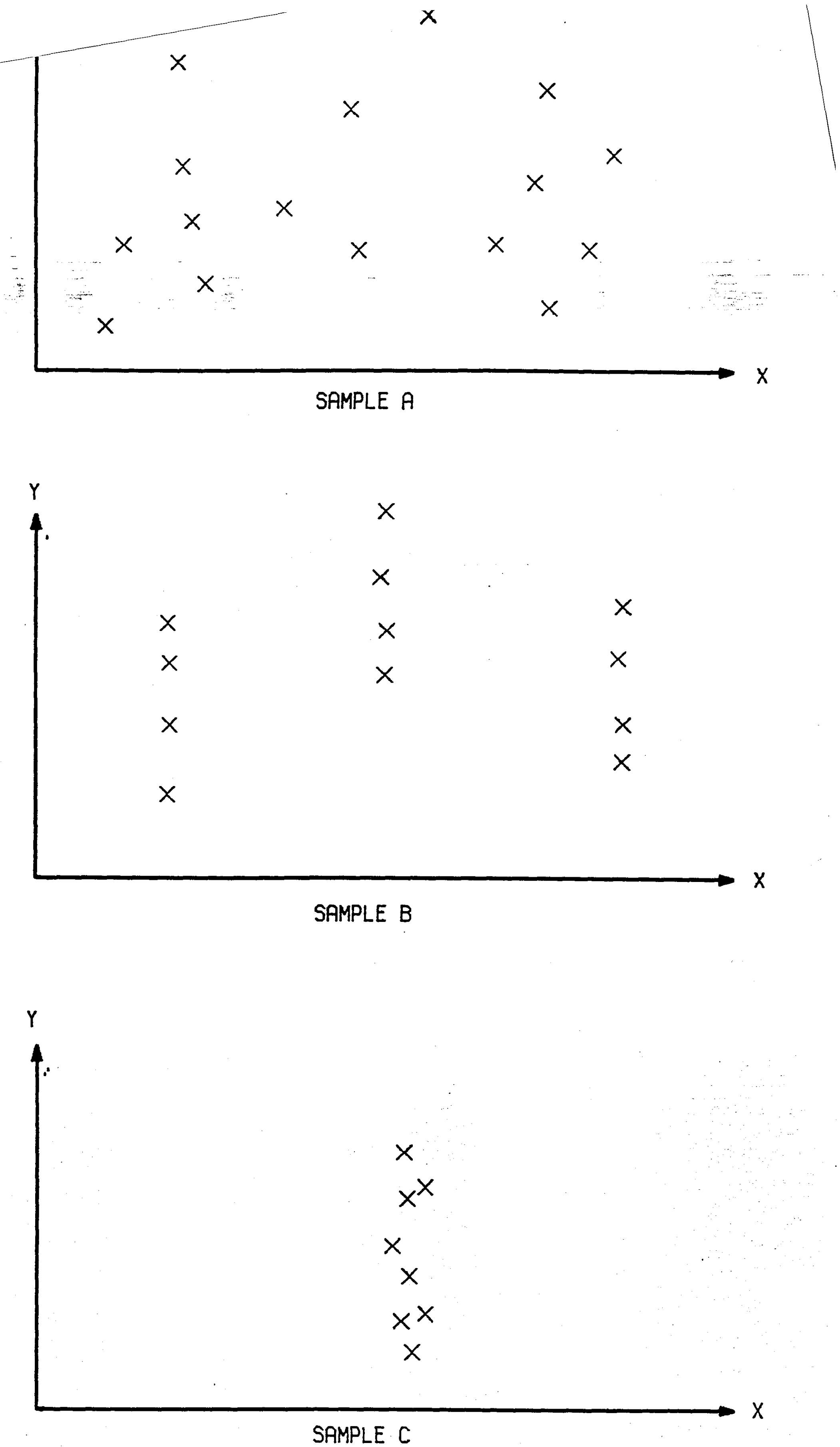


FIG 6.2 : THE ARBITRARINESS OF LASER CUTTING DATA

implications of the choices and responses made at the interrogation stage and the relevance of the acquired data to his application. This was implemented by presenting the data acquired and its interpolation, graphically.

The accuracy of approximation and consequently the prediction, is influenced by the applicability of the supplied data. That is how close is the data acquired to the required application in terms of material type and thickness, laser and beam delivery system. This is obviously dependent on the availability of such data in the database system. The prediction is also influenced by the approximation (fitting) method utilised, as indicated previously.

6.3 THE PREDICTION STRATEGY

In order to identify suitable fitting methods, it is required to identify, first, the parameters (variables) involved in the fitting processes. Additionally, the influence of these parameters upon each other must be known in order that they may be classified as independent or dependent. This greatly influences whether the fitting is simple two dimensional (one independent and one dependent variable), or of higher order.

The present analyses were made according to the parameters studied in the experimental programme only (chapter 4).

Further development of these analyses to include more parameters, may be performed at a future date.

The mutual dependencies of the parameters involved in CW laser cutting are charted in fig 6.3, and those for pulsed laser cutting in fig 6.4. It should be noted that material type is execluded from these analyses, as it is already preselected during the collection stage, i.e all the collected data is only related to one specific material type (If this is not possible, the collector will select data from one class of materials which has similar characteristics).

Assuming constant configuration of beam delivery system, fig 6.3 indicates that laser power is determined by type and thickness of the cut material, and by the required feed. Feed rate can be determined in a similar way from material characteristics and laser power. Cut quality features are affected differently by changes in process parameters. Consequently, the following relations describe the interactions of fig 6.3:

power = f (thickness, feed) (1)

feed = f (thicknes, power) (2)

kerf width = f (power, feed, gas pressure) (3)

Cut roughness, dross size, taper and H.A.Z, can be expressed exactly as in relation (3).

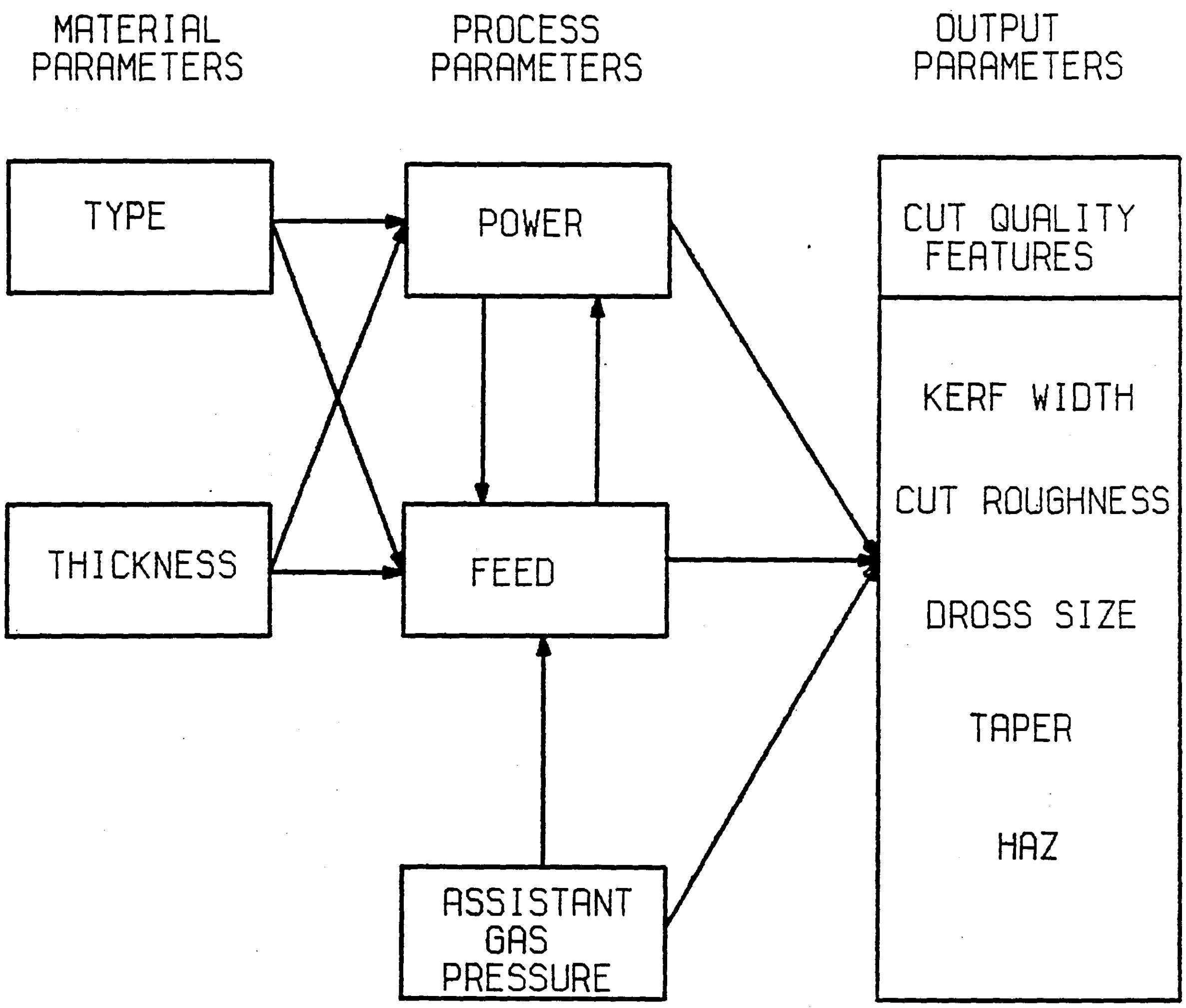


FIG 6.3 : THE DEPENDENCE BETWEEN CW LASER CUTTING PARAMETERS

When thickness is constant, i.e all the data are related to one thickness (target thickness), relations (1) and (2) can be reduced to:

power = f (feed) (4)

or in a reverse order, i.e [feed = f (power)].

Relations (1) and (2) indicate that a three dimensional function should be fitted when a thickness range is encountered.

Pulsed laser cutting is a completely different mechanism from that experienced with a CW laser, as shown previously. The energy required by each pulse is determined by the material properties and thickness only (fig 6.4). While the process of forming a continuous cut, is a function of feed and PRF, i.e higher feed can be achieved with higher PRF. The following relations describe these interactions:

pulse energy = f (thick) (5)

feed = f (PRF) (6)

kerf width = f (feed, pulse energy, PRF,
gas pressure, stand off) (7)

Relations for cut roughness, dross size, taper and H.A.Z can also be written as in (7).

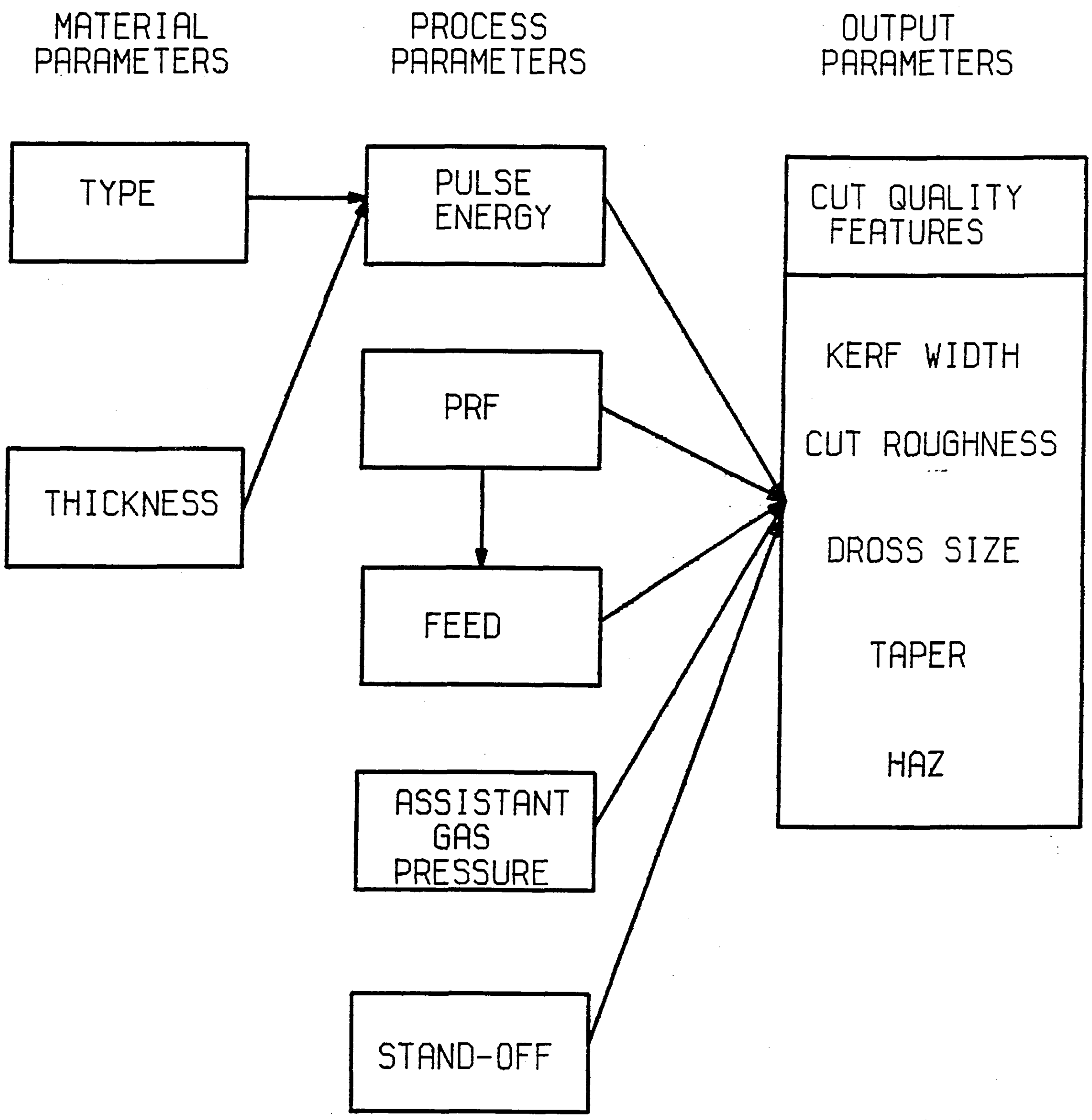


FIG 6.4 : THE DEPENDENCE BETWEEN PULSED LASER CUTTING PARAMETERS

Relation (5) is not required when the thickness is constant.

ANALYSIS OF CUT QUALITY

Relations (3) and (7) express the influence of process parameters on cut quality. However, it is believed that the configuration of these relations would not achieve the results required, because:

a) The task required is to determine the process parameters according to the preference of the various quality features, while the nature of these functions is to determine the cut quality features from a specified set of process parameters.

b) The way to find process parameters from these functions is by maximisation or minimisation of the fitted function (ref 6.3). However, this method produces multiple solutions as there is a separate relation for each of the cut quality features.

c) The function of several variables cannot be easily presented graphically, which violates the requirements of being user informative (section 6.2).

As a consequence of the above, it was decided to break down these relations (3 and 7) into two dimensional functions, as follows:

kerf width = f (feed) (8)

kerf width = f (power) (9)

kerf width = f (gas pressure) (10)

roughness = f (feed) (11)

roughness = f (power) (12)

roughness = f (gas pressure) (13)

The other quality features can be expressed in a similar way.

OUTLINE OF THE STRATEGY

The results of the experimental work showed that the most significant influence on quality is the feed rate. Therefore, feed will be considered as the key parameter for the determination of cut quality. It is believed that a reasonably accurate and practicable way of performing this task, is as follows:

- Evaluation of relations (8), (11) and subsequent relations which describe cut quality features in terms of feed rate.

- Finding feed rates that minimise each of the quality features, of the above evaluations.

- Using these feed values to calculate a normalised value of feed, taking into account the preference weights given by the user for each quality feature and for fast feed.

- Finding a power setting for CW laser, by using relation (4) for single thickness, or relation (2) if a thickness range is incurred.

- Finding a PRF setting for pulsed laser using relation (6), and using relation (5) to find a pulse energy if a thickness range is incurred.

It should be noted that the settings of power, PRF or pulse energy, computed by this procedure, represent the average requirements for successful cutting.

Relations describing the correlation between process parameters and cut quality features (e.g 9, 10, 12 and 13), should be evaluated as the results can be used in two ways:

- Graphical presentation to the user to help in perception of the effects of process parameters on cut quality, and in decision making.

- Generation of a real-time process control strategy (control matrix).

6.4 METHODS OF FITTING

This section explains the mathematical methods tested for feasibilty of application to the general fitting problems outlined in the previous section.

Suitable software for numerical analysis and fitting can be found within the "NAG FORTRAN LIBRARY". This software contains several types of subroutines for approximation by fitting, ref 6.3 documents this software. The use of this software provided several advantages, among these are:

1) Reduction in development time.
2) As a consequence of the above, it is possible to test a wider range of fitting methods.
3) Optimised performance, accurate and secure software.

The methods studied were:

- Interpolation in terms of one or two independent variables.
- Minimisation or maximisation of functions.
- Curve and surface fitting.
- Linear correlation and regression analysis in terms of one or several independent variables.

Two interpolation methods were tried and tested during the early stages of development of this module. These routines (E01AAF and E01ACF), were for one and two independent variables, respectively. Test data from literature was used to asses their suitability as no experimental data was available at the time. Both of these methods were later abandoned as they proved to be unsuitable when tested with real data (see section 6.2).

A method of function minimisation (E04JAF) was tried with a fitting function computed using linear correlation (G02BCF) and regression (G02CGF) for several independent variables. This was also abandoned as the results were highly inaccurate.

6.4.1 LINEAR CORRELATION AND REGRESSION ROUTINES

Linear correlation and regression analyses can be made for one or several independent variables. This method was pursued for two reason:

- This method provides correlation coefficients between the various variables. The correlation coefficient is a number between 1 and -1, which indicates the strength of correlation between two variables. Strong correlation is expected when this number approachs either 1 or -1, whilst a near zero value indicates poor correlation. It was, therefore, envisaged that this coefficient would provide a means of verifying the degree of dependence between the various parameters.

- Wide availability of well documented routines that analyse for one or several independent variables.

The routines available also provide for missing data. This is done by execluding incomplete variables sets from the analysis. This reduces the errors caused by missing data from the input data file (see fig 5.2).

The NAG routine G02CCF was used for linear correlation and regression, in terms of one independent variable. It fits a straight line of the form:

$$y = a + bx$$

to a set of data points. Where (y) and (x) are the dependent and the independent variables, respectively, (a) is the regression constant, and (b) is the regression coefficient. The fitting is based on minimising the sum of squares of the differences between the original data and the fitted data. This technique is called least-squares approximation (ref´s 6.1, 6.2 and 6.3).

Two routines G02BCF and G02CGF, were used successively for multiple linear correlation and regression, in terms of several independent variables. The latter routine fits an approximation, based on least-squares technique, of the form:

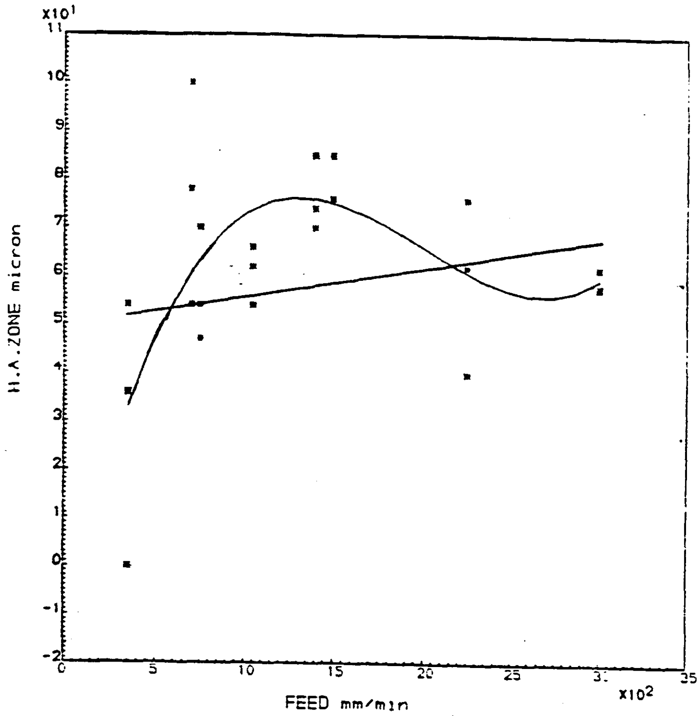
$$y = a + b_1 x_1 + b_2 x_2 + \dots\dots\dots + b_n x_n$$

The main drawback of linear approximation is that it can be very inaccurate. This is dependent mainly, on the form of the original data. The effects of data form are shown diagramatically in fig 6.5. Graph A of this figure shows that a linear fit fails to represent the profile of the original data points. While a curve fitted by a polynomial of degree 3 is more representative. Graph B shows that the higher order polynomial gives a similar result when the data is relatively linear.

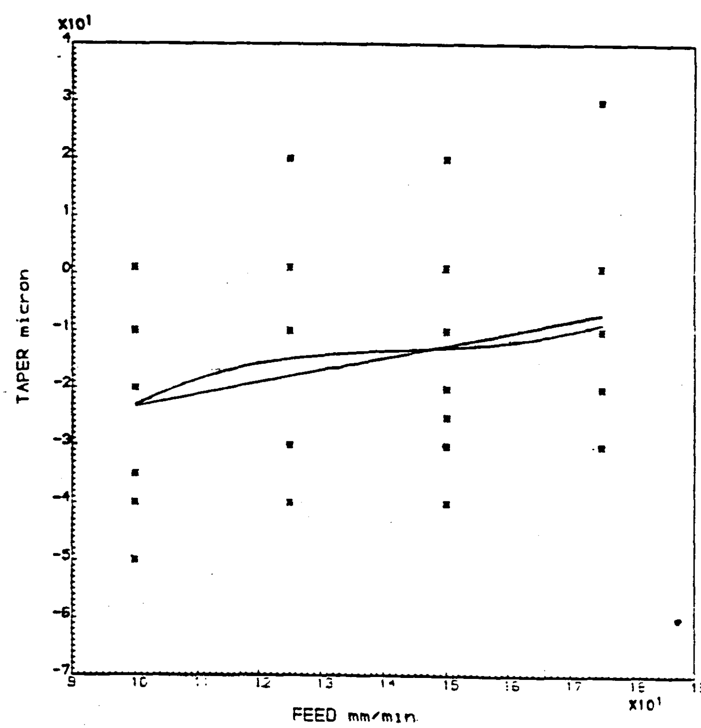
Linear approximation has the advantage of being both stable and extendable. The mathematical relation (y = a + bx) can be readily extended (extrapolated) outside the domain of the original data. Non-linear curves should generally not be extrapolated outwith the domain of original data, as the predicted data may be highly inaccurate (ref 6.2), fig 6.6 illustrates these features.

6.4.2 CURVE AND SURFACE FITTING ROUTINES

These routines deal with fitting functions for one and two independent variables respectively, and proved to be the most important for the implementation of the prediction module. A curve fitting technique was used for all the approximation operations, except for computing the three dimensional surface relation (2) of section 6.3.

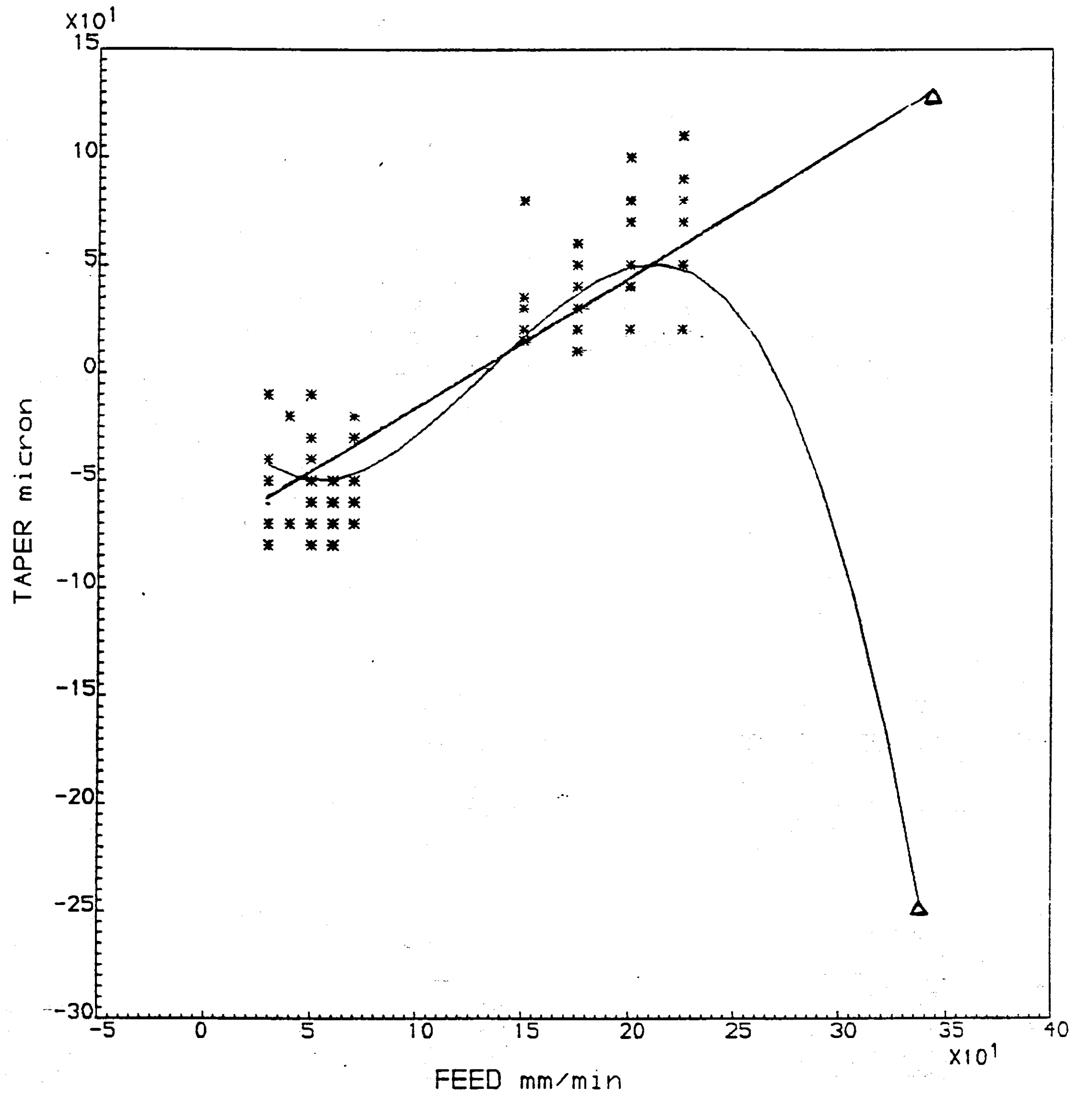

x10¹
H.A.ZONE micron
FEED mm/min
x10²

A


x10¹
TAPER micron
FEED mm/min
x10¹

B

FIG 6.5



TAPER Vs FEED FOR MILD-STEEL

FIG 6.6

CURVE FITTING

Two methods of curve fitting were tested for feasibility, cubic spline and single polynomial. These were selected for their ability to approximate to an arbitrary set of data, using a least-squares method, which was thought to be the best technique for experimental data (ref´s 6.2 and 6.3).

The first method of fitting was performed using the routine E02BAF which computes a weighted least-squares approximation by cubic spline (piecewise polynomial, ref 6.1). This works by fitting a series of polynomial splines of degree 3, for intervals defined by the user. Different weights can be given to data points, according to their level of importance, or confidence in each point. Subsequent evaluation of the function is performed by the routine E02BBF. This method is numerically stable, accurate and efficient (ref 6.4).

This method was rejected after many trials made with test and real data, due to the repeated occurance of an arithmetic overflow during the operation of this routine, which terminated the entire program. This was caused by a process which caused the division by a value of zero, arising from repeated values being encountered in one or both of the variables (a division by the zero difference between these data).

The second method of curve fitting used the NAG routine E02ADF. This routine computes a single, weighted least-squares polynomial approximation to an arbitrary set of data points. Subsequent evaluation of the fitted function is carried out using the routine E02AEF.

This method was used for all curve fitting within the prediction program, as it was found to be suitable for the current application. It proved to work with any set of data, and allowed the user to choose the degree of the polynomial to be used. This feature is very important as the stability and accuracy of the least-squares approximation is largely influenced by the polynomial degree (ref´s 6.1 and 6.2). The use of higher degree polynomial may result in better accuracy, i.e the curve passes near or through more original points. However, this may also generate higher fluctuation (see graphs B and C of fig 6.1). Consequently, it can be concluded that a low degree polynomial is preferable (for further details as to the effects of polynomial degree consult Appendix 6A).

SURFACE FITTING

The surface fitting method used is a weighted least-squares bicubic (double cubic, ref 6.5) spline approximation to a three dimensional set of data (NAG routine E02DAF). This works by dividing the data region of the two independent variables into several rectangular panels. A bicubic polynomial is fitted to each panel, and is joined with the splines of the adjacent panels. Subsequent evaluation of the fitted function is performed by the NAG routine E02DBF.

This method is described as being numerically stable (ref´s 6.3 and 6.5), but a fault similar to that detailed previously, was also experienced here. This technique also suffered from instability and inaccurate fitting. This was due to the nature of the data, which can be considered as being ill-conditioned for this type of processing. This is mainly due to the segregation of the data points in some regions, and discontinuity between regions.

Despite these difficulties, a novel three dimensional fitting method was created for evaluating the relation of: feed, power and thickness. This required re-structuring of the "raw" data of the three parameters above. This is explained in the next section.

6.5 IMPLEMENTATION OF THE PREDICTION MODULE

The prediction module was implemented as a FORTRAN 77 computer program on a VAX 11/750 computer system. This implementation was based on the principles of the prediction strategy explained previously. The outline of the prediction procedure is illustrated in chart 6.1. It should be noted that the term "CORRELATE" used in this and

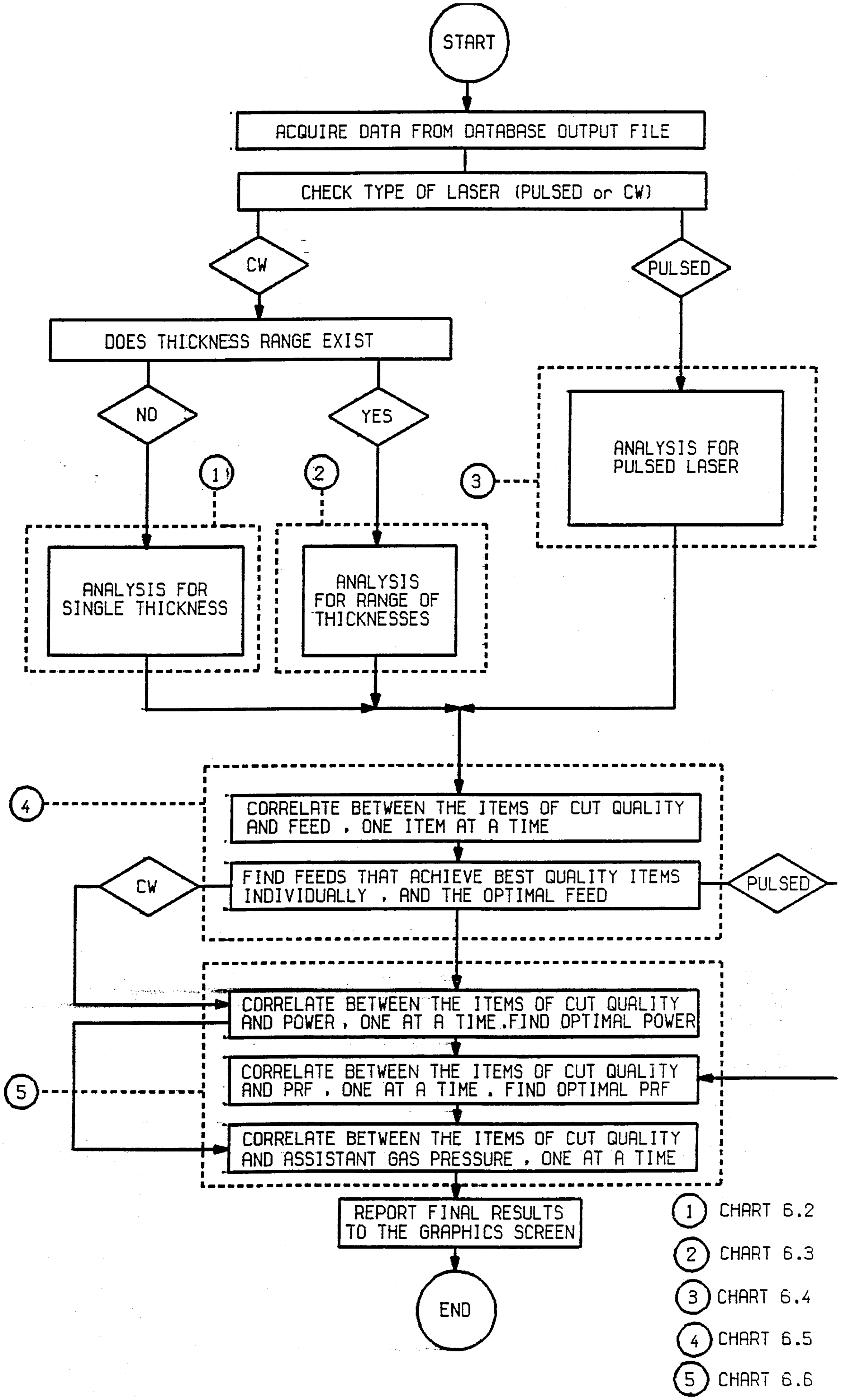


CHART 6.1 : OVERVIEW OF THE PREDICTION PROCESSES

the subsequent charts, means evaluation by curve or surface fitting.

The first step in the prediction procedure, is data acquisition from the input data file, which is output from the collection module. Subsequent processing is performed according to type of laser, i.e CW or pulsed. The fundamental process parameters are found according to the principles of the prediction strategy. The results of the correlation processes are shown graphically. Finally, a summary of important results and information is displayed, while data is directed to output files.

A listing of the prediction program is given in Appendix 6B. The main program routine is responsible for all input and output processes, and for directing strategic activities. The subroutines PROCS1 and PROCS2, perform the detailed activities of curve and surface fitting. Graphical output is performed by the subroutines GRFSTT, GRAF1, GRAF2 and GRAF3. These graphs are either displayed on the console during the program run, or sent to a plot file for hard copy. The rest of the subroutines perform auxiliary functions.

The analyses of the prediction program are discussed under two categories: Analysis for CW laser, and analysis for pulsed laser. These analyses are detailed in the following sub-sections.

6.5.1 ANALYSIS FOR CW LASER

The correlation procedures between the process parameters of CW lasers are discussed in this section. They consist of two major procedures: One for the case of single material thickness, and one for situations where a range of thicknesses exists in the input data. The remaining procedures are covered in section 6.5.3 and the subsequent sections, as they are applied to both CW and pulsed lasers.

ANALYSIS FOR SINGLE THICKNESS

Chart 6.2 illustrates this procedure. The main procedure of this analysis, is the correlation between laser power and feed, by curve fitting. Fitted data are stored in a data array for further use. The fitted curve and the original data are plotted on one graph, which is output for inspection. Fig 6.7 is a sample of this type of graph.

The selection of feed as the independent variable was due to the expectation of a more uniform distribution of data points. This is due to the more normal practice of cutting with one power setting at different feeds.

Before each correlation, the parameter being correlated is examined to determine if it is constant throughout the range of input data. In that case, the correlation is

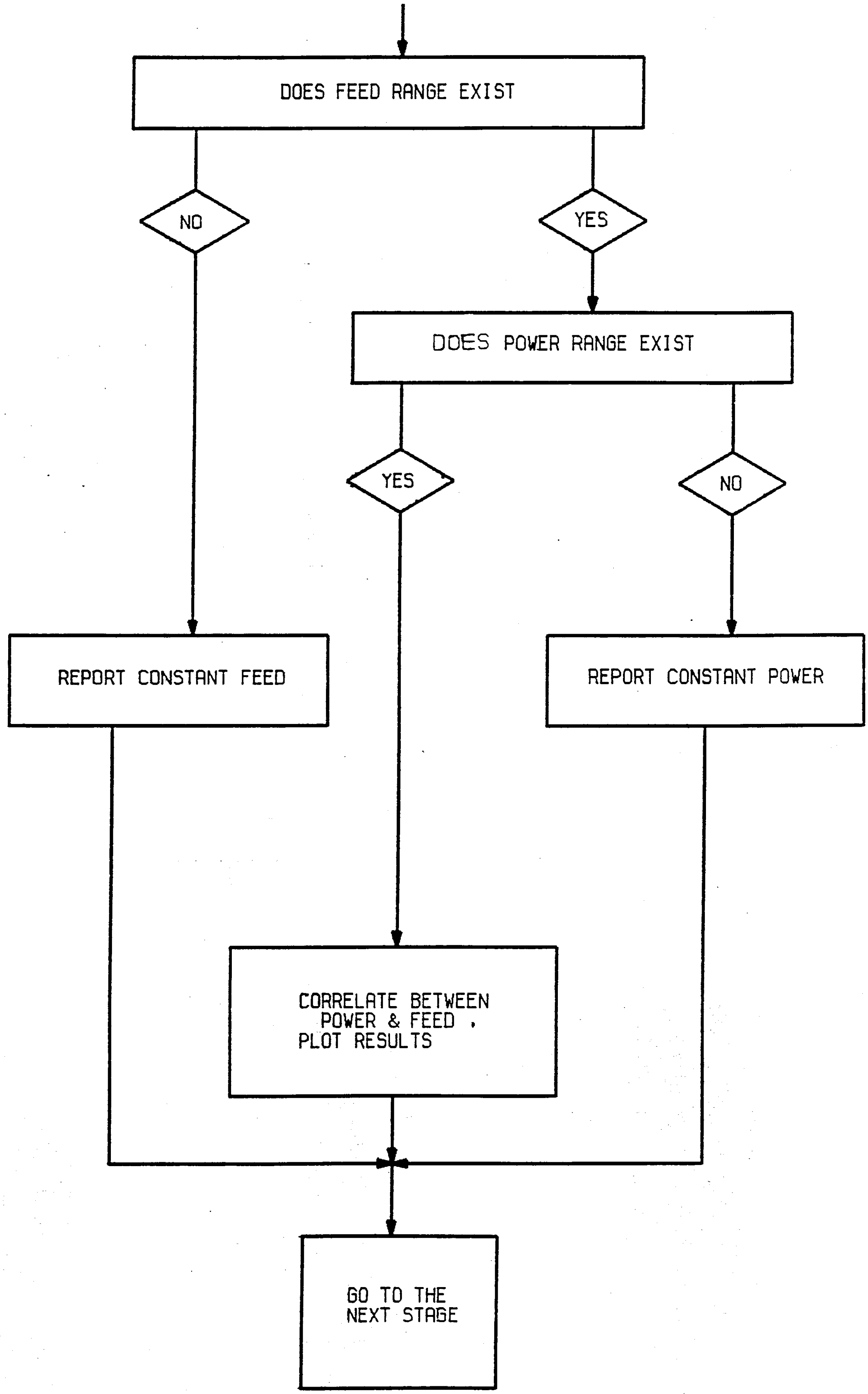
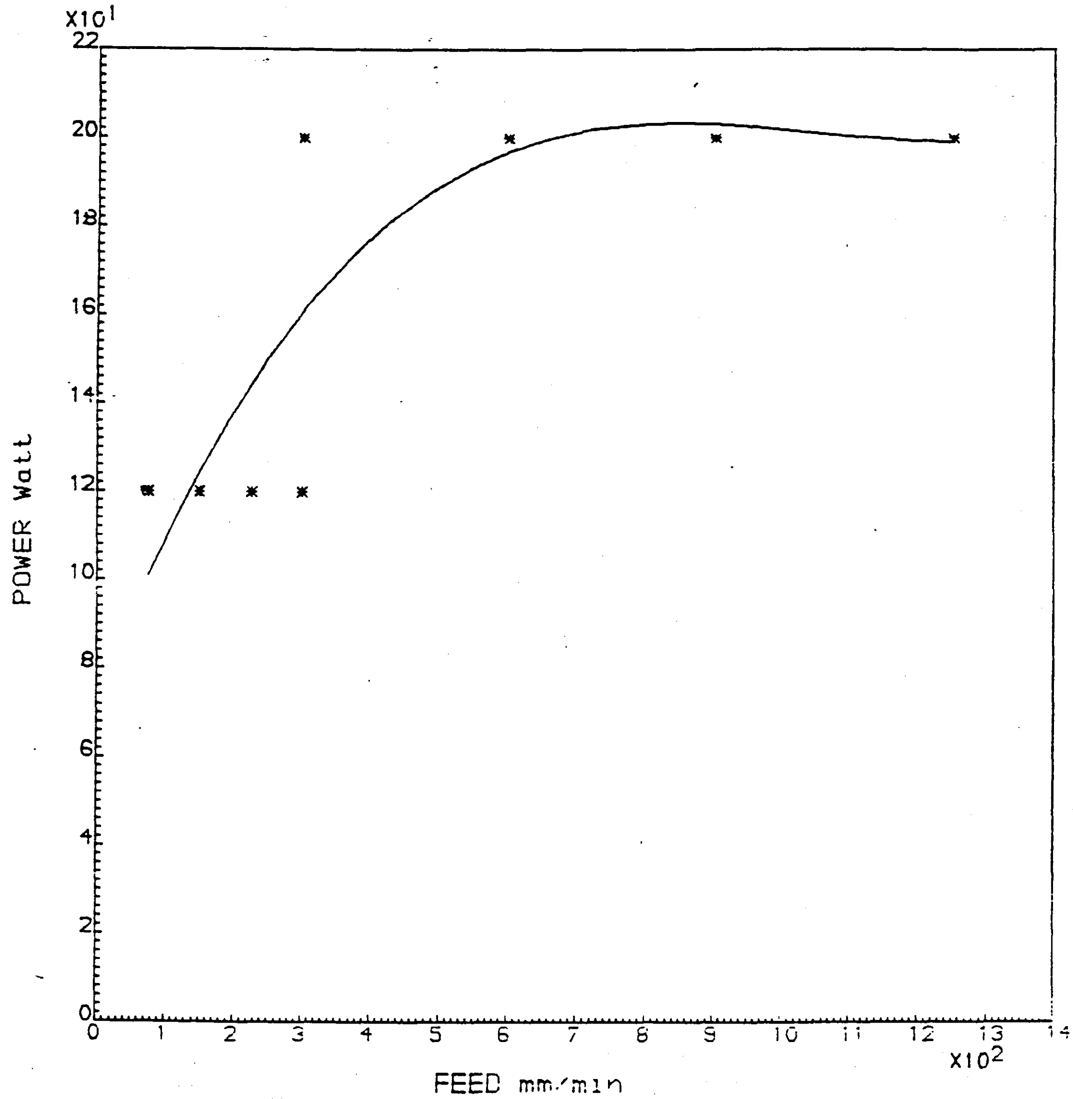


CHART 6.2 : ANALYSIS FOR CW LASER AND SINGLE THICKNESS



POWER Vs FEED FOR 1.6mm MILD-STEEL

FIG 6.7

cancelled, the parameter is reported as constant and this constant value is considered as the recommended setting.

ANALYSIS FOR THICKNESS RANGE

The procedure of this analysis is illustrated in chart 6.3. A three dimensional correlation technique, between feed, power and thickness is employed, after checking that none of these parameters is constant throughout the range of data.

Re-structuring of the input data is required, however, to ensure successful processing by the surface fitting routine (section 6.4.2). The basic requirement of this operation is the creation of a smooth (continuous) set of data from the original (arbitrary) data. Two assumptions were required for this purpose:

- The data of the three parameters (thickness, power and feed) can initially be smoothed by applying two reliable separate 2-dimensional curve fits. This assumption is based on the nature of interaction between these parameters, which suggests that both feed and power are dependent on thickness. This means that correlations exists between feed and thickness, and between power and thickness.

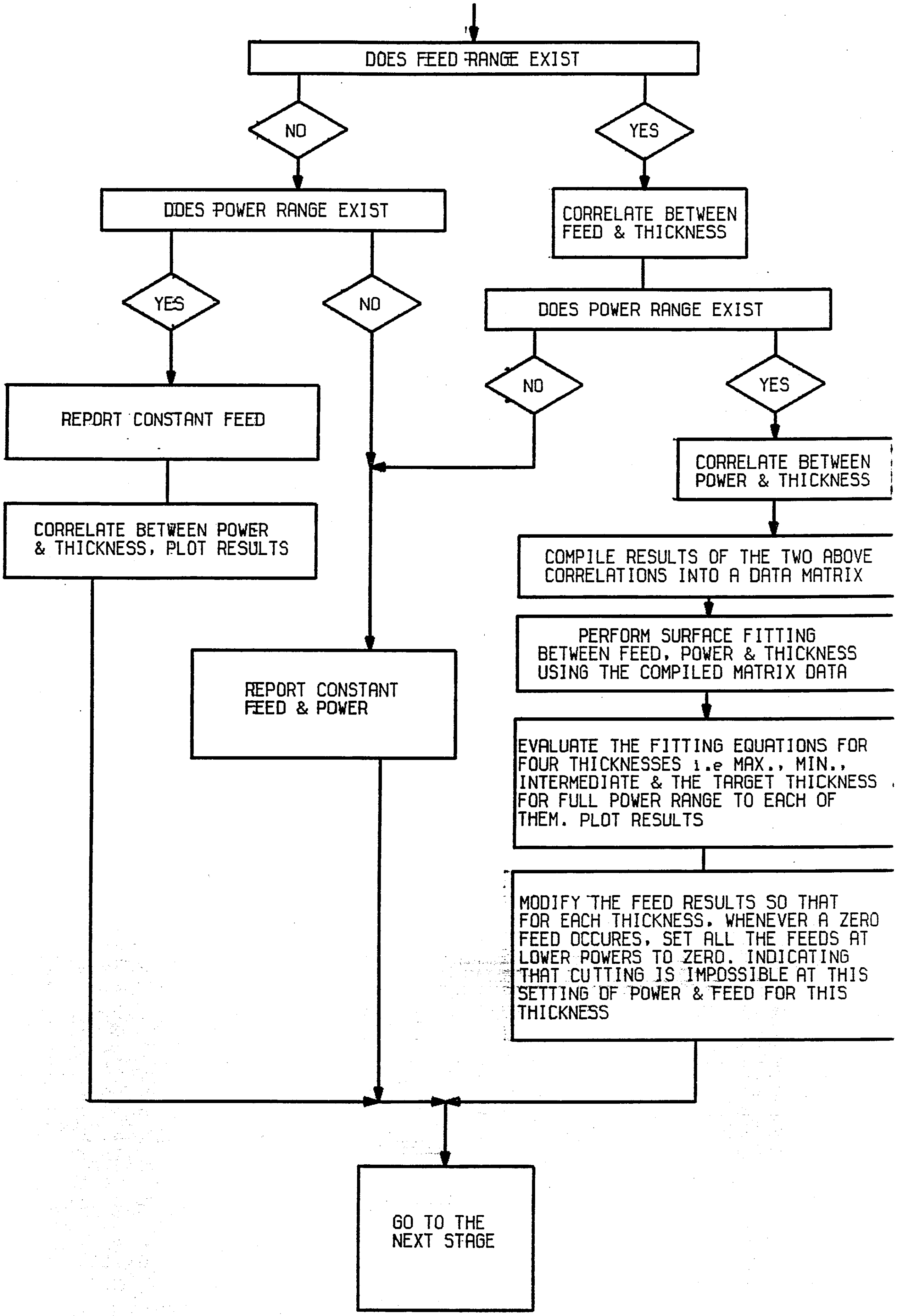


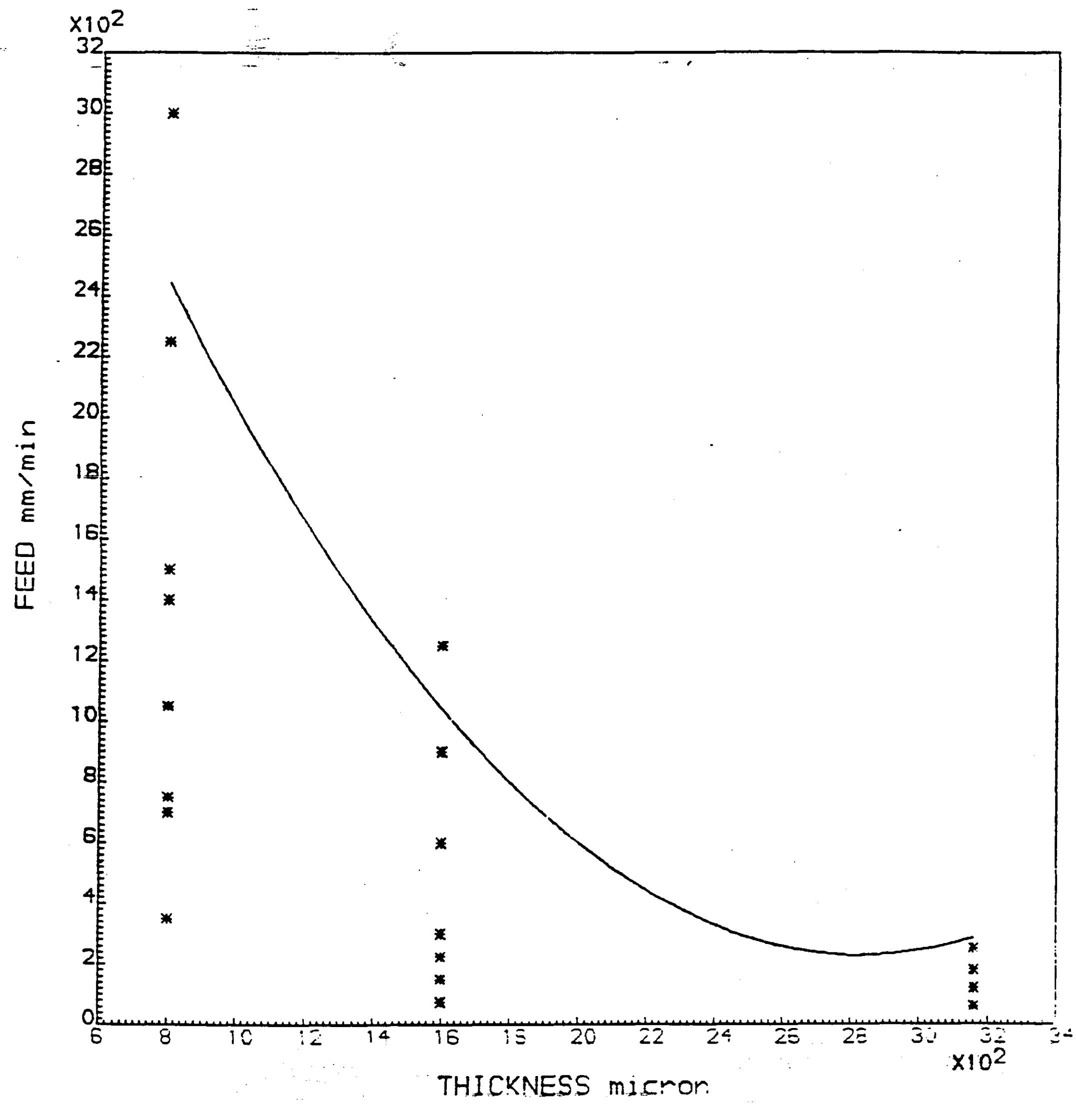
CHART 6.3 : ANALYSIS FOR CW LASER WHEN THICKNESS RANGE EXISTS

- Input data are related to one laser cutting system in order to anticipate correct results from the 2-dimensional fits, as a data set stemming from different systems may yield misleading results (e.g thinner material requiring higher power).

If the original data of the intended parameters are smoothed by applying the two separate curve fitting processes above, the fitted results and the original data can be displayed graphically as in fig´s 6.8 and 6.9. When submitting the correlated data, to the surface fitting routine, as three columns (thickness, feed and power), the results would be as that in fig 6.10.

These results are wrong because of the way the data is presented to the surface fitting routine. Fig 6.11 illustrates the profile of this data, where it can be noticed that the relation between feed and power is incorrect (i.e increasing power accompanied by decreasing feed). Additionally, the two independent variables do not cover the rectangular area required by the surface fitting routine, where only a diagonal line is covered, as in fig 6.12, which results in a poorly structured surface.

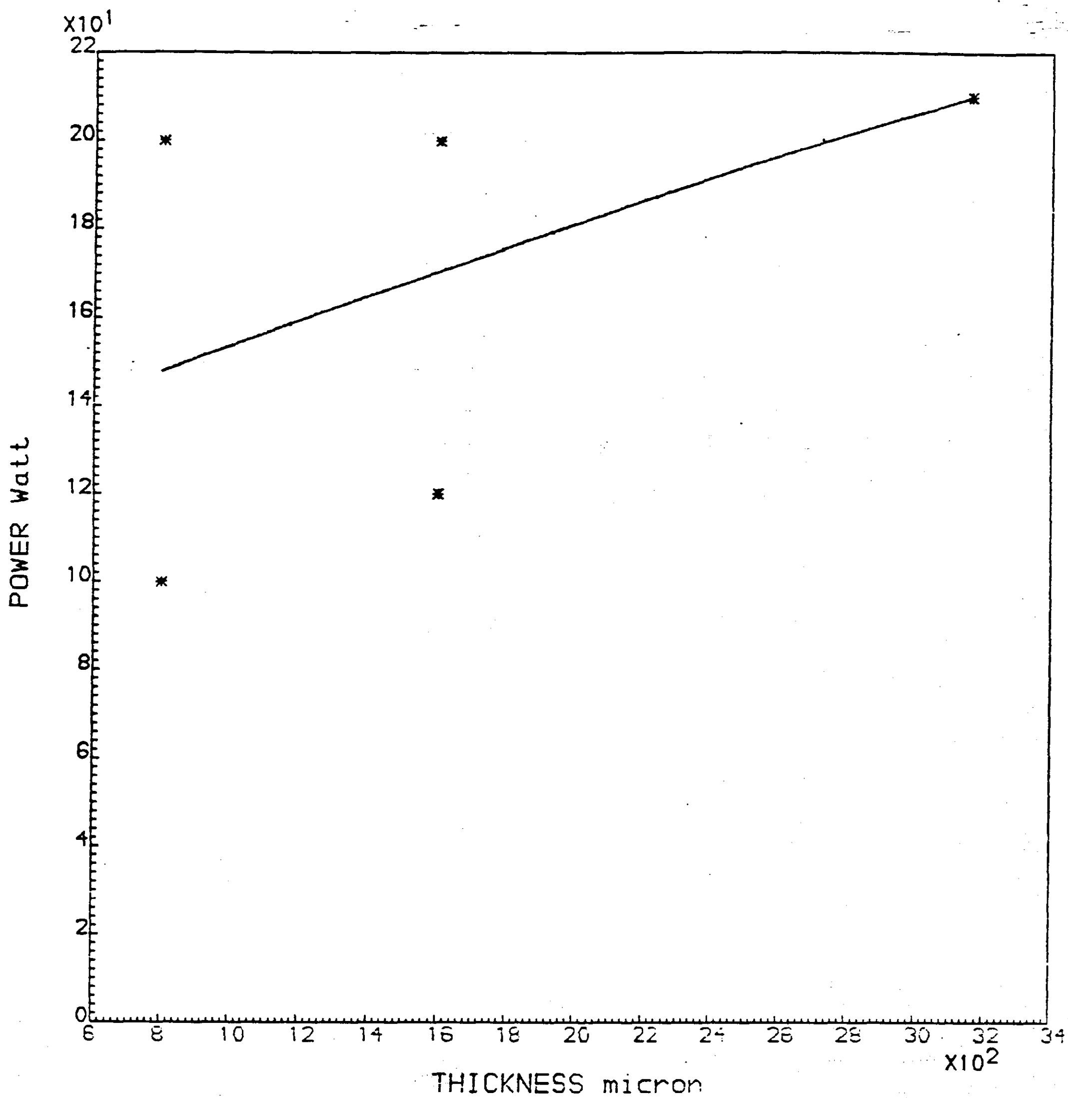
A method was devised to overcome these shortcomings, based on the following assumptions:



FEED Vs THICKNESS FOR BS970EN3B

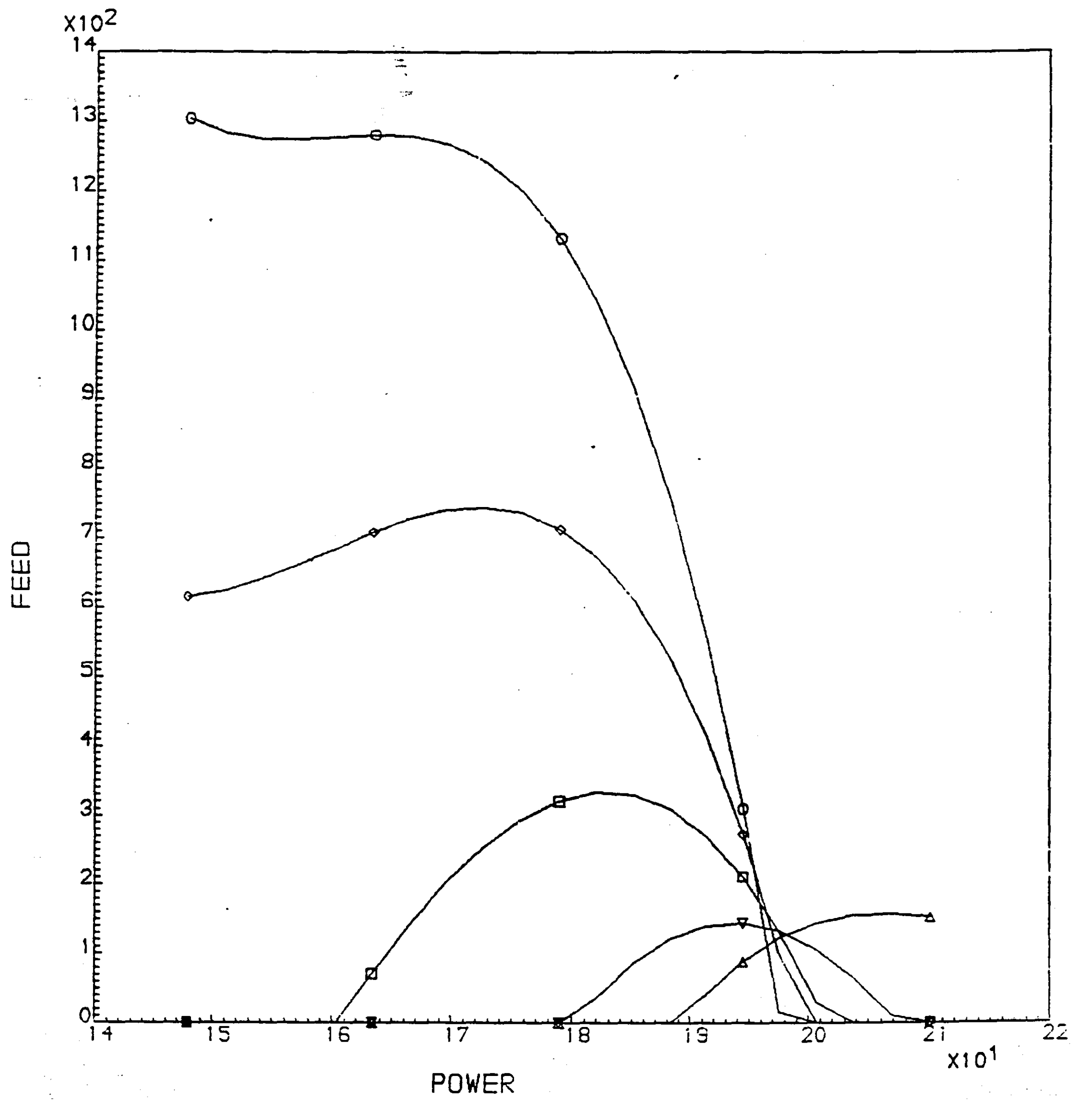
NORMALISED FOR OPTIMUM FEED RATE

FIG 6.8



POWER Vs THICKNESS FOR BS970EN3B

FIG 6.9



○ THICKNESS = 800. ◇ THICKNESS = 1390.
□ THICKNESS = 1980. ▽ THICKNESS = 2570.
△ THICKNESS = 3160.
MATERIAL CLASS > MILD STEEL
MATERIAL TYPE > BS970EN3B

FEED Vs POWER DIAGRAM

FIG 6.10

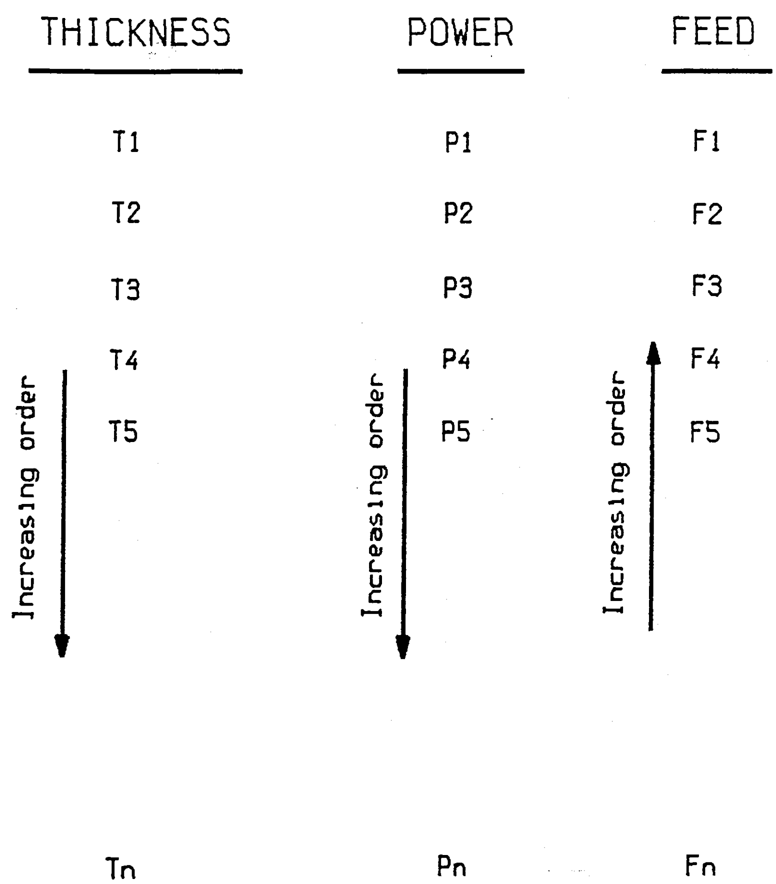


FIG 6.11 : PROFILE OF THE DATA OUTPUT FROM CURVE FITTING PROCESSES OF FIGURES 6.8 & 6.9

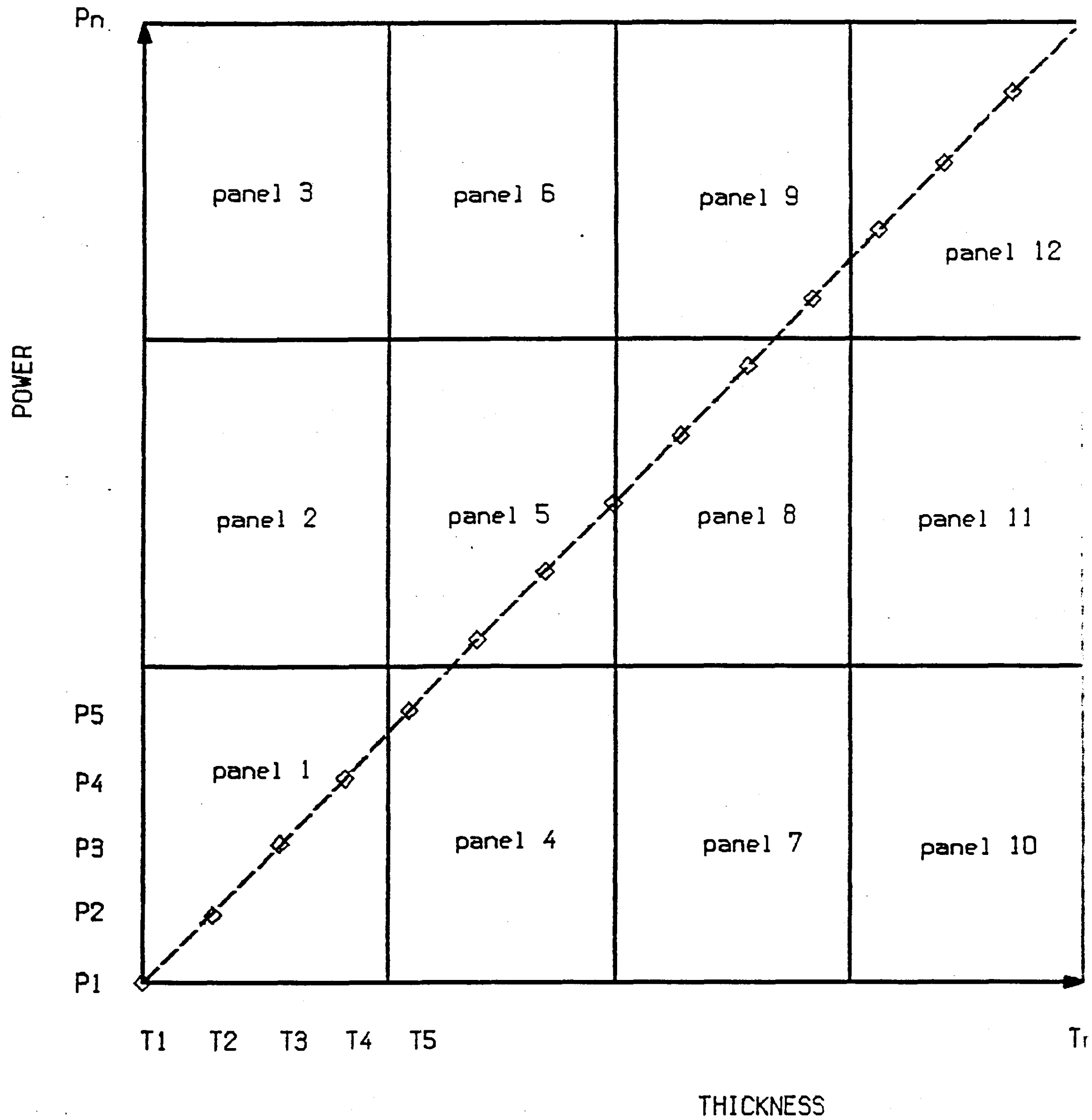


FIG 6.12 : PATTERN OF DISTRIBUTION OF THE DATA POINTS OUTPUT FROM CURVE FITTING PROCESSES, ON THE THICKNESS-POWER PLANE

- Minimum thickness is cut with minimum power at the minimum feed (within the existing range of data). This thickness can be cut at maximum feed with maximum power.

- Maximum thickness can be cut with the maximum power only, at the minimum feed.

A matrix of data as that shown in fig 6.13 will be produced accordingly. When this data is submitted to the surface fitting routine, a surface resembling to that in fig 6.14 is constructed. Sample constant thickness curves representing the maximum, minimum, intermediate and the target thickness, can be projected onto the power-feed plane, to form a graph as that in fig 6.15.

In this method, feed is considered as the dependent variable. This is because most laser cutting systems have limited power and a relatively wide controllable range of feed. Consequently, considering power as the independent variable, ensures that no over-power estimation occurs.

The example shown in fig 6.15 was for a thickness falling inside the range of the original data. Prediction for thicknesses outside the range has also been implemented, by inserting an additional data point, for the target thickness at each of the two curve fitting stages. The fitted curve can be extrapolated up to that point by the surface fitting routine.

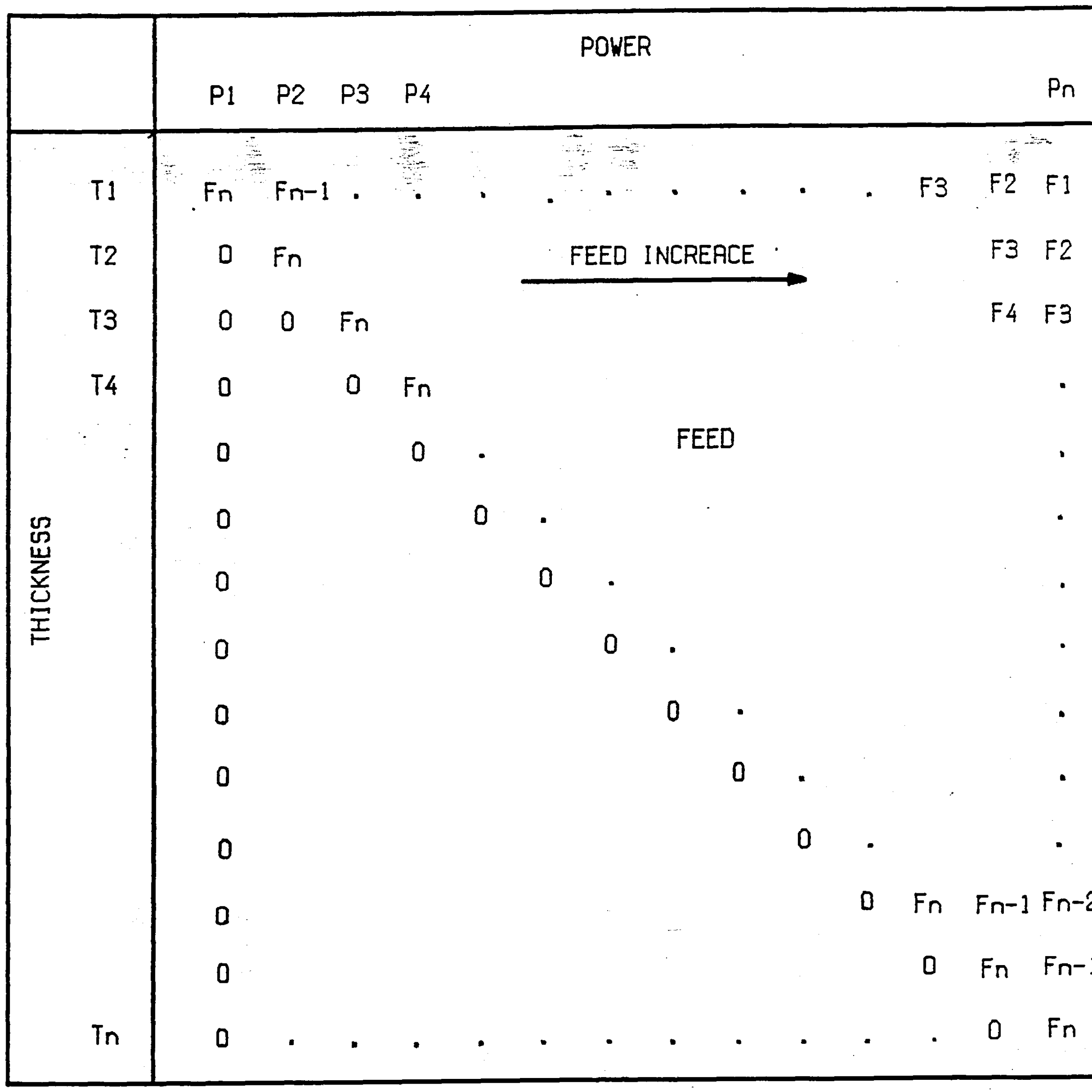


FIG 6.13 : THE MODIFIED DATA MATRIX FOR SURFACE FITTING

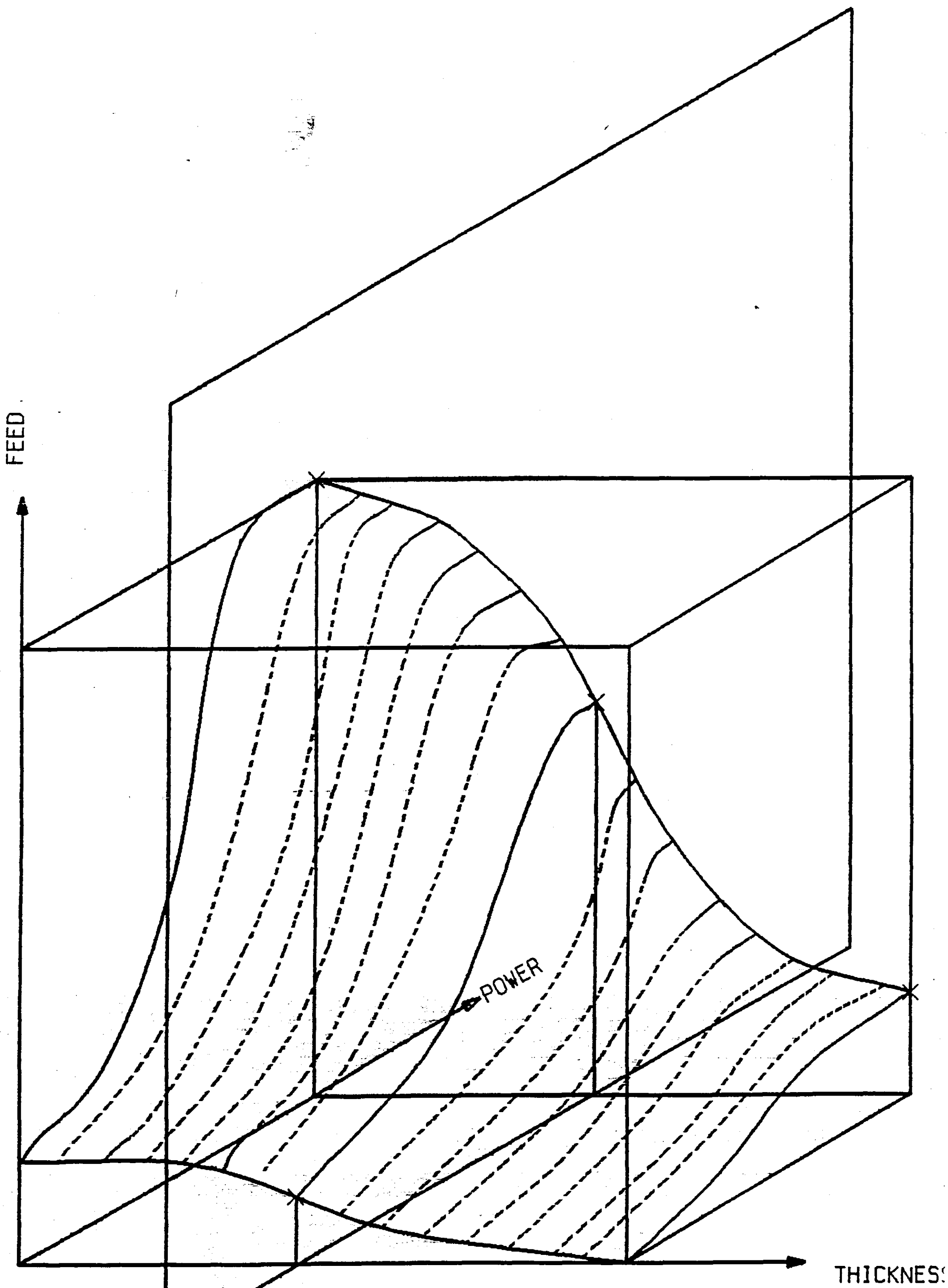
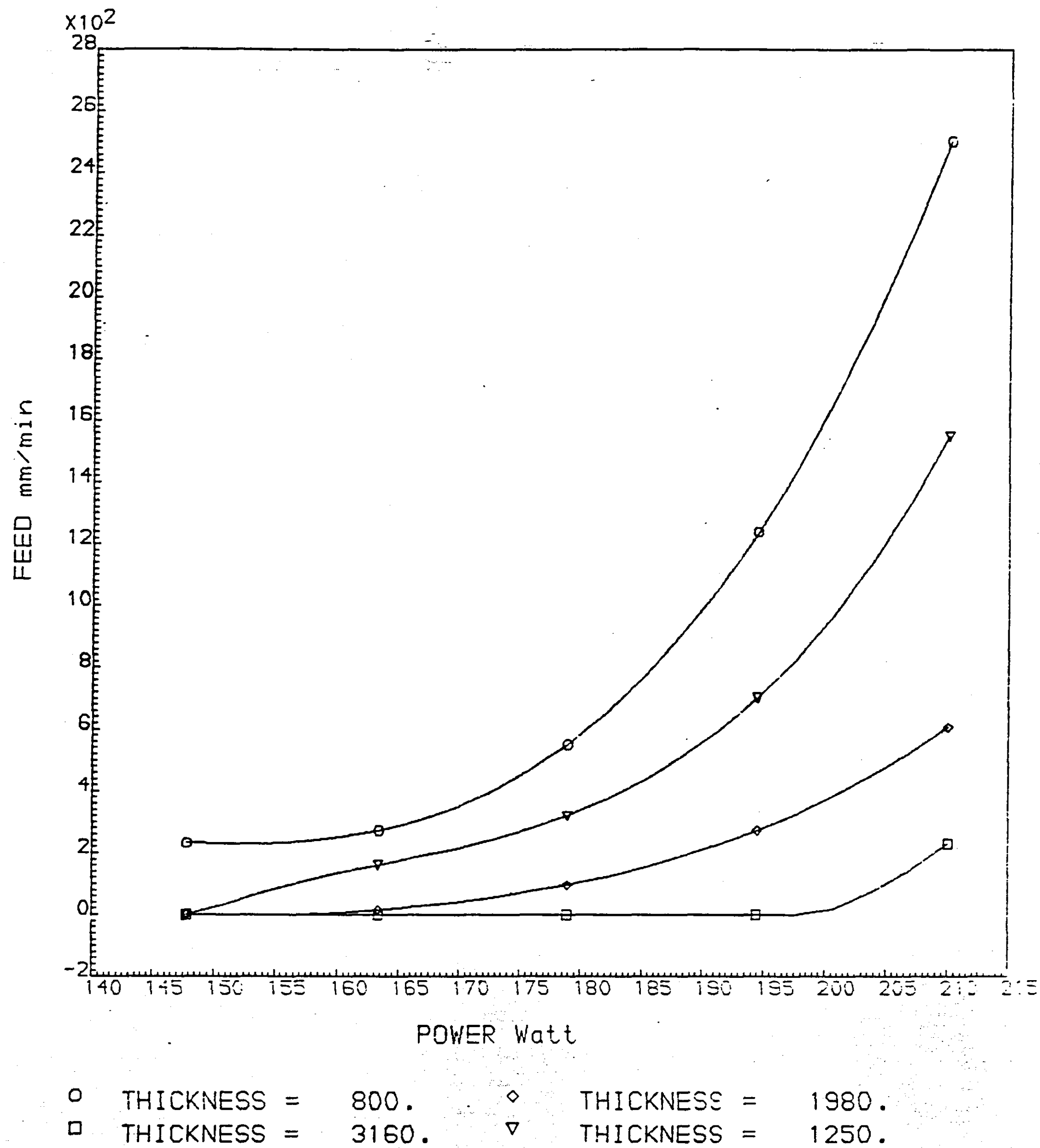


FIG 6.14 : THE SURFACE FITTED TO THE MODIFIED DATA



FEED Vs POWER FOR BS970EN3B

FIG 6.15

6.5.2 ANALYSIS FOR PULSED LASER

This analysis is shown schematically in chart 6.4. The first step in this analysis, is to check whether a range of thicknesses occurs in the input data. If there is such a range, and a range of pulse energy data, a correlation process between the data of the two parameters is performed. The results of correlation, together with the original data are plotted on a graph as in fig 6.16. A pulse energy is then determined from the above results. If the input data is for single thickness, an average pulse energy is calculated from this data.

The second major process, is the correlation between PRF and feed, and plotting the results as in fig 6.17.

6.5.3 FEED RATE PREDICTION ANALYSIS

The procedure of this analysis is presented schematically in chart 6.5. The starting point of this procedure is to re-ensure that there is a range in the feed data. If not, the whole procedure is cancelled, and a constant feed reported (charts 6.2, 6.3 and 6.4).

When there is a feed range, a series of five curve fitting processes are performed, between feed and each of the five cut quality features. The results of these processes are

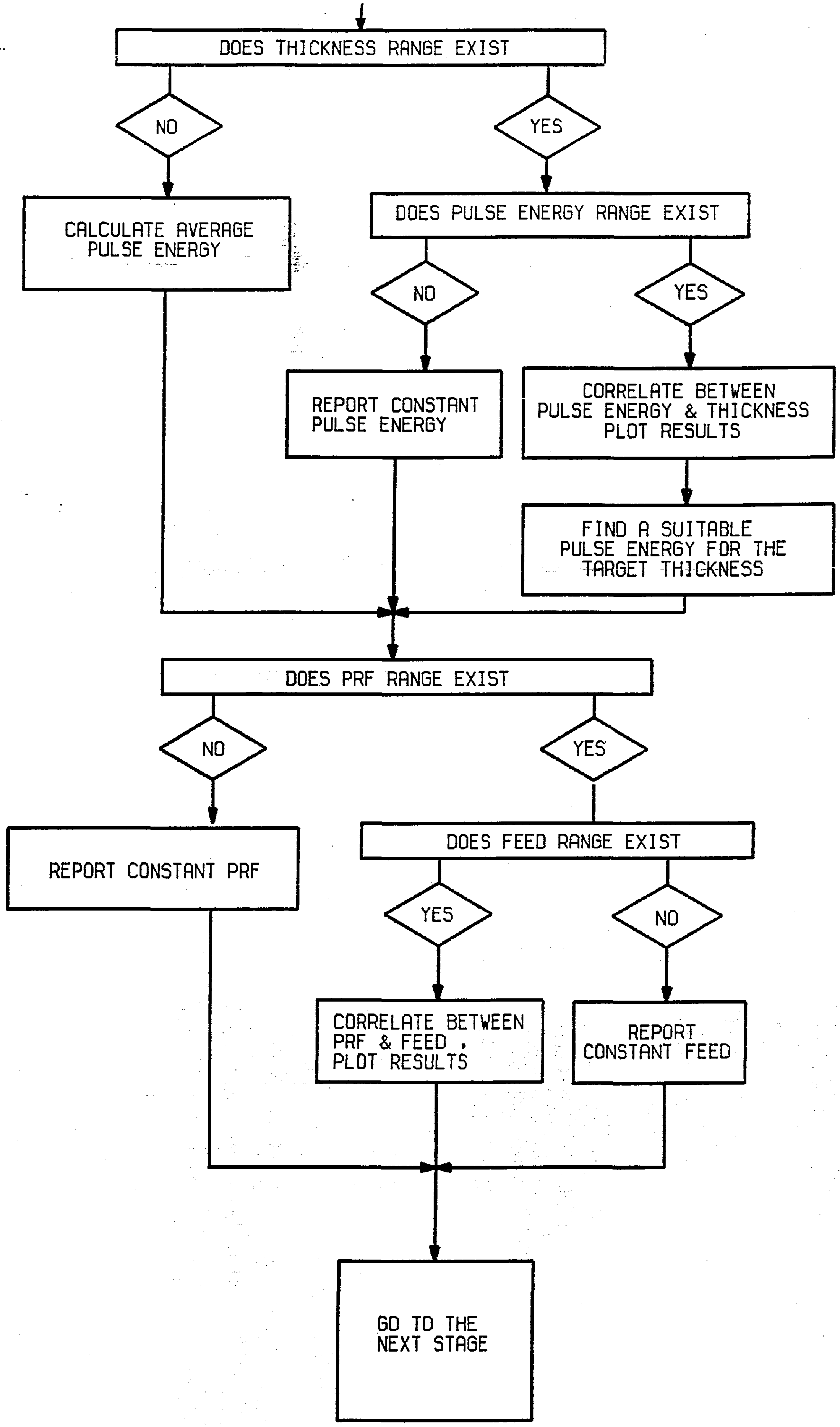
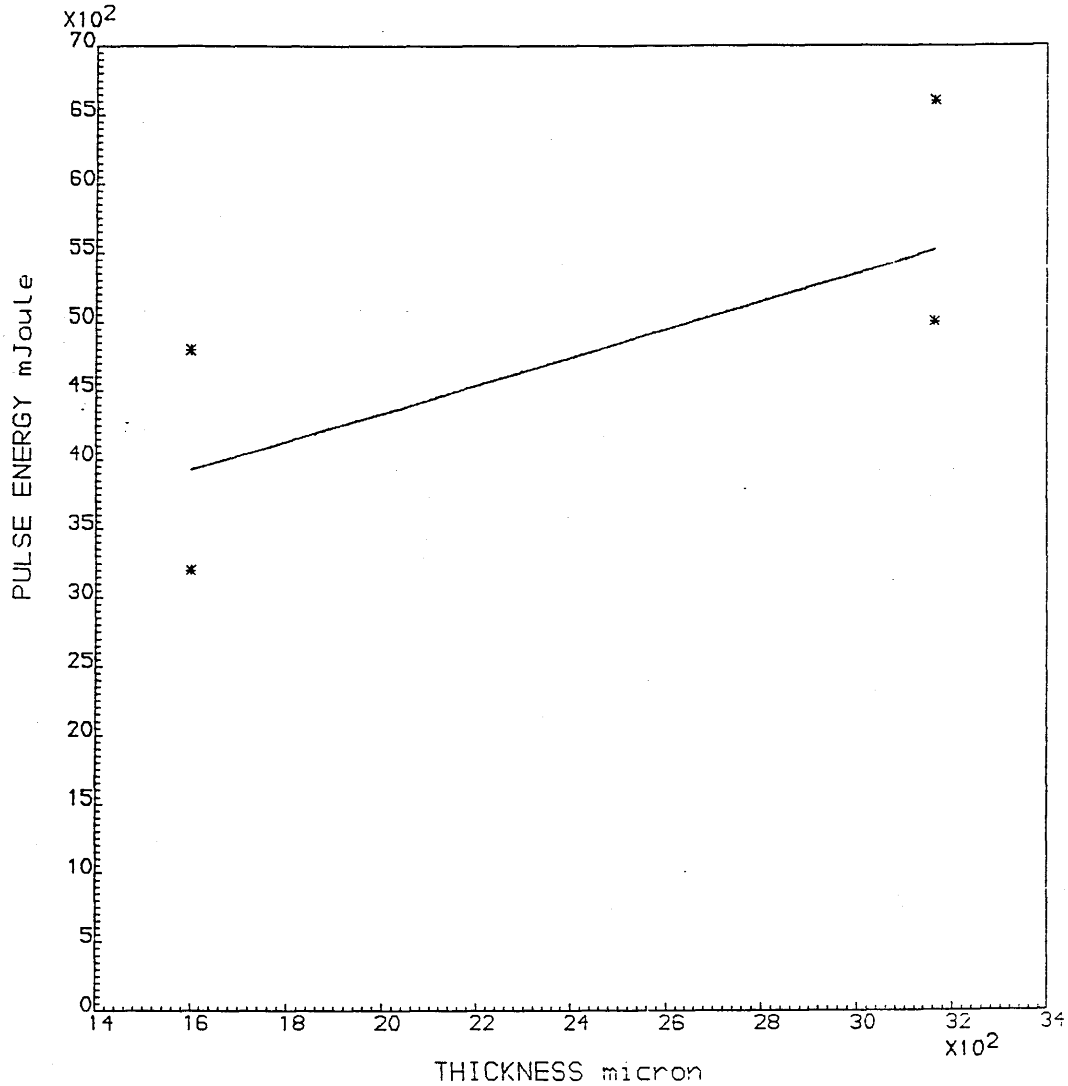
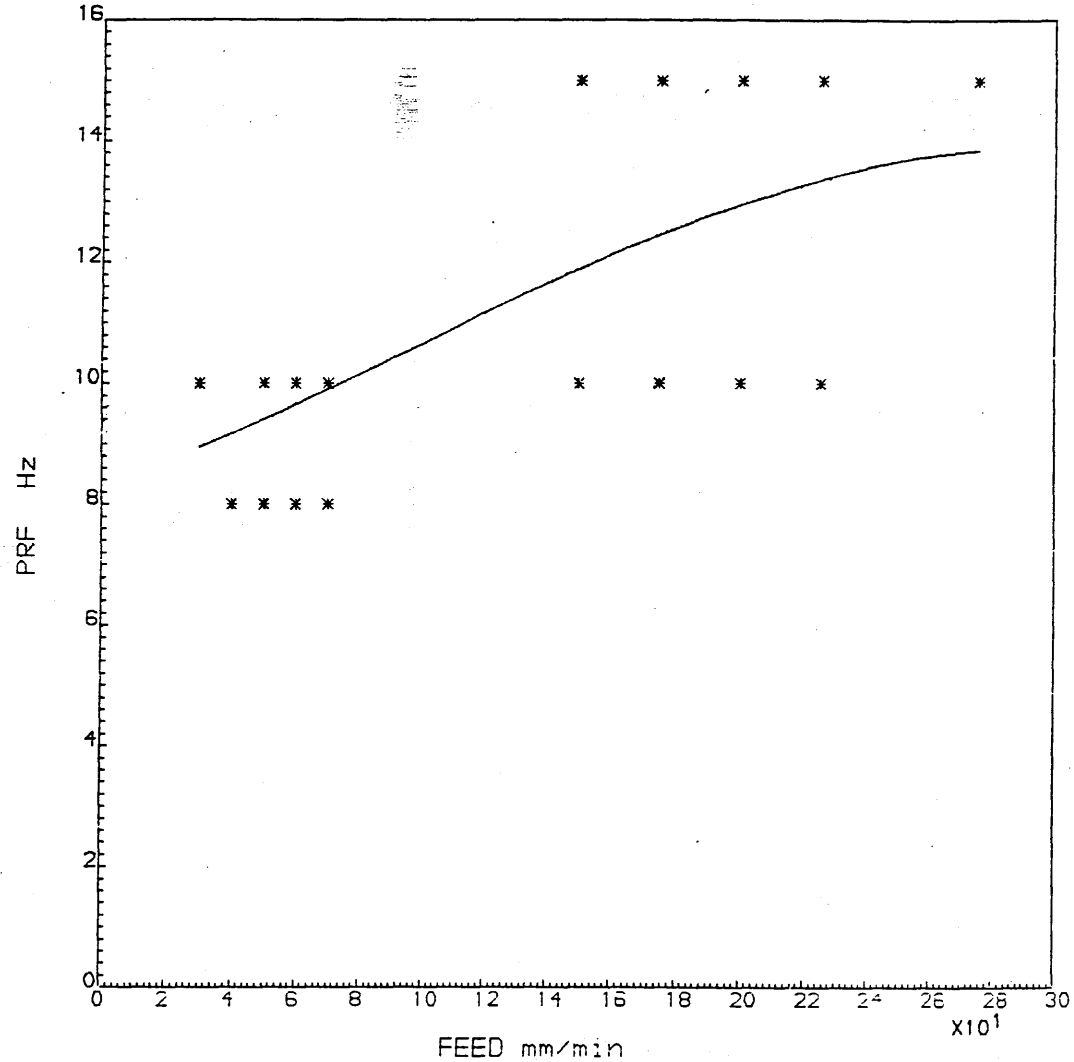


CHART 6.4 : ANALYSIS FOR PULSED LASER



PULSE ENERGY Vs THICKNESS FOR MILD-STEEL

FIG 6.16



PRF Vs FEED FOR MILD-STEEL

FIG 6.17

plotted as in fig 6.18. If any of the features found constant, the corresponding fitting process is cancelled, and a reporting message appears in the graph box assigned for that feature.

Each of the five correlation results above, is searched for a value of feed that corresponds to the minimum value of that quality feature. In other words, this process finds a feed that gives the best of each of the quality features. A tolerance of 10% above the minimum value of each quality feature (10% of the range between the minimum and the maximum), is allowed for higher feed. This allowance should not badly affect the quality approximation but may, generally, allows a large increase in feed rate. This is apparent in fig 6.18, where a slight addition to the minimum fitted value of "TAPER", causes the feed setting to transfer from the minimum to the maximum of its range.

The fitted feed data which are computed by surface fitting (section 6.5.1), are refined before submitting to the next process. The refinement is accomplished by excluding all the non-positive values from the range of feed, i.e the useful range only, is considered for the next process.

The next stage is to calculate a feed rate that would compromise between the different user requirements of quality and fast feed. If the user had not assigned preference weights for the quality features, then it is a

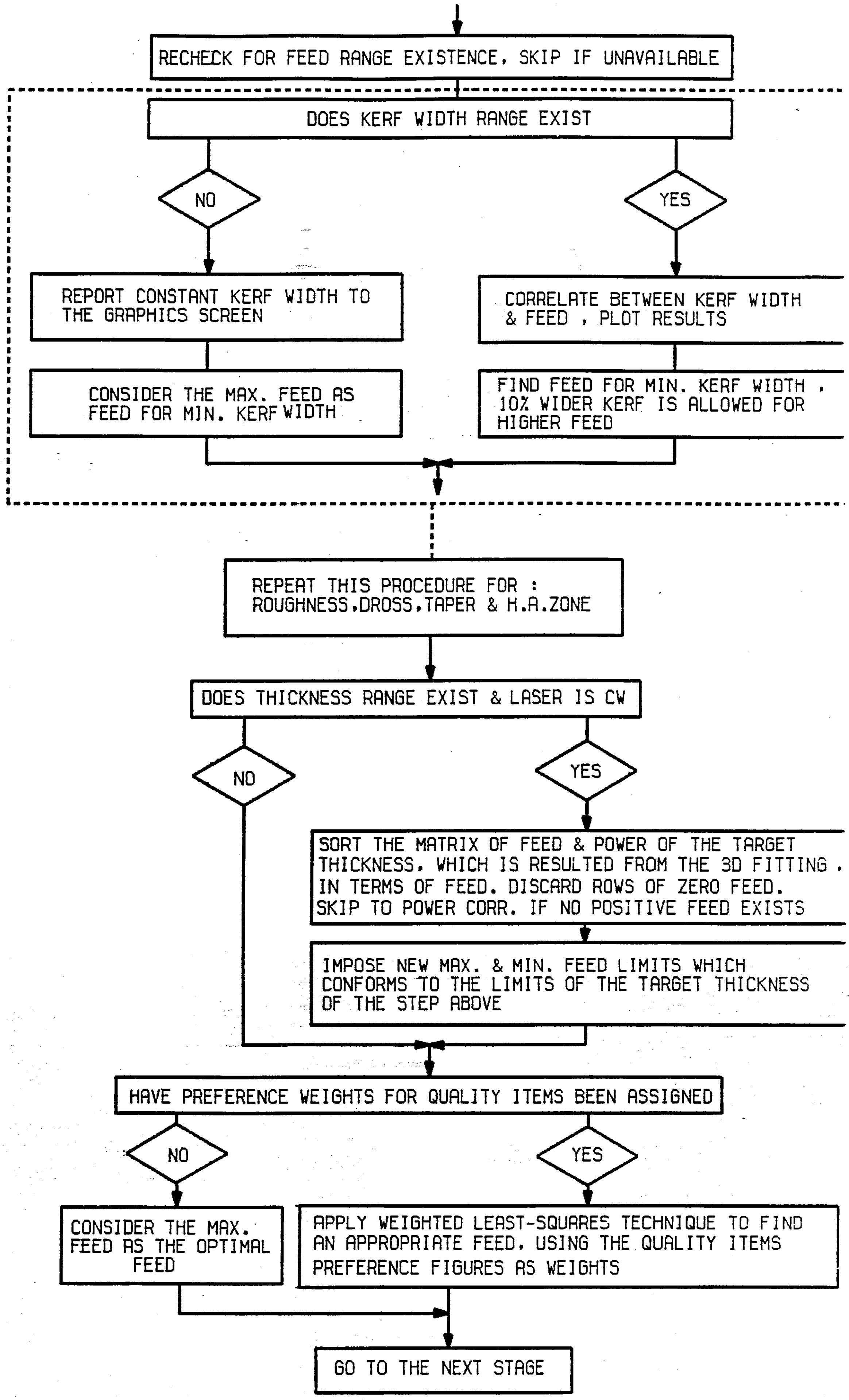
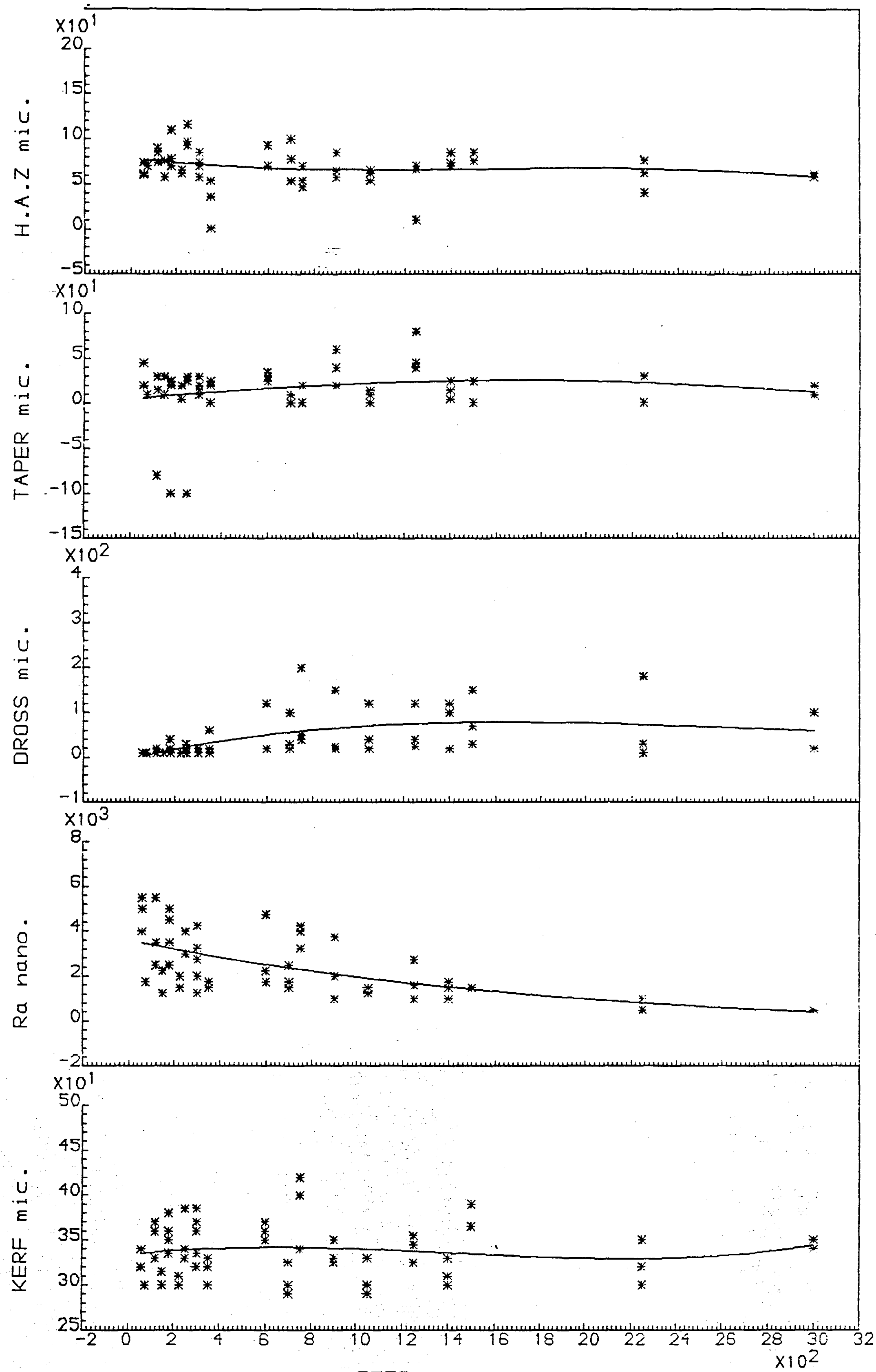


CHART 6.5 : FEED RATE ANALYSIS



QUALITY FEATURES Vs FEED

FIG 6.18

sign of not being concerned about quality. While the concern should be process economy, thus, the maximum feed is taken as the recommended feed. If preference weights were assigned, a feed is calculated by weighted least-squares method, Where the preference weights, for the five quality features and for fast feed, are taken as the weights of the calculation.

6.5.4 ANALYSIS FOR POWER, PULSE ENERGY, PRF AND ASSISTANT GAS PRESSURE

The procedure of this analysis is shown in chart 6.6. The correlation processes between cut quality features and each of the process parameters above, are similar to those in the previous section. Representative samples of the plotted results of these correlations are shown in fig's 6.19, 6.20, 6.21 and 6.22.

The results of these processes are provided for user information only. This is because, the parameters of power and PRF are found from the former correlation results of power against feed, and PRF against feed (section 6.5.2). The correlation processes in terms of assistant gas pressure, are to assist the user to select a setting of gas pressure.

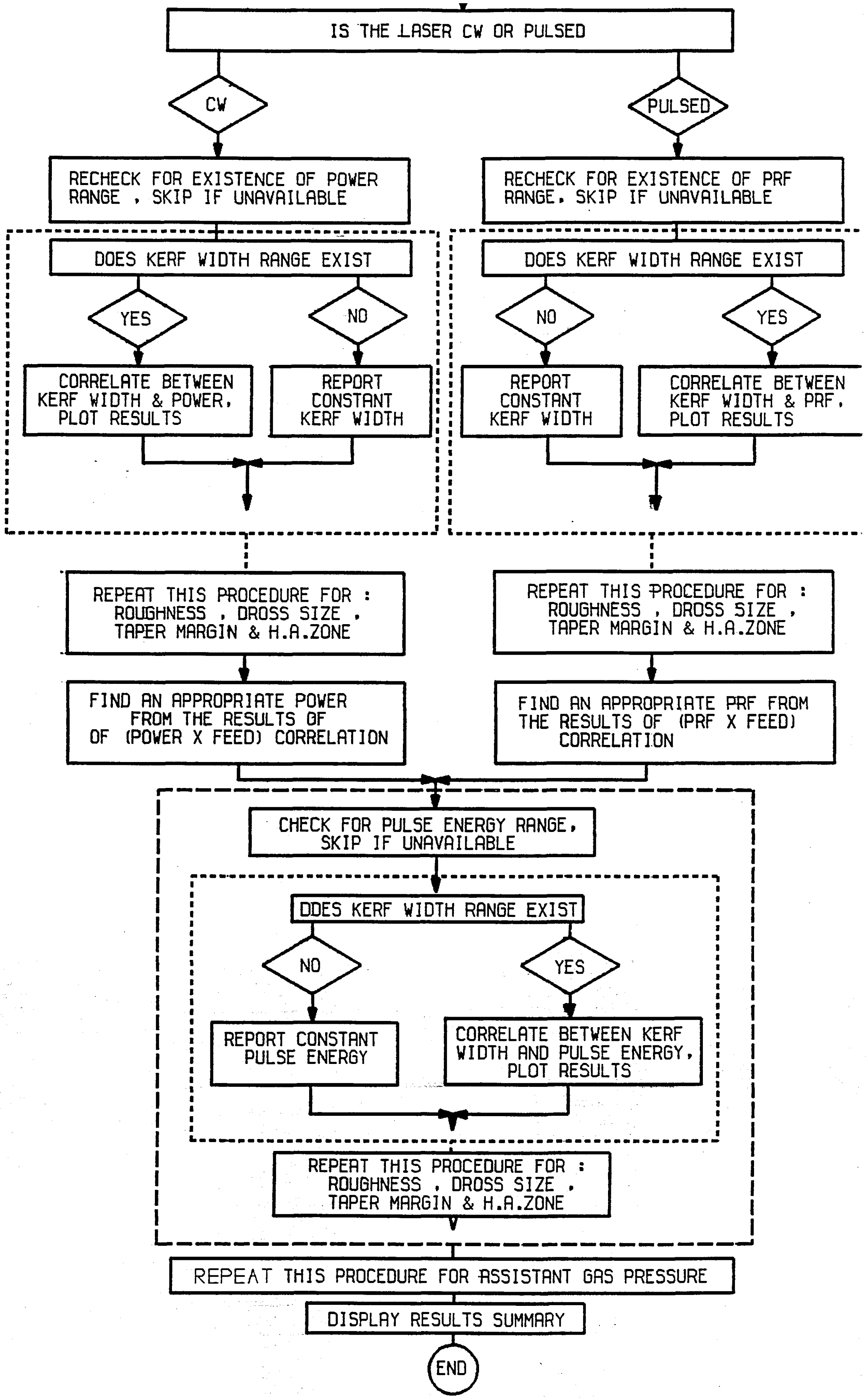
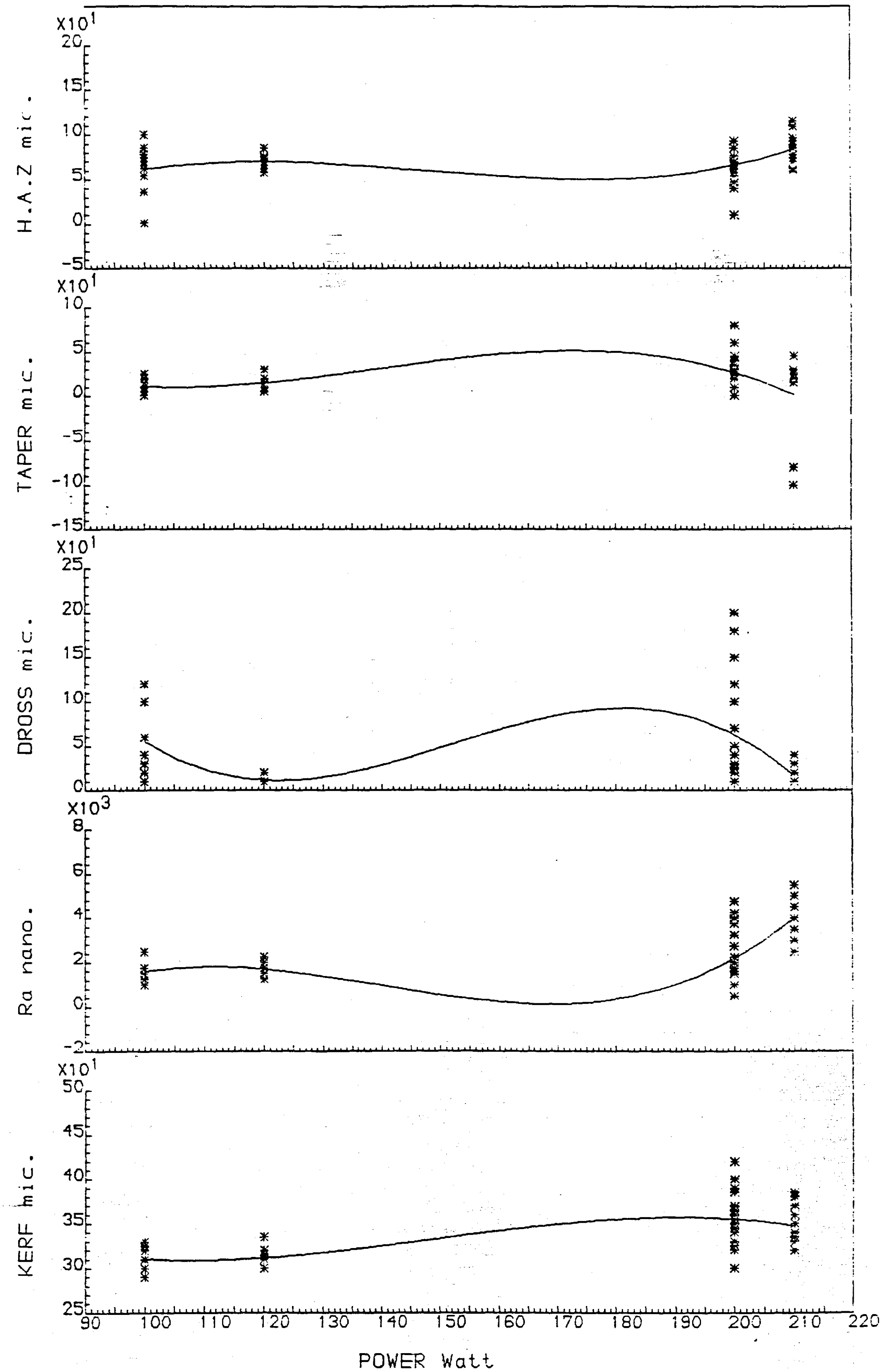
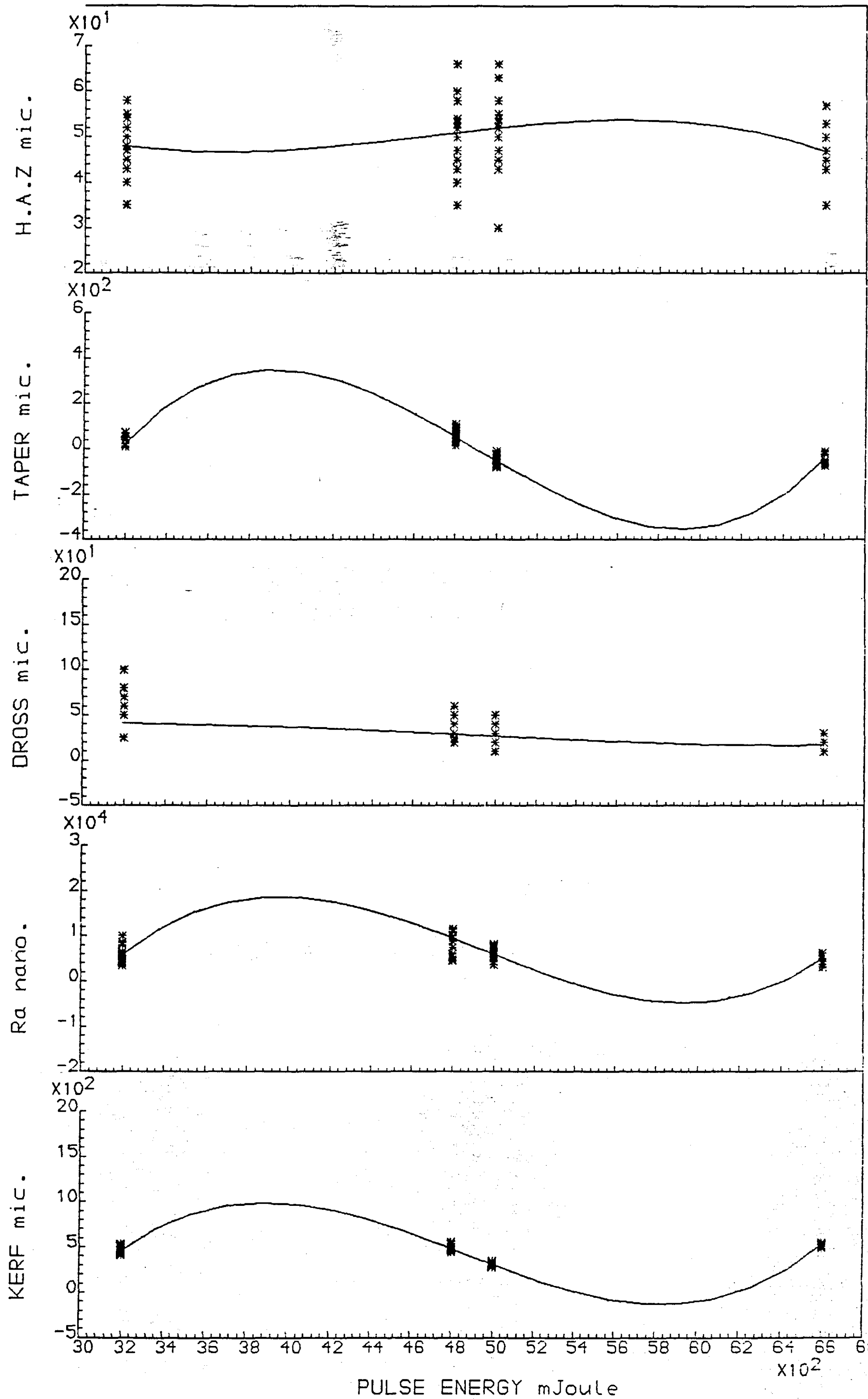


CHART 6.6 : BEHAVIOUR OF CUT QUALITY FEATURES WITH PROCESS PARAMETERS



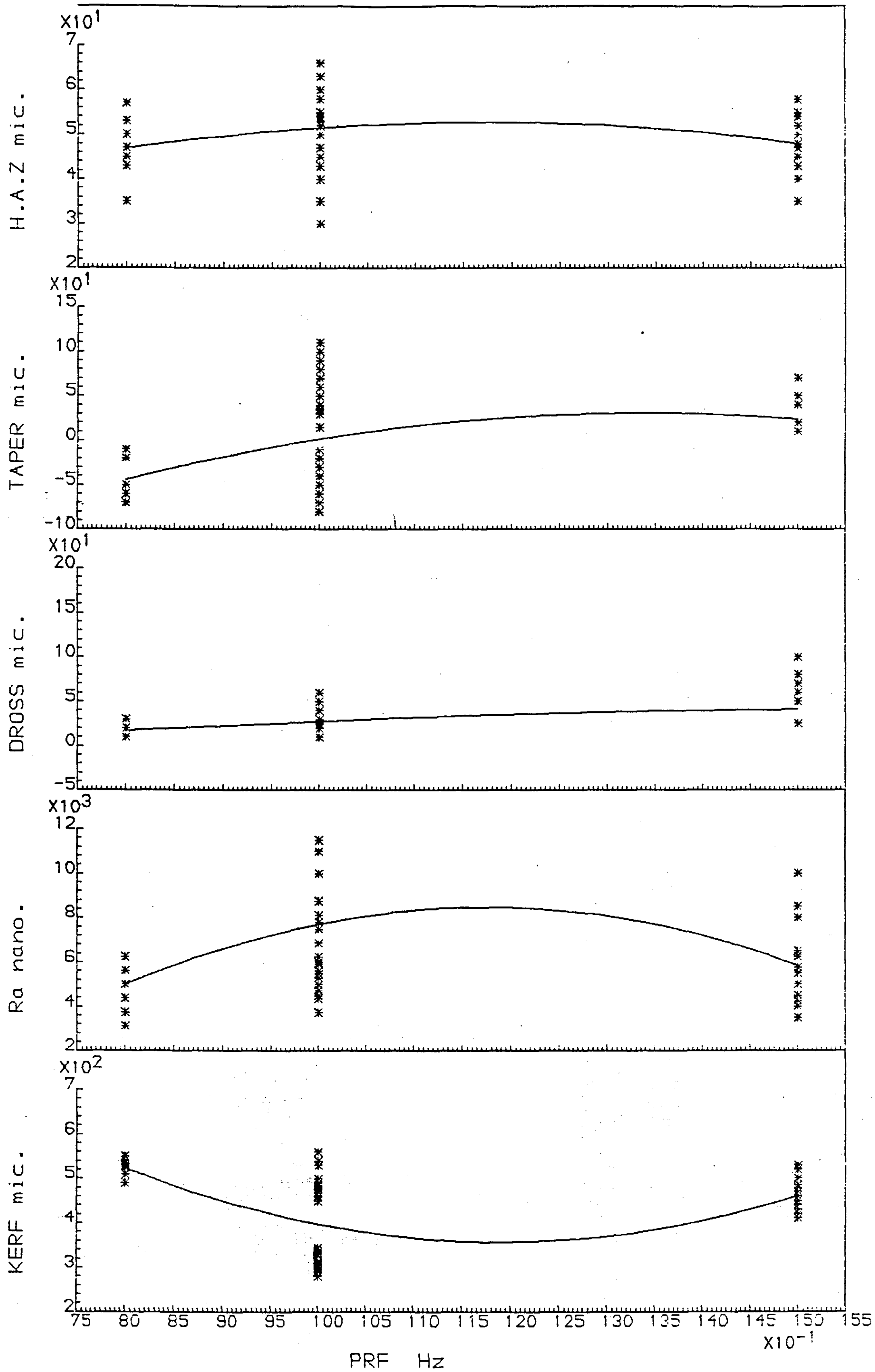
QUALITY FEATURES Vs POWER

FIG 6.19



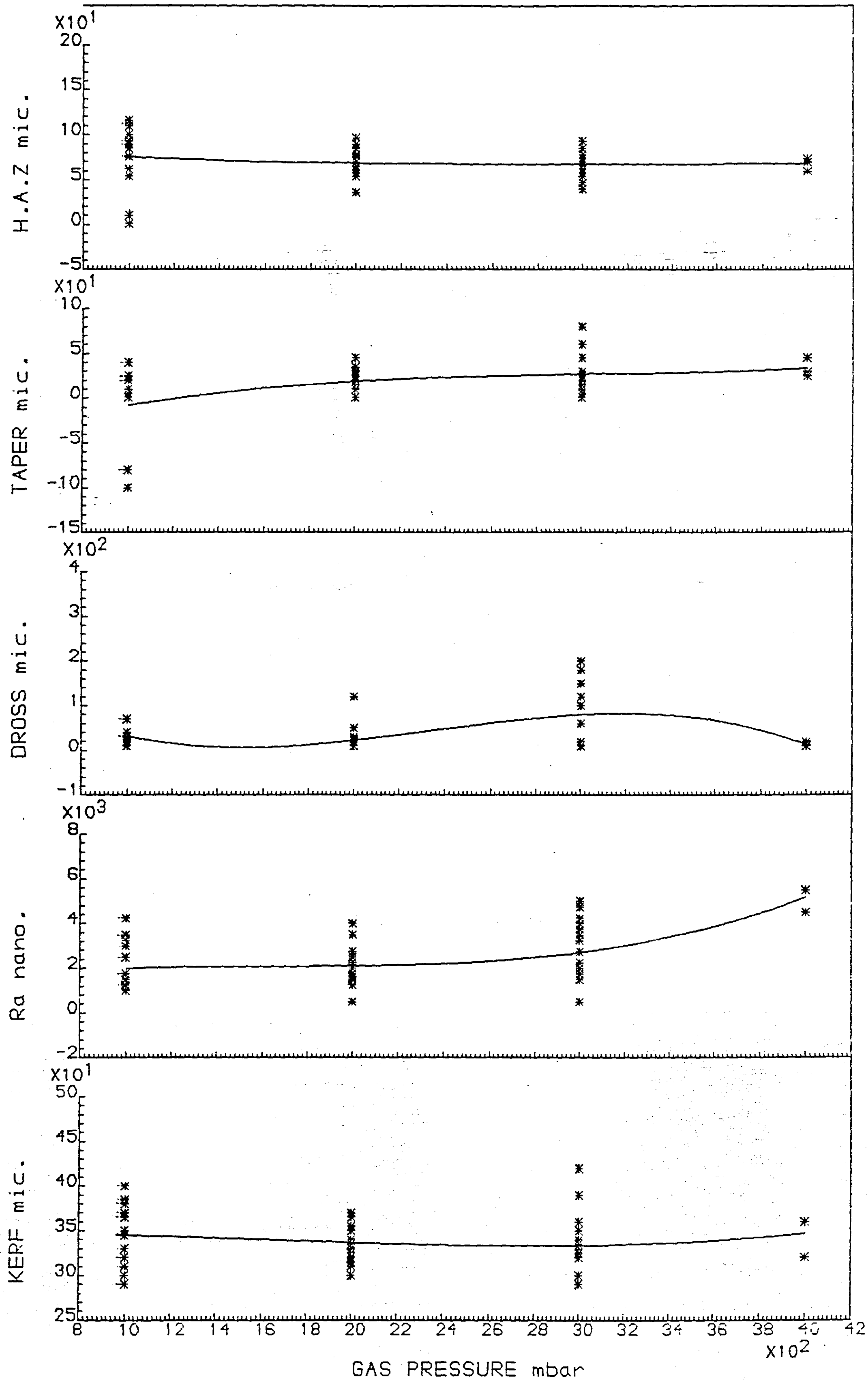
QUALITY FEATURES Vs PULSE ENERGY

FIG 6.20



QUALITY FEATURES Vs PRF

FIG 6.21



QUALITY FEATURES Vs GAS PRESSURE

FIG 6.22

6.5.5 PREDICTION MODULE OUTPUT

The output from this module is organised as: a log file, an output data file and a collection of graphs.

The log file contains any error messages given by the NAG routines.

The output data file is written to the computer memory during the program run. It is then displayed on the graphics device at the end of the run. Fig 6.23 is a representative sample of this output.

The correlation graphs can be displayed on the console screen during the program run, or sent to a plotter for hard copy. In both ways, the output data file is appended to these graphs as the last frame.

6.6 EVALUATION OF THE PREDICTION MODULE

The procedure of this evaluation is as follows:

- Running the prediction program with various sets of data, which are output from the collection program according to specified requirements.

- Application of the process parameters predicted by the first step in cutting experiments.

26-DEC-88 14:25:43

SUMMARY OF RESULTS

MATERIAL CLASS IS MILD STEEL
MATERIAL TYPE IS BS970EN3B
MATERIAL THICKNESS IS 1250 micron

AS THE TARGET MATERIAL THICKNESS HAS NOT BEEN FOUND IN THE DATABASE, INTERPOLATION/EXTRAPOLATION WILL BE USED TO PREDICT APPROXIMATE SETTINGS.

FEEDS THAT CAN ACTUATE MIN. KERF,ROUGHNESS,DROSS, TAPER & HAZ ARE ,RESPECTIVELY, AS FOLLOWS :

1548. 1548. 60. 60. 1548.

WITH PREFERENCE WEIGHTS OF :

0.20 0.80 0.00 0.00 0.50

WHILE THE PREFERENCE WEIGHT FOR MAXIMUM FEED IS 0.30

THE RECOMMENDED FEED IS 1517.mm/min.

THE RECOMMENDED POWER IS 210.Watts.

FIG 6.23

- Post-operation measurements of cut quality features resulting from the cutting experiments.

- Comparison of the measured results against the predicted results.

Four output samples of the prediction program used for this evaluation, are included in Appendix 6C, these are:

a) Presents the prediction for cutting with the YAG laser, when data for the target material thickness is found.

b) Presents the prediction for cutting with the YAG laser, when the target material thickness is outside a range data of existing thicknesses (extrapolation).

c) Presents the prediction for cutting with the CO_2 laser, when data for the target material thickness is found.

d) Presents the prediction for cutting with the CO_2 laser, when the target material thickness is within a range of data of existing thicknesses (interpolation).

The YAG laser experiments were performed with gas pressure and stand-off distance settings of 2bar and 0.75mm respectively, and the CO_2 laser experiments with 2bar and 1.0mm setting. These represent the mean values of the settings used in the original cutting experiments, in order that average effects of these parameters can be anticipated.

The predicted parameters and cut quality features are, generally, in line with the experimental results. However, differences should be expected due to variations in laser characteristics, particularly beam mode. More variances between predicted and experimental results were experienced with the YAG than the CO_2 laser. This is, mainly, due to cavity re-alignment carried out on the YAG laser during the period between performing the original cutting experiments and the evaluation experiments, which may cause changes to beam mode, spot size, power intensity and optical stand-off distance.

The following conclusions can be drawn from the examination of the above evaluation samples:

1) The experimental results of the YAG laser cutting tests reveal, in general, narrower kerf widths, smoother cut edges, smaller taper and larger dross size, compared with the predicted results. Heat affected zone measurements were quite comparable to the predicted values.

2) The CO_2 laser cutting tests exhibited a good correlation between the predicted and experimental results, though slightly better actual measurements than is predicted.

3) The prediction for missing thicknesses among data of existing thicknesses by interpolation (sample d) and extrapolation (sample b), exhibits a fairly successful performance when comparing the measured results, which is clearly announced in sample (d). This performance is further proved by the fact that the profile of variances between the practical and predicted results of these tests, is following the profile found in samples (a) and (c) which should be more accurate as they deal with the exact material thickness.

4) In some cases, the variances of cut quality features along the prediction curves are minor, while a rather slow feed rate is predicted (as in sample a), mainly due to the values of preference weights assigned by the user. This phenomenon highlights the usefulness of the method in which the prediction program results are presented, where the prediction can readily be assessed by the user, as the variances of cut quality features against various cutting parameters, and the correlation between the original and predicted data are shown clearly.

REFERENCES

REF. No.	DETAILS
6.1	Morris, J. Ll., Computational Methods in Elementary Numerical Analysis, John Wiley & Sons, 1983.
6.2	Hamming, R. W., Numerical Methods for Scientists and Engineers, 2nd Edition, McGraw-Hill Inc, 1973.
6.3	Anon, The NAG Fortran Library, Mark 13, The NAG Algorithms Group Inc, 1988.
6.4	Cox, M. G., The Numerical Evaluation of B-Spline, J. Inst. Maths. Applics, (1972) 10, PP134.
6.5	Hays, J. G. and Haliday, J., The Least-Squares Fitting of Cubic Spline Surfaces to General data sets, J. Inst. Maths. Applics, (1974) 14, PP89.

CHAPTER 7

DISCUSSION AND CONCLUSIONS

The invention and development of many kinds of industrial lasers, and the use of CNC technology for workpiece or laser beam handling, has allowed laser cutting to become almost standard practice, especially in sheet metal fabrication. However, this needs highly skilled operators to achieve the required quality as the cutting process is influenced by the interaction of many physical quantities and component parameters. In practice, each successful application depends upon extensive process development, and requires indefinable number of cutting trials.

To diminish the dependence of laser cutting process on human expertise and to avoid process errors, research and development effort was concentrated on the following alternatives:

- Mathematical simulation of laser material interaction.
- Experimental studies of laser cutting processes.
- Rule based in-process control of laser cutting process.

Simple mathematical models are unable to predict accurate results as they cannot handle the many parameters involved in the process and their complex interactions. As an example, it is very difficult (if not impossible) to determine the proportions of the material removed as vapour or in molten phase, or to determine the energy supplied by exothermic reaction between the material and any reactive assistant gas which might be used. On the other hand, more complicated models are very demanding of processing time (12 hours to simulate the interaction of a single laser pulse) which makes their use unjustifiable in terms of cost and time. They also limited in their ability to predict cut geometry.

Experimental studies to determine process trends, i.e examining the effect of each process parameter upon the process performance and output cut quality, would require a very large number of experiments over a long time scale in addition to equipment and materials (more than 65000 experiments or 33000 man.hr are needed to study 8 variables with 4 levels each). The results of this study could be specific to one material and one laser system.

The process is also difficult to control using rule based systems as a very large number of rules have to be produced for different types of materials, thicknesses, input cutting parameters and output quality, which militates against real-time operation required for in-process control. Moreover, the process of developing a reliable set of rules requires extensive cutting experiments and analyses (150 experiments and 40 rules for one material only, ref 2.6).

Therefore, an alternative solution is needed, a system that may incorporate the desirable features of the solutions above, a system that should be:

- Reasonably simple to implement.
- Generic, to span a wide range of applications.
- Comprehensive, to handle all parameters.
- Efficient, to generate control strategy in a short time, a strategy that is fast enough for real-time operation.
- Flexible, to allow for necessary future developments.

It is believed that a system that is based on an empirical approach of predicting process parameters and adaptive control strategies, using previously recorded experimental data, is the only alternative that fulfill these requirements. The continual acquisition of relevant data over a period of time, will cause the system to improve its performance. This system operates in the sequence (see fig 2.4):

- Recording laser cutting data in a data management module which can consistently identify them. The data can be generic for any laser cutting application.

- When a process control is required for a specific application, a data set relevant to the application is selected from the data store. It should be expected, however, that a data set that is fully compatible with user requirements might not be found. Therefore, data selection have to be controlled by an iterative data search procedure that justifies user constraints, by relaxing or dropping a constraint after each unsuccessful search, according to a preset priority order (see chart 5.6).

- This data is then used collectively in a correlation operation to predict a set of process parameters, generate a control matrix for in-process control and identify process trends.

This system will necessarily requires efficient and flexible facilities to fulfill these tasks, these are:

- A method to control data acquisition and manipulation.
- A method of selecting (collecting) suitable data.
- A method of correlating between the various parameters to generate a control trajectory and strategy.

A modular system for the tasking and control of laser cutting has been designed. The three basic modules (data management, collection and prediction) have been implemented, and data from 610 cutting experiments was used in the development and testing of modules. It is worthwhile to state that this approach can be used to implement planning and control systems for other types of laser material processing, by adapting system's modules for the new application.

Evaluation of the current system suggests a good level of performance, where the average prediction error of cutting parameters, from 20 data records only, was less than 20%. The time taken by the system to question a user and collect a set of suitable data (among 800 data points) is less than 1.5 minutes. The process of prediction and displaying all the graphical results on the screen, takes an average of 2.5 minutes (in a multi-user system), and 6 minutes for hard copy.

The effect of data growth can be summarised as:

- Better accuracy of prediction and process control will result. However, this improvement is expected to be exponential rather than linear, an example, is when doubling the data from 10 to 20 records may reduce prediction error from 50% to 20%, while a doubling from 1000 to 2000 may result in 3% reduction only. This also depends on data distribution within the range of the collected data, more details are given (in 4) below.

- Speed of data collection is mainly affected by the growth of cutting data (slower by 2/3 of data growth rate) in the material specific database tables (see fig's 3.4 and 3.5 for these tables). However, this can be rectified by creating more material tables with fewer data in each of them, i.e partitioning the data sets according to specific material type or thickness. This will further improve prediction accuracy as the collected data can be more material specific. Moreover, familiar operator can reduce questioning time from about 1 minute to 10 seconds .

- Speed of prediction would be affected marginally as the mathematical computation processes are mainly simple 2-dimensional fitting routines, which are much faster than higher order types (a 3-D fitting process is almost 10 times slower than a 2-D).

The following part of this chapter compiles the significant conclusions concerning the individual system modules and the experimental work:

1) The use of a relational database management system to manage the acquisition of laser cutting data, provides a flexible, efficient and secure means for the definition and manipulation of vast data sets that may accumulate during the production process.

2) A data collection module was developed to access the database, and to collect a set of data relevant to the user´s inquiry. The modular design adopted, provides the flexibility required for the expansion and modification of the program at later stages of development.

The design of the interactive user input procedures, of this module, provided a good means for user understanding as to the meaning and importance of the data he supplies, by displaying; information messages, important data within the database and possible options for some answers. This secures the integrity of the user supplied data.

3) The method of predicting cutting process parameters and cut quality, using a set of data provided by the data collection module, is generic for all cutting applications, and based on using curve fitting to correlate between output and input parameters (e.g cut quality features with feed), or amongst the input parameters (e.g power with feed). Therefore, the selection of stable fitting methods is very important for prediction reliability.

The prediction accuracy depends, also, on the distribution of the data, which is collected by the collection module, within the plane of correlation. Some forms of data distribution can cause difficulties as the data may not be readily amenable for fitting by some of the existing fitting methods, because of excessive fluctuation (as in interpolation and high degree polynomial fitting), or failure of the fitting process (as in cubic and bi-cubic spline fitting).

It is shown that 2-dimensional polynomial fitting is the most appropriate technique for these data as:

- Reasonably accurate predictions were achieved, while higher order correlation methods were extremely inaccurate (as in maximisation/minimisation method).

- Does not force limitations on data distribution and prevents processing failure, while cubic and bi-cubic are vulnerable to processing failure when repeated data occur in one or more parameters.

- facilitates user ability to select polynomial degree of the fitted curve, which provides a flexible means for more accurate approximation.

- High order correlation cannot be graphically represented, while this is readily possible in 2-dimensional fitting, as the correlation curves and the original data are displayed together, and the user can easily perceive the trends of the process (specific to a data set), e.g the profile of cut quality features variations with feed rate as they reside on one sheet of paper (or on terminal display).

4) As the fitting accuracy is highly dependent on the distribution of data acquired from the database, and consequently on the amount of this data. It is expected that as more data is accumulated in the database, more accurate prediction will result as:

- The data collected will have better continuity within the area of correlation.

- This data can be more specific to the application, as more constraints can be imposed on search operations, such as; power and feed limits, specific level of cut quality and focus lens characteristics.

5) The method devised to restructure the original data of thickness, feed and power, to make it continuous and acceptable to the 3-dimensional fitting operation, is based on logical assumptions conceived from the nature of the CW laser cutting process, these are

- Continuous data can be created by performing two separate 2-dimensional fits between feed and material thickness, and between power and material thickness, with thickness as the independent variable.

- All data are related to one laser system, so that a logical correlation can be made between the three parameters.

This method proved to work satisfactorily when predicting for missing thicknesses both within and outwith the original data range.

6) The method of correlating between cut quality and process parameters can be readily extended to other parameters which are not pursued in this work (e.g real time measurements). This is relatively simple due to the modular construction of the prediction program, for example the addition of a new correlation (and its graphical output) only requires the insertion of one statement into the main program code.

7) A programme of factorially designed cutting experiments, using a pulsed YAG laser and a CW CO_2 laser, was executed, to provide a set of data as to enable the development and testing of; the database, data collection and prediction modules. The process parameters (input variables) studied for their effects on the process performance and on the cut quality features (output variables) were:

- Material type and thickness (CO_2 and YAG).
- Cutting speed (CO_2 and YAG).
- Assistant gas pressure (CO_2 and YAG).
- Laser power (CO_2).
- Laser pulse energy (YAG).
- Laser pulse repetition frequency (YAG).
- Nozzle stand-off distance (YAG).

The post-operation cut quality measurements were:

- Kerf width.
- Dross formation.
- Cut edge roughness.
- Cut edge taper.
- Heat affected zone.

The results of these experiments lead to the conclusions:

a) The width of surface oxidisation near the cut edges is poorly correlated with the width of subsurface heat affected zone. That is the temperature rise of the material adjacent to kerf does not relate to the oxidisation process. Therefore the oxidisation margin should not be considered as representing HAZ; a microstructure examination is needed for this measurement.

b) In pulsed laser cutting, the cut roughness, generally, increases with feed rate, as longer serration pitch and greater differences between peaks and valleys results due to the removal of larger volume of metal during each pulse (fig 7.1). Therefore, a higher PRF range or slower feed rate is required for smooth cuts.

c) Cutting of non-ferrous metals with pulsed YAG lasers can result in considerable dross formation at the rear surface. This defect tends to diminish with higher feed rate, higher pulse energy, shorter stand-off distance and higher assistant gas pressure.

Feed rate proved to have the most significant effect on dross formation, this can be explained by that; at higher feed, a greater proportion of the molten metal is ejected from the top surface of the kerf. This is supported by observations made when cutting at high feed rate, a more dense metal spatter cone is formed during the power pulse at the top surface, and wider dross layer is accumulated on the top surface of the material around the cut edges.

This can be explained, with the aid of fig 7.1, as:

- At a high feed rate a smaller peak-to-peak gap between the two cut edges will result (fig 7.1-a), this is causing a higher vapour pressure to build up within the kerf, consequently greater expulsion occurs from the upper surface layers of the melt pool. This cannot happen at slow speeds due to vapour pressure dissipiation from the open side of the kerf.

- Cutting at slow speed results in using the low power intensity zone of the laser beam (fig 7.1-b), which results in poorer heating, weaker metal ejection and perhaps higher viscosity, and the molten metal will resolidify more rapidly at the rear edge.

The above discussion implies that dross free cutting of non-ferrous metals with pulsed lasers, can be achieved

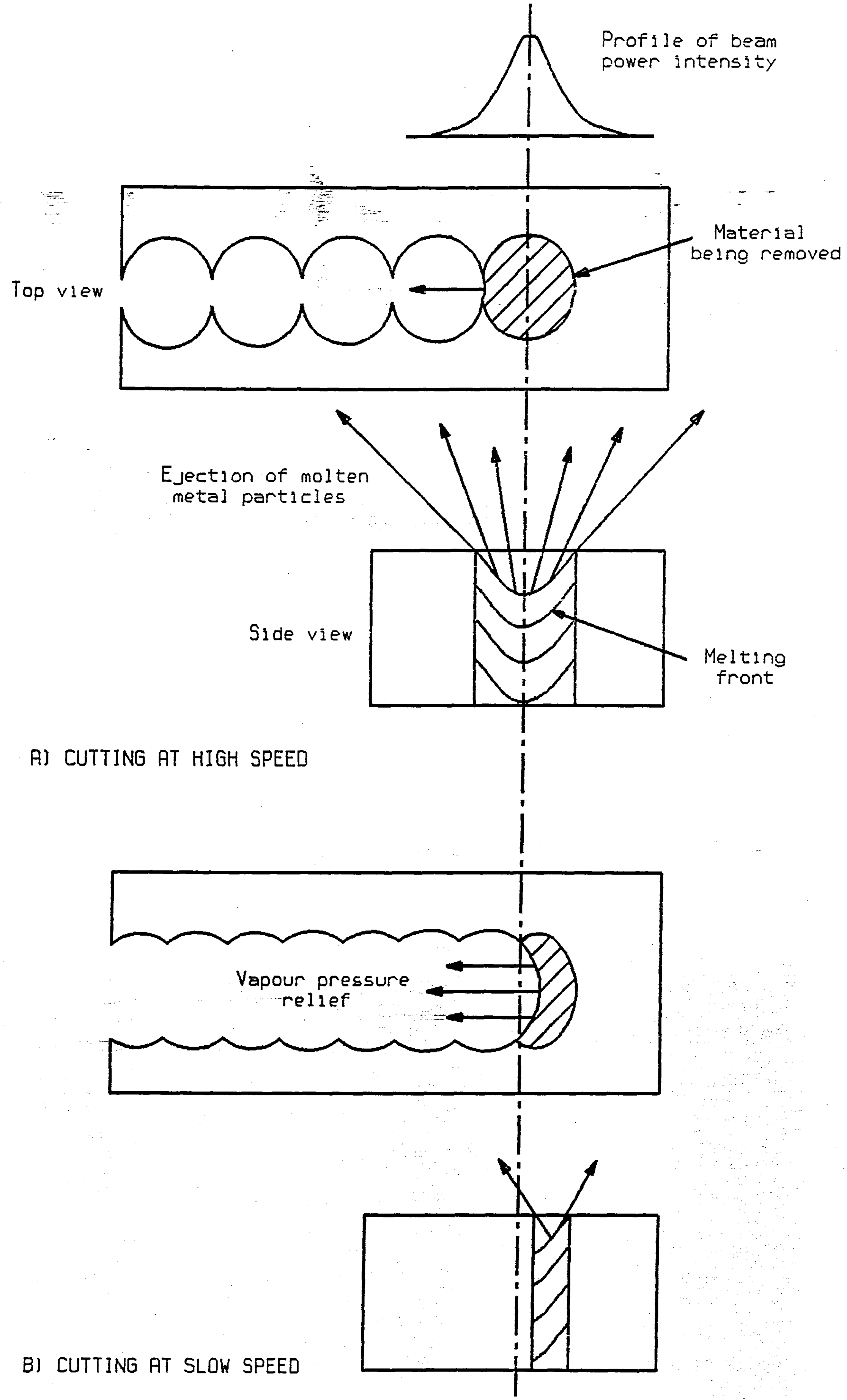


FIG 7.1 : MOLTEN METAL EJECTION IN CUTTING WITH PULSED LASER

using the highest possible cutting speed, and a sacrifice in cut smoothness will be incurred.

d) Increasing pulse energy (from 3.2 to 4.8 Joule) resulted in an increase of 20 to 40% in kerf width in all materials, due to increased metal removal. Though, this has insignificant effect on increasing maximum achievable cutting speed compared with PRF effect, as the correlation between maximum feed and PRF is positive and almost linear.

e) The effect of stand-off distance on the cutting process performance and cut quality is considerably more marked with the YAG than with the the CO_2 laser. This is because the YAG laser is highly sensitive to variations in optical stand-off distance as the depth of focal field is extremely short (120 microns) which implies large variation in focus diameter (and power intensity) with stand-off.

It is, therefore, necessary to select a stand-off distance and focusing lens focal length for a specific application that takes into account the precision of the material surface and the required level of power intensity (according to material type and thickness).

f) Dross free cutting of stainless steel can be achieved using a CW CO_2 laser with a feed rate less than 25% of the maximum achievable. The effect of slow cutting speed is that high rate oxidisation of the molten metal occurs, this seems to reduce the melt viscosity and cohesion to the base metal which produces readily removable oxides.

g) Cutting performance in polymers with CW CO_2 laser, exhibited quite stable behaviour, this could be due to the simplicity of material removal mechanism from the kerf, which is removed as vapour due to the low boiling temperature of these materials compared with metals. This is providing good opportunities for reasonably accurate modeling and prediction of cutting parameters both mathematically and empirically.

h) In general, using a pulsed YAG laser provided the benefit of producing high power intensity which enabled the cutting of a wide range of metals, with reasonable speed (up to 250mm/min) compared with its average power of 50W, and thickness up to 3mm (mild steel). This laser cannot cut polymeric materials due to their tranperency to the YAG wavelength.

The CO_2 laser (of 200W) cannot cut metals of high reflectivity and high conductivity, due to the low power intensity (as it is CW and emits a long wavelength). However, this laser is fairly effective at cutting polymers with good cut quality. Therefore, utilising a pulsed CO_2 laser for cutting can be an optimum choice for manufacturing flexibility and cutting efficiency.

The stages of the Tasking and Control System which have been implemented, indicate the viability of the approach. This is because; the underlying techniques are relatively simple to implement using standard computer hardware and software, the system does not demand extensive computer processing and can produce reasonably accurate process control strategies within a short time. System modularity provides flexibility and makes it a potential method for other types of laser based manufacturing systems, by adapting the individual modules. The future incorporation of in-process monitoring and control, and the use of artificial intelligence technology for self-learning, will make this system a practical and comprehensive solution to the problem of controlling laser based manufacturing processes.

APPENDIX 2A

CODE LISTING OF THE INTERPRETOR FOR C-TAPE POST PROCESSOR

```
******************************************************

This is a POST-PROCESSOR INTERPRETOR for the HEIDENHAIN
  controller type TNC-155A on the Laser Manufacturing
  system in the Department of Mechanical Engineering,
  the University of Glasgow, written by M.A. HUSSAIN.

******************************************************
```

- FORMATTING OUTPUT FILES

```
$POST=[
SYSTEM 6.1 0 501 30
ARC .001 1000
DFORMAT 2 1 0 1 0 3 0 0 1
DWORD 2 1 1 0 1 0 0 : :
TAPE N6 M2 G2 A2 B2 C2 D2 E2 P2 Q2 S3  T2 L4 X52 Y I52 J  F4 H3

**********************************
G    AS G.1         G00,G01
A    AS G.2         G17,G18,G19
B    AS G.3         G70,G71
C    AS G.4         G54
D    AS G.5         G02,G03
E    AS G.6         G50,G04
P    AS G.7         G90,G91,G98
Q    AS G.8         G40

N    Is a block number in the tapefile
M    Is a miscellaneous code
S    Is the spindle speed (laser beam power)
T    Is the tool number
L    Is the lable number for Subroutine definition.
     When it is (0),it is the end of a Subroutine.
F    Is the Feed Rate of the M/C.
****************************************
```

TAPE FILE FORMAT (CNC CODE)

```
FORMAT $S 3 1 0 1 0 3 0 0 -1
FORMAT $F 4 1 0 1 3 3 0 0 -1
FORMAT $X 52 2 1 1 3 3 0 1 -1
FORMAT $Y 52 2 1 1 3 3 0 1 -1
FORMAT $I 52 1 1 1 3 3 0 1 -1
FORMAT $J 52 1 1 1 3 3 0 1 -1
FORMAT $N 5 1 0 1 1 1 0 0 -1
FORMAT $L 4 1 0 1 3 3 0 0 -1
FORMAT $H 3 1 0 1 3 3 0 0 -1
CANCEL $G.1 -$X -$Y
CANCEL $C -$X -$Y
WORD $H 0 1 4 0 1
COUNT $N 10 10 9999 2
ENABLE 0 MM CMND CTCO NEWS ARCT TFRO
```

```
- PRINT FILE FORMAT
-
   PRINT HP 1
   LINK $H $HP
   COLUMN $H 80
   COLUMN $N 1 $G 10 $X 20 $Y 30
   COLUMN $I 50 $J 60  :  :
   COLUMN $F 10 $S 20
   ADDRESS $A G
   ADDRESS $B G
   ADDRESS $C G
   ADDRESS $D G
   ADDRESS $E G
   ADDRESS $P G
   ADDRESS $Q G
   ADDRESS $H :*:

-
-          ************************************
- START OF PROCESSING
-
   $PART=[
   #WRITE      ****************************************************
   #WRITE
   #WRITE      This is a "CTAPE" POST-PROCESSOR for the"HEIDENHAIN"
   #WRITE      Controller Type TNC-155A of the Laser Manufacturing
   #WRITE      System in the Department of Mechanical Engineerig,
   #WRITE                   University of Glasgow
   #WRITE
   #WRITE      ****************************************************
   #WRITE
   #FORMAT
   #WRITE      ***   The GNC part number is $TEXT   ****
   #FEND
   #WRITE
   #WRITE      *******************************************
   MESSAGE 0 :$TEXT G71 * ]
   COMM 0 -10 : PROFTYP          DISPLSMNT
   OUT
   COMM 0 -10 :---------       -----------
   OUT

-
-          ************************************
- COMMENTS TO BE PASSED TO OUTPUT FILES
-
   $COLO=[MESSAGE 0 :$TEXT *
   OUT]

-
-          ************************************
- PROGRAMED M/C STOP
-
   $STOP=[SET $M 6
   OUT
   COMM 0 -10 :/ BEAM POWER OFF /:
-  OUT
-  ]

-
```

```
             **************************************
SET LASER POWER

   $SPIN=[SET $M 4
   OUT
   COMM 0 -10 :/ BEAM POWER UP /:
   OUT
   ]

             **************************************
POSITIONING

   $RAPI=[SET $G.1 0 $X (R1) $Y (R2)  OUT]
   $FROM=[SET $A 17 $G.1 0 $P 90 $Q 40 $X (R1) $Y (R2)
   OUT]
   $GOTO=[
   SET $G.1 1 $X (R1) $Y (R2) -$F (W1)
   OUT
   #FORMAT
   W111 (R4)
   COMM 0 -10 : LINEAR          #W111/-3
   #WRITE LINEAR INTERPOLATION OF #W111/-3 MM
   #FEND
   OUT
   ]
   $FPM=[ W1 (R1)
   OUT]
   $GOHO=[$RAPI]

             **************************************
   CIRCULAR INTERPOLATION

   $ARC=[
   SET $D #IF R9=-1 2 #ELSE 3
   SET $X (R1) $Y (R2) $I (R4) $J (R5) -$F (W1)
   OUT
   #FORMAT
   W112 (R11)
   COMM 0 -10 : CIRCLE          #W112/-3
   #WRITE CICULAR INTERPOLATION OF #W112/-3 MM
   #FEND
   OUT]

            **************************************
SUBROUTINES

   $SUB=[#IF R5=1 #GOTO L1
   #IF R5=-1 #GOTO L2
_L1
   SET $P 98 $L (R4) OUT
   SET $G.1 0 $X (R1) $Y (R2) OUT
   #GOTO L3
_L2
   SET $G.1 0 $X (R1) $Y (R2)  OUT
   SET $P 98 $L 0
   #GOTO L3
```

```
_L3
   OUT]

-          *************************************
- CALLING SUBROUTINES
-
   $CALL=[
   #IF R5=1 #GOTO L1 #ELSE #GOTO L2
_L1
   SET $C 54 $X (R1) $Y (R2)  OUT
   ENABLE 0 GOTO GOHO SPIN DWEL FRO STOP COLO COOL RAPI ARC PPTR
   SET $L (R4) OUT
   #GOTO L3
_L2
   ENABLE 1 GOTO GOHO RAPI SPIN STOP DWEL COLO COOL FRO ARC PPTR
   SET $G.1 0 $X (R1) $Y (R2) OUT
_L3 OUT]
           *************************************
- TOOL LOADING
-
   $LOAD=[
   SET $T 1
   OUT]

-             *************************************
- TRANSFORMATIONS
-
   $PPTR=[
   SET $C 54 $X (R1) $Y (R2)
   OUT
   ]

-          *************************************
- DWELLING
-
   $DWEL=[SET $E 4 SET $F (R1) OUT]
   EL
   ]
          *************************************
- END OF OUTPUT FILE
-
   $END=[
   #WR
   $TEXT=#QUERY ENTER GNC PART NUMBER>
   #WR
   #WR   -------------------
   SET $M 5
   OUT
   MESSAGE 0 :N9999 $TEXT G71 *]

-          *************************************
- END OF INTERPRETOR
- ************************************************************
```

APPENDIX 3A

SQL/RT DATABASE USER INTERFACE

The user can work with the SQL/RT in three ways:

1) Using the Interactive Command Interface. The high level SQL data language statements are typed directly at the terminal, and are interpreted by the database software. An on-line help facility for inexperienced users is provided.

This interface requires a user who has some previous experience with databases. It is capable of making all the data definition operations such as: creating, altering, indexing and deleting tables and views. All relational and non-relational operations of data manipulation can be done using this interface, e.g retrieve, update and insert . Also, control commands to organise the various activities of the system can be entered such as the organisation of a report form. Table 3A-1 contains a summary of all the commands which can be used with this interface (ref 3A.1).

2) Using the Easy SQL/RT interface. This is a program which provide a simplified method of working with an SQL/RT database. This uses a system of menus, panels and boxes that appear on the display, in which the user can define

and manipulate data. This method is aided by a detailed help facility, and is intended for inexperienced users. Full description of this interface is in ref 3A.2.

This interface is more limited than the previous one. Although, this interface proved to be more efficient for inserting large quantities of data (see Appendix 3C). Table 3A-2 contains a comparitive list of limitations of the two types.

3) Using SQL statements embeded in an application program. This method was used in the implementation of the data collection program. Specially constructed SQL statements were embeded within an application program written in the "C" programming language. These SQL statements are then executed at run time.

REFERENCES

REF. NO.	DETAILS
3A.1	IBM UK LTD., SQL/RT Interactive Command Interface User's Guide.
3A.2	IBM UK LTD., Easy SQL/RT Interface User's Guide

SQL Commands

Command	Description
ALTER TABLE	Increases the size or number of columns in a table.
COMMIT WORK	Causes changes in the data base to take effect.
CREATE INDEX	Builds and maintains an index which helps SQL/ RT Data Base locate information more rapidly.
CREATE SYNONYM	Defines a convenient name to refer to a table or view.
CREATE TABLE	Defines a table for storing information.
CREATE VIEW	Defines an alternative perspective on a table or tables.
DELETE	Removes rows from a table.
DROP INDEX	Removes a previously created index on a table.
DROP SYNONYM	Drops a previously defined synonym.
DROP TABLE	Removes a table from the data base and deletes all its rows.
DROP VIEW	Removes a view from the data base.
FROM	(clause) Identifies the table(s) to work with.
GRANT	Allows other users access to your tables and views.
GROUP BY	(clause) Divides a table into groups of rows. based on the values in a column.
HAVING	(clause) Selects which groups to include in the results of a GROUP BY query.
INSERT	Adds new rows to an existing table.
LOCK TABLE	Temporarily prevents other users from modifying a table.
ORDER BY	Determines the order in which selected rows are presented.
REVOKE	Eliminates or reduces the access previously granted to other users.
ROLLBACK WORK	Cancels changes you have made to the data base since the last COMMIT WORK command.
SELECT	Retrieves information from a table.
SELECT FOR UPDATE	Temporarily prevents other users from modifying rows in a table.
UPDATE	Changes values in a table.
VALIDATE INDEX	Checks the integrity of an index.
WHERE	(clause) Specifies search conditions to select rows in a query.

TABLE 3A-1 : SQL/RT DATABASE COMMANDS FOR THE INTERACTIVE COMMAND INTERFACE (PART 1)

SOURCE : SQL/RT INTERACTIVE COMMAND USER'S QUIDE (IBM UK LTD)

Control Commands

Command	Description
APPEND	Adds text to the right of the current line in the command buffer.
BREAK ON	Breaks the rows of a query result into groups.
BTITLE	Places a title at the bottom of each report page.
CHANGE	Substitutes new text for old text in the current line of the command buffer.
CLEAR	Clears the screen or erases the contents of a command buffer.
COLUMN	Changes the way a column is displayed.
COMPUTE SUM OF	Computes column subtotals for each group of rows.
DEF	Assigns a value to a substitution variable.
DEL	Deletes the current line from the command buffer.
DESC	Gives a brief description of the columns in a table.
EXIT	Leaves the Interactive Command Interface program and returns to the operating system.
HELP	Shows information about SQL or control commands.
INPUT	Adds a new line after the current line in the command buffer.
LIST	Displays the current SQL statement in the command buffer.
RUN	Executes the current SQL statement in the command buffer.
SAVE	Stores the current SQL statement in a file.
SET	Changes the Interactive Command Interface environment.
SHOW	Shows the status of the Interactive Command Interface environment.
SPOOL	Transfers your work with the Interactive Command Interface to a file or printer.
START	Retrieves and executes SQL and control commands from a file.
TTITLE	Places a title, page number, and today's date at the top of each report page.
UNDEFINE	Cancels a previous assignment of a value to a substitution variable.

TABLE 3A-1 : SQL/RT DATABASE COMMANDS FOR THE INTERACTIVE COMMAND INTERFACE (PART 2)

SOURCE : SQL/RT INTERACTIVE COMMAND USER'S QUIDE (IBM UK LTD)

Item	SQL/RT Data Base Limit	Easy SQL/RT Limit
Tables in a data base	No limit*	
Rows in a table	No limit*	
Columns in a table	254	
Characters in a row	60,960	
Characters in a character field	240	
Digits in a number field	105	
Significant digits in a number field	40	
Range of values in a date field	1-JAN-4712 BC to 31-DEC-4712 AD	1-JAN-1900 AD to 31-DEC-1999 AD
Indexes allowed on a table	No limit*	
Indexes automatically created on a table	0	1
Tables or views joined in a query	No limit*	3
Levels of nested subqueries	16	0
Characters in a table name	11	
Characters in a view, query, or report name	11	
Characters in a column name	135	30
Characters in a comment field	N/A	80

*No enforced limits; however, practical limits may vary according to system configuration.

TABLE 3A-2 : SQL/RT DATABASE LIMITATIONS

SOURCE : EASY SQL/RT USER'S QUIDE (IBM UK LTD)

APPENDIX 3B

PHYSICAL STRUCTURE OF THE LASER CUTTING DATABASE TABLES IN SQL/RT

MATERIAL TABLE

****

COLUMN NAME	TYPE OF DATA	WIDTH	STATUS	REMARKS
TYPE	character	20	optional	Type of material
CONST	character	25	optional	Significant Constituents
THICK	number	10	mandatory	Material thickness (microns)
SURFIN	number	5	optional	Surface roghness (nanometre)
ID	number	9.2	mandatory	Unique identification code
CL	number	6	mandatory	Code of laser system
CD	number	6	optional	Code of beam delivery system
SRFTRTMT	character	25	optional	Material surface treatment

SAMPLE OF THE TABLE

TYPE / ID	CONST / CL	CD	SRFTRTMT	THICK	SURFIN
BS970EN3B	.25C,1.Mn,.35Si			1600	825
85.01	2	2	cold rolled		
BS970EN3B	.25C,1.Mn,.35Si			1600	825
86.01	2	2	cold rolled		
BS970EN3B	.25C,1.Mn,.35Si			1600	825
87.01	2	2	cold rolled		
BS970EN3B	.25C,1.Mn,.35Si			1600	825
88.01	2	2	cold rolled		
BS970EN3B	.25C,1.Mn,.35Si			1600	825
89.01	2	2	cold rolled		
BS970EN3B	.25C,1.Mn,.35Si			1600	825
90.01	2	2	cold rolled		
BS970EN3B	.25C,1.Mn,.35Si			1600	825
91.01	2	2	cold rolled		
BS970EN3B	.25C,1.Mn,.35Si			1600	825
92.01	2	2	cold rolled		

LASER SYSTEM TABLE
****

COLUMN NAME	TYPE OF DATA	WIDTH	STATUS	REMARKS
CL	number	6	mandatory	Code of laser system
LSRTYP	character	15	optional	Type of laser
PULSED	character	3	mandatory	Pulsation ability (YES or NO)
BMDVRGNC	number	3	optional	Beam divergence (milliradian)
MODQLTY	number	3	optional	Mode quality as percentage
WSTSIZ	number	5	optional	Beam waist diameter (micron)
MAXPWR	number	6	mandatory	Maximum acievable power (Watt)
MINPWR	number	6	optional	Minimum conrollable power (Watt)

SAMPLE OF THE TABLE

CL	LSRTYP	PUL	BMDVRGNC	MODQLTY	WSTSIZ	MAXPWR	MINPWR
1	CO2	NO				1500	
2	Nd/YAG	YES				60	
3	CO2	no	3		11000	500	0

BEAM DELIVERY SYSTEM TABLE
****

COLUMN NAME	TYPE OF DATA	WIDTH	STATUS	REMARKS
CD	number	6	mandatory	Code of beam delivery system
BMDSTNS	number	3.1	optional	Beam route distance (metre)
LNSTYP	character	15	optional	Type of focusing lens
FCLENTH	number	3	optional	Focal length (mm)
FCSDMTR	number	6	optional	Focus diameter (micron)
INTNSTY	number	9	optional	Beam power intensity (kW/cm)
LNSMTRL	character	20	optional	Lens material
PLRZN	character	5	optional	Polarisation (C or L)
NZLDMTR	number	5	optional	Nozzle orifice diameter (micron)

SAMPLE OF THE TABLE

```
        CD    BMDSTNS LNSTYP                FCLENTH      FCSDMTR      INTNST'
---------- ---------- --------------- ---------- ---------- ----------
LNSMTRL                 PLRZN   NZLDMTR
-------------------- ----- ----------

         1             planoconvex             127          200        320(
ZnSe                                1200

         2           1 planoconvex              80
ZnSe                    L           1000

         3           3 planoconvex             150
ZnSe

3 records selected.

> spool of
```

PROCESS PARAMETERS TABLE

COLUMN NAME	TYPE OF DATA	WIDTH	STATUS	REMARKS
ID	number	9.2	mandatory	Unique identification code
POWER	number	5	mandatory	Laser Power (Watt)
FEED	number	5	mandatory	Cutting speed (mm/min)
GASTYP	character	15	optional	Assistant gas type
GASPRSR	number	5	optional	Assistant gas pressure (millibar)
STDOF	number	5	optional	Nozzle stand-off (micron)
FCSDPTH	number	4	optional	Focus location (micron)
PLSTAT	character	6	mandatory	Laser pulsing (PULSED / CW)
PRF	number	3	optional	Pulse repetition frequency (Hz)
PLSWDTH	number	5	optional	Pulse duration (microsecond)
PLSENRGY	number	6	optional	Pulse energy (millijoule)

SAMPLE OF THE TABLE

ID	POWER	FEED	GASTYP	GASPRSR	STDOF	FCSDPTH	PLSTAT	PRF	PLSWDTH	PLSENRGY
85.01	48	150	O2	1000	500	1	PULSED	15	1000	3200
85.02	48	150	O2	3000	500	1	PULSED	15	1000	3200
85.08	48	250	O2	3000	800	1	PULSED	15	1300	3200
85.20	48	140	O2	3000	1000	500	PULSED	20	900	2400
200.01	120	150	O2	3000	1000	500	CW			
201.01	120	225	O2	3000	1000	500	CW			

REAL-TIME CONTROL TABLE

****

COLUMN NAME	TYPE OF DATA	WIDTH	STATUS	REMARKS
ID	number	9.2	mandatory	Unique identification code
INCDNT	number	5	optional	Incident power (Watt)
RFLCTD	number	5	optional	Reflected power (Watt)
SRFTMP	number	4	optional	Material surface temperature (C)

SAMPLE OF THE TABLE

ID	INCDNT	RFLCTD	SRFTMP
11	500	80	1450

CUT QUALITY FEATURES TABLE

****

COLUMN NAME	TYPE OF DATA	WIDTH	STATUS	REMARKS
ID	number	9.2	mandatory	Unique identification code
KERF	number	5	optional	Kerf width (micron)
HAZ	number	5	optional	Heat affected zone (micron)
RFNESS	number	6	optional	Cut roughness (nanometre)
TAPER	number	5	optional	Cut taper (micron)
DROSIZ	number	5	optional	Dross height (micron)
QLTYGRD	number	2	optional	Quality grade (%)
SPDGRD	number	2	optional	Cutting speed grade (%)

SAMPLE OF THE TABLE

ID	KERF	HAZ	RFNESS	TAPER	DROSIZ
QLTYGRD	SPDGRD				
200.01	300	58	2250	30	10
	50				
201.01	300	66	2000	5	10
	75				
202.01	320	74	2000	15	10
	99				
203.01	370	90	3500	-80	20
	50				
204.01	380	110	2500	-100	40
	75				
205.01	385	116	3000	-100	30
	99				

MISCELLANEOUS INFORMATION TABLE

COLUMN NAME	TYPE OF DATA	WIDTH	STATUS	REMARKS
ID	number	9.2	mandatory	Unique identification code
INFSRC	character	15	optional	Information source
CRTRDS	number	5	optional	Critical contour radius (micron)
CRTLNTH	number	5	optional	Critical cut length (micron)
LOOP	character	3	optional	Looping necessity at corners (YES/NO)
REFCOD	character	6	optional	User reference code

SAMPLE OF THE TABLE

ID	INFSRC	CRTRDS	CRTLNTH	LOO	REFCOD
80.01	lit				b-26
81.01	lit				b-26
82.01	lit				b-26
82.08	own				
83.01	lit				b-26
84.01	lit				b-26
85.01	own				
88.2	own				
98.02	own				
99.02	own				
122.02	own				
129.01	own				
130.01	own				
160.01	own				
161.01	own				

PRODUCTION INFORMATION TABLE

****

COLUMN NAME	TYPE OF DATA	WIDTH	STATUS	REMARKS
ID	number	9.2	mandatory	Unique identification code
PRTNO	character	10	optional	Part program number
BTCHNO	character	10	optional	Batch number
PRCSNO	character	10	optional	Process number
CUTNO	number	5	optional	Cut number
OPERATOR	character	10	optional	Operator name
PRCSDAT	date	N/A	optional	Process date (31-DEC-99)

SAMPLE OF THE TABLE

ID	PRTNO	BTCHNO	PRCSNO	CUTNO	OPERATOR	PRCSDAT
160.01			029	24	M.H , G.F	09-MAR-88
165.01			029	22	M.H , G.F	09-MAR-88
166.01			029	7	M.H , G.F	09-MAR-88
169.01			051	1	M.H , C.M	15-JUL-88
183.01			051	12	M.H , C.M	15-JUL-88
184.01			052	1	M.H , C.M	15-JUL-88

APPENDIX 3C

METHOD OF DATA INSERTION INTO THE DATABASE

After the required tables were created, two kinds of data were inserted into them: Data extracted from literature and experimental data from the current project. Because of delays in the experimental programme, literature data was used to test the modules of the planning system. The amount of data entered into the database was considerable, and the author found it more efficient to the use Easy SQL/RT interface for this task. This was due to the following facilities:

- The multi-tasking capability of the IBM 6150 computer system, which is explained in section 3.3.

- The multi-user feature of the SQL/RT database system.

- The multiple row insertion mode of the easy SQL/RT.

A number of virtual screens were opened on the console display, and the Easy SQL/RT program was run on each virtual screen with different tables called for insertion. Each table was opened using the multiple insertion mode (in contrast of the single insertion mode). The table was displayed as columns occupying the full lengh of the display, headed by its name, ready for insertion.

Five advantages are incurred by this method:

a) Fast data insertion, due to the ability to enter data as normal text without special commands, and the ability to move between the rows and columns is using single key strokes.

b) A facility to copy a complete row to the next line with a single key. This is useful when the next row to be inserted is slightly different from the one above. The latter can be copied and edited.

c) The data inserted during a session can be edited any time while that session is current, simply by overwriting. The other types of interface need a separate update process.

d) When inserting related data into several tables, the multiple screen facility allow for cross checking between tables.

e) Two types of help is available to the user. One is the system help which describes how to perform various operations. The other is the owner help comments about each column, which were written when the table was created.

APPENDIX 5A

THE PRE-COMPILER CODE LISTING OF THE DATA COLLECTION PROGRAM

```
/* -------------------------------------------------------------------
   "collect" is a program  to select laser cutting parameters from
    the existing laser cutting database system, according to user
                        recommendations.
   ------------------------------------------------------------------

#include <stdio.h>
#include <ctype.h>
EXEC SQL BEGIN DECLARE SECTION;

/* ---------------- Output parameters -------------------------  */

int   thick, surfin, cl, cd, power, feed, gasprsr, stdof ;
int   prf, plswdth, plsenrgy ;
int   kerf, drosiz, haz, rfness, qltygrd, spdgrd ;
int   bmdvrgnc, modqlty, wstsiz, maxpwr, minpwr, bmdstns, fclenth ;
int   nzldmtr, fcsdmtr, intnsty, crtrds, crtlnth , incdnt, rflctd ;
int   srftmp ;
long float    id ;

/* ------The following two variables are considered as RAW DATA
    CODE 06-APP.D-SQL/RT prgrammer's guide so that they can handle
    NEGATIVE values from the database sys. ---------            */

char  taper[6] ,fcsdpth[6] ;
VARCHAR   const[20] , type[20], gastyp[20] , lsrtyp[15], pulsed[3] ;
VARCHAR   lnstyp[15], loop[3] ,operator[10] , prcsdat[9], plstat[6];
VARCHAR   infsrc[20] , refcod[6], prtno[10], btchno[10], prcsno[10] ;

/* ---------------- Input parameters --------------------------  */

int   ithick, isurfin, icl, icd, ipower, ifeed, igasprsr, istdof,ifcsdpth ;
int   ikerf, idrosiz, ihaz,irfness, iqltygrd, ispdgrd ,itaper ;
int   ibmdvrgnc, imodqlty, iwstsiz, imaxpwr, iminpwr, ibmdstns, ifclenth ;
int   inzldmtr, ifcsdmtr, iintnsty, icrtrds, icrtlnth , iincdnt, irflctd ;
int   isrftmp ;
long float   iid ;
VARCHAR   iconst[20] , itype[20], igastyp[20] , ilsrtyp[15], ipulsed[3] ;
VARCHAR   ilnstyp[15], iloop[3] ,ioperator[10] , iprcsdat[9],iplstat[6];
VARCHAR   iinfsrc[20] , irefcod[6], iprtno[10], ibtchno[10], iprcsno[10];

/* ----------------- General variables -----------------------  */

int    flg1, flg2, flg3, flg4, flg5, flg6 , nagcod, *fout ;
int    fdwt,krfwt,hazwt,drswt,rfwt,tprwt ;
char   slctyp[256], mtrl[56], mtrlcls[20] , stmt1[356] ;
char   stmt2[356], stmt3[356], stmt4[356],stmt5[356], stmt6[356] ;
char   stmt7[356], stmt8[356], stmt9[356], stmt10[356], stmt11[356] ;

   EXEC SQL END DECLARE SECTION ;
```

```
/* ------------- Open an SQL communication area ----------------  */

   EXEC SQL INCLUDE SQLCA ;

/* --------------- Start the main routine ----------------------- */

main()
{
   int    ansr ;

/* ------------- ask for database connection ----------------    */

 EXEC SQL CONNECT;

printf("\n                   program started\n\n\n\n\n\n");
printf("\n *****************************************************");
printf("\n                     *** collect ***");
printf("\n           is a program to select laser cutting");
printf("\n          parameters from a laser cutting database");
printf("\n      system according to the user's recommendations.");
printf("\n       Created by M.A.HUSSAIN, in the department of");
printf("\n     Mechanical Engineering, The University of Glasgow,");
printf("\n                          in 1987.");
printf("\n *****************************************************");
printf("\n\n\n\n");
  getansr:
        printf ("Options for input data !!!!\n");
        printf ("\n Interactive data input                  1 \n");
        printf ("\n input by datafile                       2 \n");
        rdn(" \n         enter option  >:",&ansr);
         if (ansr == 1 )  {printf("\n\n\n");       getkey();    }
   /*    if (ansr == 2 )        getfil();    */
           else    goto getansr;
}
/* ++++++++++++++++++++++  SUBROUTINES   ++++++++++++++++++++++ */

/* ----------- The user interrogation routine ----------------- */

getkey()
{
   int   *ls1, *ls2, *ls3, *ls4, *ls5, *ls6, *ls7, *ls8, *ls9 ;
   int   i1, i2, i3, i4, i5, i6  ;
   int   *fout, class, ans2, ans3, ans4, ans5 ,ans6, ans7 ;
   int   ans8, ans9 , ans10, ans12, ans14 ;
   char  ans1[3];
  getclass:
    printf("\n**Supply material class code from the following codes **");
    printf("\n\n1/mild-steel     2/stainless-steel  3/carbon-steel\n");
    printf("\n4/alloy-steel      5/special-steel    6/titanium&alloys\n");
    printf("\n7/aluminum&alloys  8/plastic          9/wood \n");
    printf("\n10/copper&alloys   11/nickel&alloys\n") ;

    rdn("\n            Type your class code >",&class);
    if(class < 1  || class >11 )    goto getclass;
    printf("\n");
```

```
gettype:

   printf("\n    ****************** INFORMATION *****************\n");
   printf("\n Types of materials which are subsets of your material");
   printf("\n class, will be shown. Check them for further queries.");
   printf("\n Press Ctrl/S to stop scrolling ,Ctrl/Q to continue.\n");
   lsl= rds("\n Hit RETURN when you are ready >",ans1);

/* ------- Proceedures to show available types of materials ------ */

  matype:

   switch (class) {

    case 1:
      printf("\n");
      strcpy(mtrl," mldst01");
      strcpy(mtrlcls,"MILD-STEEL")  ;
               break  ;

    case 2:
      strcpy(mtrl," stnst02");
      strcpy(mtrlcls,"STAINLESS-STEEL") ;
               break  ;
    case 3:
      strcpy(mtrl," crbnst03");
      strcpy(mtrlcls,"CARBON-STEEL") ;
               break  ;
    case 4:
      strcpy(mtrl," alyst05");
      strcpy(mtrlcls,"ALLOY-STEEL") ;
               break  ;
    case 5:
      strcpy(mtrl," spclst07");
      strcpy(mtrlcls,"SPECIAL-STEEL") ;
               break  ;
    case 6:
      strcpy(mtrl," ttnm08");
      strcpy(mtrlcls,"TITANIUM & ALLOYS") ;
               break  ;
    case 7:
      strcpy(mtrl," almnm20");
      strcpy(mtrlcls,"ALUMINUM & ALLOYS") ;
                break  ;
    case 8:
      strcpy(mtrl," plstc70");
      strcpy(mtrlcls,"PLASTICS") ;
               break  ;
    case 9:
      strcpy(mtrl," wood80");
      strcpy(mtrlcls,"WOOD") ;
               break  ;
    case 10:
      strcpy(mtrl," copr25") ;
      strcpy(mtrlcls, "COPPER & ALLOYS") ;
               break ;
```

```
    case 11:
      strcpy(mtrl," nikel15") ;
      strcpy(mtrlcls, "NICKEL & ALLOYS") ;
               break ;

                      }         /* the switch end */

      strcpy(slctyp," SELECT DISTINCT type FROM ");
      strcat(slctyp,mtrl);

      EXEC SQL WHENEVER SQLERROR GOTO  EREXIT;
      EXEC SQL WHENEVER NOT FOUND GOTO endl;

      EXEC SQL PREPARE S0 FROM :slctyp ;
      EXEC SQL DECLARE Q1 CURSOR FOR  S0 ;

      EXEC SQL OPEN Q1 ;
      for (i1=0; ;i1++)
       {
        EXEC SQL FETCH Q1 INTO :type;
        type.arr[type.len]='\0';
        printf("\n                    %s",type.arr );
        EXEC SQL COMMIT WORK ;
       }
      endl:
           printf("\n");
           EXEC SQL CLOSE Q1;

/* -------- checking with the user for material type ---------  */

   if (i1 >= 1)
     {
    rdn("\n Do you want to see the types again (1/y , 2/n) >", &ans2);
    if (ans2 == 1) goto matype  ;
     rdn("\n Is your material in these types (1/yes , 2/no) >" ,&ans3);
       switch (ans3)
            {
        case 1:
          ls2=rds("\nPrint your type . As shown above >", itype.arr );
          if (ls2 < 1)     goto matype;
              flgl = 1 ;            /* Material type exist    */
              break;
        default:
          printf("\n    ***I* You choosed the (no) option ***");
          ls2=rds("\nPlease supply your material type >",itype.arr);
          if (ls2 < 1)   {   strcpy(itype.arr , "NULL");
            printf("\n    ***I* Material type is set to (NULL) ***\n");}
              flgl = -1 ;     /* No type   */
             }
     }
     if(i1 < 1)

          {
       printf("\n    *** Sorry nothing was found ***\n");
       ls2=rds("\nPlease supply your material type >",itype.arr);
       if (ls2 < 1)  {    strcpy(itype.arr , "NULL");
```

```
          printf("\n    ***I* Material type is set to (NULL) ***\n");}
             flg1 = -1 ;         /* No type   */
            }
       itype.len = ls2 ;


/* ---------------- Material thickness query -----------------    */

   rdn("\n\n    Enter material thickness (in microns) >", &ithick);
   while(!ithick)
   rdn("\n ***W* You must enter the material thickness *** >",&ithick);

/* ---------------- Material constituents query ----------------    */

   printf("\n\nSupply the material constituents as(1.3C/.20 ...),\n");
   ls3=rds("or hit RETURN if unknown >", iconst.arr);
   if (ls3 < 1)  {   strcpy(iconst.arr, "NULL");
    printf("\n ***I* Matrial constituents is set to(NULL)***\n");
                 }
    iconst.len = ls3 ;
    printf("\nSupply the material surface finish (in microns),\n");
    rdn("or hit RETURN if unknown >", &isurfin) ;
    if (isurfin < 1)      isurfin = -1 ;

/* ---------------- laser machine particulars ---------------    */

   printf("\n\n     *************** INFORMATION **************");
   printf("\nNext queries are about your laser machine particulars.\n");
   printf("\n     Choose from the following options \n");

  getlsr:
    printf("\n    You know your laser machine code (CL)              1");
    printf("\n    You wish to see the exisiting types                2");
    printf("\n    you want to enter details individually             3");
    rdn("\n\n             Enter option code >", &ans4);
     if (ans4 <1 || ans4 >3)  goto getlsr ;

     switch (ans4){

  case 1:
        rdn("\nSupply the laser machine code (CL) >",&icl) ;
          if (icl < 1) {printf("\n***W* Improper answer ***");
            goto getlsr;}
      flg2 = 2 ;           /*  CL exist  */

  lsqry:

      printf("\n");     /* dummy line  */
      EXEC SQL WHENEVER NOT FOUND GOTO end30 ;
      EXEC SQL SELECT  lsrtyp, pulsed, maxpwr
           INTO   :ilsrtyp, :ipulsed, :imaxpwr
           FROM   lsrprms
           WHERE  cl= :icl ;
        ilsrtyp.arr[ilsrtyp.len]='\0' ;
        ipulsed.arr[ipulsed.len]='\0' ;
```

```
    if (toupper(ipulsed.arr[0]) == 'Y')    flg5=1  ;
    if (toupper(ipulsed.arr[0]) == 'N')    flg5=2  ;
    break;

case 2:

 printf("\n  CL    laser  Pulse  beam   mode   waist  max.   min.");
 printf("\n  code   type  stat.  diver. qlty.  diam.  power  power");
 printf("\n------  ------ -----  ------ -----  -----  -----  -----\n");
    EXEC SQL WHENEVER NOT FOUND GOTO end31;
    EXEC SQL DECLARE Q31 CURSOR FOR
      SELECT DISTINCT  cl,lsrtyp,pulsed,bmdvrgnc,modqlty,
                       wstsiz,maxpwr,minpwr
               FROM lsrprms;
    EXEC SQL OPEN Q31;
     for(i2=0; ;i2++)
         {
     EXEC SQL FETCH Q31 INTO  :cl, :lsrtyp, :pulsed, :bmdvrgnc,
                              :modqlty, :wstsiz, :maxpwr, :minpwr;
       lsrtyp.arr[lsrtyp.len]='\0';
       pulsed.arr[pulsed.len]='\0';
       printf("\n%6d  %6s  %5s  %6d  %5d  %5d  %5d   %5d",
              cl, lsrtyp.arr, pulsed.arr, bmdvrgnc, modqlty,
              wstsiz, maxpwr, minpwr);
         }

  end31:
      printf("\n");
      EXEC SQL CLOSE Q31 ;
      rdn("\nIs your machine within these types (1/y , 2/n) >",&ans5);
      if (ans5 == 1) { rdn("\nEnter the relevant CL code >",&icl);
        if (icl < 1) {printf("\n***W* Improper answer ***");
          goto getlsr;}
        flg2 = 2 ;      goto lsqry ; }
      else goto getlsr;

 case 3:

    flg2 = 1 ;

  lstype:

    ls4 =rds("\nSupply the laser type (Nd/YAG,CO2,..) >",ilsrtyp.arr);
    if (ls4 < 1)              goto lstype;
    ilsrtyp.len = ls4 ;
    while (!imaxpwr)
    rdn("\nEnter the maximum laser power (in Watts) >", &imaxpwr);

  getpls:

    rdn("\nIs it pulsed (1/YES , 2/NO)    ? >", &flg5);
    if (flg5 < 1 || flg5 >2)   goto getpls ;
    printf("\n***I* For the next items,hit RETURN if unknown ***\n");
    rdn("\nWhat is the beam divergence (in millirads.)",&ibmdvrgnc);
    rdn("\nEnter the mode quality (between 1 & 99) >", &imodqlty);
    rdn("\nEnter the beam waist diameter (in mm) >", &iwstsiz);
```

```
    rdn("\nEnter the minimum laser power (in Watts) >", &iminpwr);
    break;

 end30:

    printf("\n   ***W* No such laser  *** \n");
    goto  getlsr ;

 default:

    goto  getlsr ;
    }          /* the switch end   */

/* --------------- Specify the laser type --------------------- */

    if (flg5 ==1 )     strcpy(iplstat.arr, "PULSED");
    if (flg5 ==2 )     strcpy(iplstat.arr, "CW");

/* -------------- beam delivery system particulars ------------- */

   printf("\n\n     *************** INFORMATION **************");
   printf("\n   Next queries are about your beam delivery system.\n");
   printf("\n        Choose from the following options \n");

 getbeam:

   printf("\n    You know your beam delivery system code (CD)       1");
   printf("\n    You wish to see the exisiting types                2");
   printf("\n    you want to enter details individually             3");
   printf("\n *** Hit RETURN if you want to skip this section ***    ");
   rdn("\n\n             Enter option code >", &ans10);

 switch (ans10){

   case 1:

    rdn("\nSupply the beam delivery system code (CD) >", &icd);
   if (icd < 1) {printf("\n***W* Improper answer ***");
    goto getbeam;}
    flg3 = 2 ;          /*  CD exist  */
    break;

  case 2:

  printf("\n  CD    beamroute    lens      focal   focus   beam");
  printf("        nozzle");
  printf("\n  code  length(m)    type      length  dia.    intensity");
  printf("  diameter");
  printf("\n------  ---------  ----------  ------  ------  --------- ");
  printf(" --------\n");
     EXEC SQL WHENEVER NOT FOUND GOTO end32;
     EXEC SQL DECLARE Q32 CURSOR FOR
         SELECT DISTINCT  cd, bmdstns, lnstyp, fclenth, fcsdmtr,
                          intnsty,nzldmtr
                FROM  bmdlvry;
     EXEC SQL OPEN Q32;
```

```
      for(i3=0; ;i3++)
         {
         EXEC SQL FETCH Q32 INTO    :cd, :bmdstns, :lnstyp,:fclenth,
                                    :fcsdmtr, :intnsty, :nzldmtr ;

         lnstyp.arr[lnstyp.len]='\0';
         printf("\n%6d  %9d  %10s  %6d  %6d  %9d  %8d", cd, bmdstns,
                lnstyp.arr, fclenth, fcsdmtr, intnsty, nzldmtr );
        }

  end32:

     printf("\n");
     EXEC SQL CLOSE Q32 ;
     rdn("\nIs your system within these types (1/y , 2/n) >",&ans12);
     if (ans12 == 1) { rdn("\nEnter the relevant CD code >",&icd);
     if (icd < 1) {printf("\n***W* Improper answer ***");
         goto getbeam;}     flg3 = 2 ;            break;      }
         else goto getbeam;

  case 3:

    rdn("\nEnter the beam route distance (in m) >",&ibmdstns);
    ls6=rds("\nSupply the nozzle lens type >", ilnstyp.arr);
    if (ls6 < 1)   strcpy(ilnstyp.arr, "NULL");
    ilnstyp.len = ls6  ;
    rdn("\nEnter focal length of the nozzle lens (in mm) >",&ifclenth);
    rdn("\nEnter focus diameter of the focused beam (in microns)>",
          &ifcsdmtr);
    rdn("\nEnter the focused beam intensity(kW/cm.sqr) >", &iintnsty);
    if (!ibmdstns && !ifclenth && !ifcsdmtr && !iintnsty && ls6 < 1)
       flg3 = -1 ;          /* No beam delivery parameters */
    else flg3 = 1 ;        /* beam delivery parameters exist */
                 break;

  default:

    printf("\n***I* you choosed to skip this section ***\n");
    flg3 = -1 ;/* No beam delivery information */
                 }          /*  the switch end   */



/* ------- Asking for the max. feedrate of the cutting m/c ------- */
   while (!ifeed)
    rdn("\nEnter the max. feedrate the ctting m/c can achieve(mm/min) >"
          , &ifeed);

/*------------------ Cut quality queries ----------------------    */

    printf("\n ****************** INFORMATION ******************\n");
    printf("\n Next queries are about the priorities of the") ;
    printf("\n  desired cut quality features expressed as") ;
    printf("\n    percentages ranging between 1 & 99 .");
    flg4= 0 ;    /* Initiation  */
```

```
getqlty:

  rdn("\nHit RETURN to skip this section.Or type (1) to proceed >",
      &ans14) ;

  if (ans14 != 1)   {   flg4 = -1 ;  /* No quality queries */
     printf("\n***I* You chossed to skip this section ***\n");
     fdwt = 99 ;     }

  if (ans14 == 1)
  { flg4 = 1 ;
   rdn("\n Priority  of min. kerf is               >", &krfwt);
   rdn("\n Priority weight of min. H.A.zone is  >", &hazwt);
   rdn("\n Priority of min. cut roughness is     >",  &rfwt);
   rdn("\n Priority of min. dross size is        >", &drswt);
   rdn("\n Priority of min. taper is             >", &tprwt);
   rdn("\n Propose a priority for max. feed      >",  &fdwt);
   if (!krfwt,!hazwt,!rfwt,!drswt,!tprwt) {
      flg4 = -1 ;     fdwt = 99 ;          }
  }

/* ------------- write the numerical data to a file ----------- */

          datfil();

/* ------------- go to the data retrieval routine ------------ */

          prmsqyl();
   EREXIT:
         rprteror();
         return ;
 }

/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ */

/* ------------- The data retrieval routine ----------------- */

prmsqyl()

       {

 char   cthk[8], cpwr[8], cfeed[8], ccl[8], ccd[8], crfness[8] ;
 char   chaz[8],ckerf[8],cdrosiz[8],ctaper[8],cqlty[8],cspd[8] ;
 char   stmt01[80], stmt02[80] ,stmt03[80],stmt04[80] ;
 char   stmt05[80],stmt06[80],stmt00[80];
 char   crngA1[7],crngA2[7],crngB1[7], crngB2[7], crngC1[7] ;
 char   crngC2[7], crngD1[7], crngD2[7] ;
 int    i, j, *input, cod , stage ;

   cod=0 ;   /* initiating the thickness code   */
```

```
/* ---------- Delete if the view is existing ----------  */

    EXEC SQL WHENEVER SQLERROR CONTINUE  ;
        EXEC SQL EXECUTE IMMEDIATE
                " DROP VIEW tmptbl " ;

    EXEC SQL WHENEVER SQLERROR GOTO EREXIT1 ;

    input=fopen("nmfil.dat","r") ;
    fscanf(input,"%7s %7s %7s", cthk, cpwr, cfeed) ;
    fscanf(input,"\n%7s %7s", ccl, ccd) ;
    fscanf(input,"\n%7s %7s %7s %7s %7s %7s %7s",
           ckerf, chaz, crfness, cdrosiz, ctaper, cqlty, cspd) ;
    fscanf(input,"\n%7s %7s %7s %7s %7s %7s %7s %7s" ,
           crngA1,crngA2,crngB1,crngB2,crngC1,crngC2,crngD1,crngD2);

/* ------------ Composing preliminary SQL statments --------- */
    strcpy(stmt00,"CREATE VIEW tmptbl AS ");
    strcpy(stmt01,"SELECT  thick,power,feed,gasprsr,stdof,");
    strcat(stmt01,"prf,plswdth,plsenrgy,");
    strcat(stmt01,"kerf,haz,taper,rfness,drosiz ");
    strcpy(stmt03," FROM  prcprms, ");
    strcpy(stmt04,", cutqlty WHERE prcprms.id= ") ;
    strcpy(stmt05,".id AND prcprms.id=cutqlty.id AND thick = ") ;
    strcpy(stmt06,".id AND prcprms.id=cutqlty.id AND thick BETWEEN ");
    printf("\n         Starting at ");

/* ---------------- Directing the search -------------------  */

    if ( flg1==1 && flg2==2 && flg3==2 )         goto stage1 ;
    if ( flg1==1 && flg2==2 && flg3 !=2 )        goto stage2 ;
    if ( flg1==1 && flg2 !=2 && flg3==2 )        goto stage3 ;
    if ( flg1==1 && flg2 !=2 && flg3 !=2 )       goto stage4 ;
    if ( flg1 !=1 && flg2==2 )                   goto stage5 ;
    goto stage6 ;       /*   for all other cases    */

/* ------------------------------------------------------------- */

 stage1:
   printf("\nStage 1\n");
   fout=fopen("lsr.dat","w");
   filhd1(cod,1);
   filhd2();
     strcpy(stmt1,stmt01);
     strcat(stmt1,stmt03);              strcat(stmt1,mtrl)   ;
     strcat(stmt1,stmt04) ;             strcat(stmt1,mtrl)   ;
     strcat(stmt1,stmt05) ;             strcat(stmt1,cthk)   ;
     strcat(stmt1," AND type = '") ;    strcat(stmt1,itype.arr);
     strcat(stmt1,"'") ;                strcat(stmt1," AND cl=");
     strcat(stmt1,ccl) ;                strcat(stmt1," AND cd=") ;
     strcat(stmt1,ccd) ;
    EXEC SQL WHENEVER NOT FOUND GOTO EndA1;
    EXEC SQL PREPARE S1 FROM :stmt1 ;
    EXEC SQL DECLARE CA1 CURSOR FOR  S1 ;
    EXEC SQL OPEN CA1 ;
```

```
    for (i=0; ;i++)
    {

   EXEC SQL FETCH CA1 INTO  :thick,:power,:feed,:gasprsr,:stdof,
                            :prf,:plswdth,:plsenrgy,:kerf,
                            :haz,:taper,:rfness,:drosiz ;
    datout();
    }
   EndA1:
  printf("\n***  End of Stage 1 ***   %d record(s) selected ***\n",i);
       fflush(fout);
       close(fout) ;
       EXEC SQL CLOSE CA1;
     if (i >= 5)         chekflg() ;

/* ------------------------------------------------------------- */

  stage2:
    printf("\nStage 2\n");
    fout=fopen("lsr.dat","w");
    filhd1(cod,1);
    filhd2() ;
     strcpy(stmt2,stmt01);
     strcat(stmt2,stmt03);                strcat(stmt2,mtrl)  ;
     strcat(stmt2,stmt04) ;               strcat(stmt2,mtrl) ;
     strcat(stmt2,stmt05) ;               strcat(stmt2,cthk) ;
     strcat(stmt2," AND type = '") ;      strcat(stmt2,itype.arr) ;
     strcat(stmt2,"'") ;                  strcat(stmt2," AND cl= ") ;
     strcat(stmt2,ccl) ;
   EXEC SQL WHENEVER NOT FOUND GOTO EndA2;
   EXEC SQL PREPARE S2 FROM :stmt2 ;
   EXEC SQL DECLARE CA2 CURSOR FOR  S2 ;
   EXEC SQL OPEN CA2 ;
      for (i=0; ;i++)
          {
   EXEC SQL FETCH CA2 INTO  :thick,:power,:feed,:gasprsr,:stdof,
                            :prf,:plswdth,:plsenrgy,:kerf,
                            :haz,:taper,:rfness,:drosiz ;
     datout();
          }
   EndA2:
  printf("\n***  End of Stage 2 ***   %d record(s) selected ***\n",i);
     EXEC SQL CLOSE CA2;
     fflush(fout);
     close(fout) ;
     if (i >= 5)         chekflg() ;
     if (flg3 != 2)      goto stage4 ;

/* ------------------------------------------------------------- */

  stage3:
    printf("\nStage 3\n");
    fout=fopen("lsr.dat","w");
    filhd1(cod,1);
    filhd2() ;
     strcpy(stmt3,stmt01);
```

```
     strcat(stmt3,stmt03);        strcat(stmt3,mtrl)  ;
     strcat(stmt3,stmt04) ;       strcat(stmt3,mtrl) ;
     strcat(stmt3,stmt05) ;       strcat(stmt3,cthk) ;
     strcat(stmt3," AND type='"); strcat(stmt3,itype.arr) ;
     strcat(stmt3,"' AND cd= ") ;
     strcat(stmt3,ccd) ;          strcat(stmt3," AND plstat='");
     strcat(stmt3,iplstat.arr);   strcat(stmt3,"'");

    EXEC SQL WHENEVER NOT FOUND GOTO EndA3;
    EXEC SQL PREPARE S3 FROM :stmt3 ;
    EXEC SQL DECLARE CA3 CURSOR FOR  S3 ;
    EXEC SQL OPEN CA3 ;
       for (i=0; ;i++)
           {

    EXEC SQL FETCH CA3 INTO  :thick,:power,:feed,:gasprsr,:stdof,
                             :prf,:plswdth,:plsenrgy,:kerf,
                             :haz,:taper,:rfness,:drosiz ;
       datout();

          }
  EndA3:
  printf("\n***  End of Stage 3 ***   %d record(s) selected ***\n",i);
       EXEC SQL CLOSE CA3;
       fflush(fout);
       close(fout) ;
    if (i >= 5)         chekflg() ;

/* ----------------------------------------------------------------- */

  stage4:
    printf("\nStage 4\n");
    fout=fopen("lsr.dat","w");
    filhd1(cod,1);
    filhd2() ;
     strcpy(stmt4,stmt01);
     strcat(stmt4,stmt03);        strcat(stmt4,mtrl)  ;
     strcat(stmt4,stmt04) ;       strcat(stmt4,mtrl) ;
     strcat(stmt4,stmt05) ;       strcat(stmt4,cthk) ;
     strcat(stmt4," AND type='"); strcat(stmt4,itype.arr);
     strcat(stmt4,"' AND plstat='");
     strcat(stmt4,iplstat.arr);   strcat(stmt4,"'");

    EXEC SQL WHENEVER NOT FOUND GOTO EndA4;
    EXEC SQL PREPARE S4 FROM :stmt4 ;
    EXEC SQL DECLARE CA4 CURSOR FOR  S4 ;
    EXEC SQL OPEN CA4 ;
       for (i=0; ;i++)
           {
    EXEC SQL FETCH CA4 INTO  :thick,:power,:feed,:gasprsr,:stdof,
                             :prf,:plswdth,:plsenrgy,:kerf,
                             :haz,:taper,:rfness,:drosiz ;
       datout();
        }
  EndA4:
  printf("\n***  End of Stage 4 ***   %d record(s) selected ***\n",i);
```

```
      EXEC SQL CLOSE CA4;
      fflush(fout);
      close(fout) ;
    if (i >= 5)          chekflg() ;
    if (flg2 != 2)       goto stage6 ;

/* ------------------------------------------------------------------ */

  stage5:
    printf("\nStage 5\n");
    fout=fopen("lsr.dat","w");
    filhdl(cod,1);
    filhd2() ;
     strcpy(stmt5,stmt01);
     strcat(stmt5,stmt03);           strcat(stmt5,mtrl) ;
     strcat(stmt5,stmt04) ;          strcat(stmt5,mtrl) ;
     strcat(stmt5,stmt05) ;          strcat(stmt5,cthk) ;
     strcat(stmt5," AND cl= ") ;     strcat(stmt5,ccl)  ;

    EXEC SQL WHENEVER NOT FOUND GOTO EndA5;
    EXEC SQL PREPARE S5 FROM :stmt5 ;
    EXEC SQL DECLARE CA5 CURSOR FOR  S5 ;
    EXEC SQL OPEN CA5 ;
       for (i=0; ;i++)
           {
    EXEC SQL FETCH CA5 INTO  :thick,:power,:feed,:gasprsr,:stdof,
                             :prf,:plswdth,:plsenrgy,:kerf,
                             :haz,:taper,:rfness,:drosiz ;
        datout();
         }
  EndA5:
  printf("\n***  End of Stage 5 ***   %d record(s) selected ***\n",i);
       EXEC SQL CLOSE CA5;
       fflush(fout);
       close(fout) ;
     if (i >= 5)         chekflg() ;

/* ---------------------------------------------------------------- */

  stage6:
    printf("\nStage 6\n");
    fout=fopen("lsr.dat","w");
    filhdl(cod,1);
    filhd2() ;
     strcpy(stmt6,stmt01);
     strcat(stmt6,stmt03);           strcat(stmt6,mtrl)  ;
     strcat(stmt6,stmt04) ;          strcat(stmt6,mtrl) ;
     strcat(stmt6,stmt05) ;          strcat(stmt6,cthk) ;
     strcat(stmt6," AND plstat='");
     strcat(stmt6,iplstat.arr);      strcat(stmt6,"'");

    EXEC SQL WHENEVER NOT FOUND GOTO EndA6;
    EXEC SQL PREPARE S6 FROM :stmt6 ;
    EXEC SQL DECLARE CA6 CURSOR FOR  S6 ;
    EXEC SQL OPEN CA6 ;
       for (i=0; ;i++)
```

```
        {
    EXEC SQL FETCH CA6 INTO  :thick,:power,:feed,:gasprsr,:stdof,
                             :prf,:plswdth,:plsenrgy,:kerf,
                             :haz,:taper,:rfness,:drosiz ;
        datout();
        }

   EndA6:
   printf("\n***  End of Stage 6 ***   %d record(s) selected ***\n",i);
        EXEC SQL CLOSE CA6;
        fflush(fout);
        close(fout) ;
        if (i >= 5)         chekflg() ;
        if (flg1 != 1)      goto stage8 ;
/* -------------------------------------------------------------  */
   stage7:
     printf("\nStage 7\n");
     stage=7 ;
      strcpy(stmt7,stmt00);         strcat(stmt7,stmt01);
      strcat(stmt7,stmt03);         strcat(stmt7,mtrl)  ;
      strcat(stmt7,stmt04) ;        strcat(stmt7,mtrl) ;
      strcat(stmt7,stmt06) ;        strcat(stmt7,crngA1) ;
      strcat(stmt7," AND ") ;       strcat(stmt7,crngA2) ;
      strcat(stmt7," AND type='");  strcat(stmt7,itype.arr) ;
      strcat(stmt7,"' AND plstat='");
      strcat(stmt7,iplstat.arr);    strcat(stmt7,"' ");

     EXEC SQL EXECUTE IMMEDIATE :stmt7 ;
     prmsqy2(stage);

/* --------------------------------------------------------------- */

   stage8:
     printf("\nStage 8\n");
     stage=8 ;
      strcpy(stmt8,stmt00);         strcat(stmt8,stmt01);
      strcat(stmt8,stmt03);         strcat(stmt8,mtrl)  ;
      strcat(stmt8,stmt04) ;        strcat(stmt8,mtrl) ;
      strcat(stmt8,stmt06) ;        strcat(stmt8,crngA1) ;
      strcat(stmt8," AND ") ;       strcat(stmt8,crngA2) ;
      strcat(stmt8," AND plstat='");
      strcat(stmt8,iplstat.arr);    strcat(stmt8,"' ");

      EXEC SQL EXECUTE IMMEDIATE :stmt8 ;
      prmsqy2(stage);
/* ------------------------------------------------------------- */

   stage9:
     printf("\nStage 9\n");
     stage=9 ;
      strcpy(stmt9,stmt00) ;        strcat(stmt9,stmt01);
      strcat(stmt9,stmt03);         strcat(stmt9,mtrl)  ;
      strcat(stmt9,stmt04) ;        strcat(stmt9,mtrl) ;
      strcat(stmt9,stmt06) ;        strcat(stmt9,crngB1) ;
      strcat(stmt9," AND ") ;       strcat(stmt9,crngB2) ;
```

```
        strcat(stmt9," AND plstat='");
        strcat(stmt9,iplstat.arr);     strcat(stmt9,"' ");

        EXEC SQL EXECUTE IMMEDIATE :stmt9 ;
        prmsqy2(stage) ;

/* ------------------------------------------------------------ */

     stage10:
        printf("\nStage 10\n");
        stage=10 ;
        strcpy(stmt10,stmt00);         strcat(stmt10,stmt01);
        strcat(stmt10,stmt03);         strcat(stmt10,mtrl)  ;
        strcat(stmt10,stmt04) ;        strcat(stmt10,mtrl) ;
        strcat(stmt10,stmt06) ;        strcat(stmt10,crngC1) ;
        strcat(stmt10," AND ") ;       strcat(stmt10,crngC2) ;
        strcat(stmt10," AND plstat='");
        strcat(stmt10,iplstat.arr);     strcat(stmt10,"' ");

        EXEC SQL EXECUTE IMMEDIATE :stmt10 ;
        prmsqy2(stage);

/* ------------------------------------------------------------ */

     stage11:
        printf("\nStage 11\n");
        stage=11 ;
        strcpy(stmt11,stmt00);         strcat(stmt11,stmt01);
        strcat(stmt11,stmt03);         strcat(stmt11,mtrl)  ;
        strcat(stmt11,stmt04) ;        strcat(stmt11,mtrl) ;
        strcat(stmt11,stmt06) ;        strcat(stmt11,crngD1) ;
        strcat(stmt11," AND ") ;       strcat(stmt11,crngD2) ;
        strcat(stmt11," AND plstat='");
        strcat(stmt11,iplstat.arr);     strcat(stmt11,"' ");

        EXEC SQL EXECUTE IMMEDIATE :stmt11 ;
        prmsqy2(stage);

        printf("\n*** SORRY unable to find enough data ***\n") ;
        exit() ;

/* ------------------------------------------------------------ */

   /* ------  Procedure when error has occured ------ */

   EREXIT1:
          rprteror();
          return ;

}                /*     end of search routine     */

/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ */

/* -------- Routine to create a VIEW for data retrieval ------ */
```

```
prmsqy2(stage)
  int    stage ;
  {
  int    thk[50],pwrmn[50],fdmx[50] ,prfmn[50], plngymn[50];
  int    kthk, cod, i ;
      cod = 1 ;         /* i.e thickness range  */
      EXEC SQL WHENEVER SQLERROR GOTO EREXIT2 ;
      EXEC SQL WHENEVER NOT FOUND GOTO EndA71 ;
      EXEC SQL DECLARE CA71 CURSOR FOR
               SELECT DISTINCT thick FROM tmptbl ;
      EXEC SQL OPEN CA71 ;
      for (i=0; ;i++){
        EXEC SQL FETCH CA71 INTO :thick ;
        thk[i] = thick ;
        EXEC SQL
             SELECT MAX(feed),MIN(power),MIN(prf),MIN(plsenrgy)
                INTO :feed, :power, :prf, :plsenrgy
                FROM tmptbl
                WHERE thick = :thick ;
        pwrmn[i]=power ;          fdmx[i]=feed ;
        prfmn[i]=prf ;            plngymn[i]=plsenrgy ;
                      }
     EndA71:
        printf("\n Number of distinct thicknesses=%2d\n",i);
        EXEC SQL CLOSE CA71 ;
        if (i >= 2)          goto more ;
        printf("\n**** Less than 2 distinct values ****\n") ;
        EXEC SQL EXECUTE IMMEDIATE
                " DROP VIEW tmptbl " ;
        return;
     more:
      kthk = i ;
      fout=fopen("lsr.dat","w") ;
      filhdl(cod,kthk);
      fprintf(fout," thick  feed power   prf enrgy\n") ;
      for (i=0 ; i<kthk ; i++) {
      fprintf(fout,"%6d%6d%6d%6d%6d\n"  ,thk[i],
              fdmx[i], pwrmn[i], prfmn[i], plngymn[i]) ;
                                   }
      filhd2() ;
      EXEC SQL WHENEVER NOT FOUND GOTO EndA72;
      EXEC SQL DECLARE CA72 CURSOR FOR
           SELECT * FROM tmptbl  ;
      EXEC SQL OPEN CA72 ;
         for (i=0; ;i++)
             {
      EXEC SQL FETCH CA72 INTO  :thick,:power,:feed,:gasprsr,:stdof,
                                :prf,:plswdth,:plsenrgy,:kerf,
                                :haz,:taper,:rfness,:drosiz ;
           datout();
             }
    EndA72:
    printf("\n*** End of Stage %d *** %d record(s) selected ***\n",
         stage,i) ;
         EXEC SQL CLOSE CA72;
         fflush(fout);
```

```
        close(fout) ;
        EXEC SQL EXECUTE IMMEDIATE
                " DROP VIEW tmptbl " ;
    chekflg() ;

   EREXIT2:
        rprteror();
        return ;
  }

/* +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ */

/* ----------- routine to read numbers from the user ---------- */

int rdn(text,nmbr)
        char   text[];
        int    *nmbr;
        {
        char s[15];
        printf(text);
        if (gets(s) == (char *)NULL)
        return (EOF);
        *nmbr= atoi(s);
        return(1);
        }

/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ */

/* ------- routine to read character string from the user ------ */

int rds(text,string)
        char  text[],string[];
        {
         printf(text);
         return(gets(string) == (char *)NULL
         ? EOF : strlen(string) );
        }

/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ */

/* ------- A routine to write the numerical data to a file ---- */

datfil()
   {
     int  *out ,rngA1, rngA2, rngB1, rngB2 ,rngC1 ;
     int  rngC2, rngD1, rngD2 ;

      out=fopen("nmfil.dat","w") ;
      rngA1= 0.8 * ithick   ;
      rngA2= 1.2 * ithick   ;
      rngB1= 0.7 * ithick   ;
      rngB2= 1.3 * ithick   ;
      rngC1= 0.5 * ithick   ;
      rngC2= 1.5 * ithick   ;
      rngD1= 0.3 * ithick   ;
      rngD2= 1.7 * ithick   ;
```

```
      fprintf(out,"%7d %7d %7d", ithick, imaxpwr, ifeed) ;
      fprintf(out,"\n%7d %7d", icl, icd) ;
      fprintf(out,"\n%7d %7d %7d %7d %7d %7d %7d",
            ikerf, ihaz, irfness, idrosiz, itaper, iqltygrd, ispdgrd);
      fprintf(out,"\n%7d %7d %7d %7d %7d %7d %7d %7d" ,
              rngA1, rngA2, rngB1, rngB2 ,rngC1, rngC2, rngD1, rngD2);
      fprintf(out,"\n%s,%s\n", ilsrtyp.arr,iplstat.arr) ;
      fflush(out) ;
      close(out) ;
      return ;
   }

/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ */

/* ----------  Two Routines to write preliminary data to the
                            output file heading  --------------  */

filhd1(cod,kthk)
   int  cod ,kthk ;
   {
     if (cod == 0)    kthk=1 ;
     fprintf(fout,"%-25s\t\t\t**Class of material\n", mtrlcls) ;
     fprintf(fout,"%-25s\t\t\t**Type of material\n", itype.arr) ;
     fprintf(fout,"%6d\t\t\t\t\t\t**Thickness\n", ithick) ;
     fprintf(fout,"%6d%6d       \t\t\t\t**code of thickness \n",cod,kthk);
     fprintf(fout,"%6d%6d       \t\t\t\t**Max.power & max.feed\n",
             imaxpwr,ifeed);
     chekflg1();
   }
filhd2()
  {

/* --------------- Output to the data file -------------------- */

       fprintf(fout," thick power  feed gaspr stdof");
       fprintf(fout,"   prf plswd plsen  kerf");
       fprintf(fout,"   haz taper rfnes drsiz\n");

/* ------------------ Output to the VDU ----------------------- */

       printf(" thick power  feed gaspr stdof");
       printf("   prf plswd plsen  kerf");
       printf("   haz taper rfnes drsiz\n");
   }

/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ */

/* ------- Two routines to write evaluation code to the
                 heading of the output file ------------------ */

chekflg1()
   {
   char   pls[6] ;
      if (flg5 == 1)     strcpy(pls, "PULSED") ;
      if (flg5 == 2)     strcpy(pls, "CW") ;
```

```
/* ---------------- Write the evaluation codes -------------- */

      fprintf(fout,"%6d     \t\t\t\t\t**Feed evaluation code\n",flg4) ;
      fprintf(fout,"%6d%6d%6d%6d%6d%6d   \t\t**Preference weights\n" ,
              fdwt,krfwt,rfwt,drswt,tprwt,hazwt );
      fprintf(fout,"%-6s  \t\t\t\t\t**pulse status\n", pls) ;
      return;
   }

/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ */

chekflg()
          {   exit() ;     }

/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ */

/* ---------------- SQL error reporting routine --------------- */

rprteror()
   {
    EXEC SQL WHENEVER SQLERROR STOP;
     printf("\n***SORRY unable to continue .");
     printf(" Check error message below ***\n");
     printf ("\n %.70s (%d)\n", sqlca.sqlerrm.sqlerrmc, -sqlca.sqlcode);
   }

/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ */

/* ------------------- Routine for data output ----------------- */

datout()
   {
     fprintf(fout,"%6d%6d%6d%6d%6d%6d%6d%6d%6d%6d%s%6d%6d\n",
              thick,power,feed,gasprsr,stdof,prf,plswdth,
              plsenrgy,kerf,haz,taper,rfness,drosiz);

     printf("%6d%6d%6d%6d%6d%6d%6d%6d%6d%6d%s%6d%6d\n",
              thick,power,feed,gasprsr,stdof,prf,plswdth,
              plsenrgy,kerf,haz,taper,rfness,drosiz);
   }

/* --------------- End of the program ---------------------- */
```

APPENDIX 5B

PRECOMPILER AND SQL STATEMENTS

SOURCE : SQL PROGRAMMER'S MANUAL, IBM UK LTD.

SQL/RT Data Base Precompiler Statements

The following is a list of statements used by the SQL/RT Data Base precompiler. Each of these statements must start with EXEC SQL and end with a semicolon (;) in a C precompiler program.

Statement	Description
BEGIN DECLARE SECTION	Delimits the start of the description of host variables.
CLOSE	Ends the use of a cursor.
CONNECT	Establishes a connection to the SQL/RT Data Base system.
DECLARE CURSOR	Establishes a cursor for SQL statements.
DESCRIBE BIND VARIABLES	Places information about input host variables into an SQL descriptor area.
DESCRIBE SELECT LIST	Places information about output host variables into an SQL descriptor area.
END DECLARE SECTION	Delimits the end of the description of host variables.
EXEC SQL	Delimits the start of an SQL/RT Data Base statement embedded in a C program.
EXECUTE	Executes a parameterized precompiler statement.
EXECUTE IMMEDIATE	Executes a prepared SQL statement dynamically.
FETCH	Returns the values of a multi-row SQL query.
INCLUDE	Embeds predefined C structures into an SQL/RT Data Base C program.
INTO	Establishes the names of output host variables.
OPEN	Establishes a cursor.
PREPARE	Compiles a dynamic SQL statement.
WHENEVER	Sets conditions and branching for SQL errors and warnings.

SQL Statements

The following SQL statements do not appear in this appendix surrounded by the SQL/RT Data Base precompiler delimiters EXEC SQL and semicolon (;). Some of them may be delimited this way in a precompiler program and some of them may be used in conjunction with a declared cursor. The following is a list of SQL Statements supported by the SQL/RT Data Base system.

Statement	Description
ALTER TABLE	Increases the size or number of columns in a table.
COMMIT WORK	Causes changes in the data base to take effect.
CREATE INDEX	Builds and maintains an index which helps the SQL/RT Data Base system locate records more rapidly.
CREATE SYNONYM	Defines a convenient name to refer to a table or view.
CREATE TABLE	Defines a table for storing information.
CREATE VIEW	Defines an alternative perspective on a table or tables.
DELETE	Removes rows from a table.
DROP INDEX	Removes a previously created index on a table.
DROP SYNONYM	Drops a previously defined synonym.
DROP TABLE	Removes a table from the data base and deletes all its rows.
DROP VIEW	Removes a view from the data base.
FROM	Identifies the table(s) to work with.
GRANT	Allows other users access to your tables and views.
GROUP BY	Divides a table into groups of rows, based on the values in a column.
HAVING	Selects which groups to include in the results of a GROUP BY query.
INSERT	Adds new rows to an existing table.
LOCK TABLE	Temporarily prevents other users from modifying a table.

ORDER BY	Determines the order in which selected rows are presented.
REVOKE	Eliminates or reduces the access on tables or views previously granted to a user.
ROLLBACK WORK	Cancels changes you have made to the data base since the last COMMIT WORK statement.
SELECT	Retrieves information from a table.
SELECT FOR UPDATE	Temporarily prevents users from modifying rows in a table.
UPDATE	Changes values in a table.
VALIDATE INDEX	Checks the integrity of an index.
WHERE	Specifies search conditions to select rows in a query.

APPENDIX 5C

IMPLICATIONS AND LIMITATIONS OF THE DATA COLLECTION PROGRAM

The data collection program was limited by the following:

1) Not all SQL facilities are implemented for this type of interface. Consequently, the program had to be designed according to the facilities available to the interface.

2) Numerical values, which are input by conventional READ, cannot be used for database operations, e.g comparison with a database value or insertion into a database table. The reason is unkown. However, a technique was given in one of the examples in the SQL/RT programmer´s manual, which reads a number as a series of individual characters, and uses a "C" function called "atoi" (alphabetical to integer) to return the integer equivalent of each character. See subroutine "int rdn ()" in the program text. That allowed the tranfer of integers, but would not handle decimal numbers. Accordingly, all the numerical columns of the database module (except ID) were redefined as integers, to maintain program flexibility and data independency.

3) The unique identification code (ID) code had to be kept as decimal number, as the two decimal digits of this code are used as material class identifiers. A distinct pair of decimal digits were assigned to ID codes of each material table in the database module. This was done to provide a consistent method of data identification among the different tables of the database, and to free the ID code from serial number restriction, which would be implied if an integer format was used (i.e definite series of numbers for each material table). This is helpful especially in the parameters tables where data related to different materials are gathered in one table, e.g process parameters table. As the ID code is handled internally only, i.e no input or output is involved, the problem of exporting integers does not arise.

4) Data which is defined as characters in the database, has to be declared in the program as VARCHAR data type (variable lengh character string). Such data has to be handled by reading the string characters one by one, until its end (EOF) is reached (see subroutine "int rds ()"). This method works properly when reading any text form the terminal, but won´t work for multi-word string from a datafile. It stops reading at the first space in that string and considers that the job has finished. The reason could be that, the "RETURN" character which is entered at the end of typing a string from a keyboard, is considered as the EOF signal. While when reading from a data file,

any space is considered as EOF. Therefore, all the character data type in the database tables were made continuous by inserting a minus sign or an underscore, e.g MILD-STEEL.

5) For unknown reason, negative numbers could not be retrieved from the database. This problem was tackled by declaring the corresponding variable as "char" data type (character), which SQL/RT consideres as raw data (ref 5.3). This procedure was necessary for taper data, because it may be negative (chapter 4).

6) Any variable from the host language that is used within SQL statements, must be declared as a "Host Variable". It must be explicitly declared within a special declaration section within the program code. That section is specified by BEGIN DECLARE SECTION and END DECLARE SECTION. When using a host varible within an SQL statement, it must be preceeded by a colon (:). A declared host variable becomes common to the main routine and all the subroutines called during the run.

7) Names of database tables and columns must be explicitly stated in SQL statements, i.e cannot be submitted as character constants. This restriction did not affect the dealing with columns because they were all predetermined and specified at the program writing stage. Dealing with tables was affected, specifically material tables. A

material table which is required to be searched, cannot be anticipated before the run time. Tha is, specifically after the user specifies the target class of material. That restriction was tackled by using a technique called "Dynamically Defined SQL Statements". In this method of programming, the SQL statement (and hence the table name) is not known to the SQL/RT before the run time.

APPENDIX 5D

DYNAMICALLY DEFINED SQL STATEMENTS

Dynamically Defined SQL statements are the heart of the database search processes used by the program. "Stage1:" of the program code is a representative sample of these statements. The outline of this method is:

1) The SQL statement is prepared from a string that comprises the required statement. That string must have already been declared as a host variable. It is prepared as follows:

```
EXEC SQL PREPARE S1 FROM   :stmt1 ;
```

2) Declaration of a CURSOR that is associated with that statement, as in:

```
EXEC SQL DECLARE CA1 CURSOR FOR S1 ;
```

3) Opening that CURSOR for execution of the statement:

```
EXEC SQL OPEN CA1 ;
```

4) FETCHing the required data from the database tables one row at a time, using host variables as the transfer media of the data:

```
EXEC SQL FETCH CA1 INTO  :hv1, :hv2, .... ;
```

5) Output the data to the specified devices, e.g to the display or a data file.

This method was used to solve the problem of dealing with table names at run time, by using a technique which generates the string (stmt1) in step 1 above. This technique assembled the required string by concatenation of several text components (e.g stmt01, stmt02 and stmt03 in Appendix 5A) at run time, the material table name was passed as one of these components. The SQL statement of "stage1:", for example, would read as in fig 1, when assembled at the program run time.

It should be noted that all of the string components should be declared as character data type, but not necessarily as host variables. Numerical variables like "thick" and "c1" are not accepted. To get around this, all the numerical values that are required in such positions, are written to a data file (see "datfil()" subroutine) at the end of the interrogation routine . These data are read at the begining of the search routine, as character data type.

In stage 7 and the subsequent stages a different search method, from that of the preceeding stages, is used. The prepared statement is used to create a VIEW containing the collected data. This view is then inspected for the number of distinct thickness values (subroutine "prmsqy2()"). If this number is more than "one", the data of the whole view is written to the output data file. Otherwise, a new search process is commenced.

The major limitation of the Dynamically Defined SQL Statement method is that, a declared CURSOR is associated with only one specific SQL statement. So that once a particular statement is joined with a CURSOR, no other statement can replace it for the entire run of the program. This phenomena implicated a long program code, since each search stage requires a different SQL statement. Therefore, at each stage it is required to: Explicitly define a statement, explicitly declare a CURSOR for that statement and explicitly execute the prepared statement. Consequently, it is impossible to implicitly or conditionally define an SQL statement in compliance with the implied set of constraints. In other words it is impossible to use one program code for database search, that would simulate all the search stages, in an iterative mode. While each stage must be equiped with an individual program code.

The explanation of the above limitation was the conclusion of a series of extensive trials to overcome that limitation. The main goals of these trials were: Program compactness and unlimited number of conceptual search stages in one physical block of program code.

Assuming that the user specified: 1, SS41, 2600, 2 and 2, for: material class, material type, thickness, CL code and CD code, respectively. The SQL statement will read:

```
SELECT   thick, power, feed, gasprsr, stdof,
         prf, plswdth, plsenrgy, kerf, haz,
         taper, rfness, drosiz
FROM     mldst01, prcprms, cutqlty
WHERE    prcprms.id = mldst01.id
 AND  prcprms.id = cutqlty.id
AND      thick = 2600 AND type = ´SS41´
AND      cl= 2  AND   cd = 2
```

N.B:

"mldst01" Is the material table of MILD STEEL material class.

"prcprms" and "cutqlty" are parameters tables as per APPENDIX 3A.

FIG 1 : EXAMPLE OF SQL STATEMENT ASSEMBLY AT RUN TIME

APPENDIX 6A

THE SELECTION OF POLYNOMIAL DEGREE

The effects of polynomial degree and data distribution on polynomial curve fitting, are discussed in this appendix, as follows:

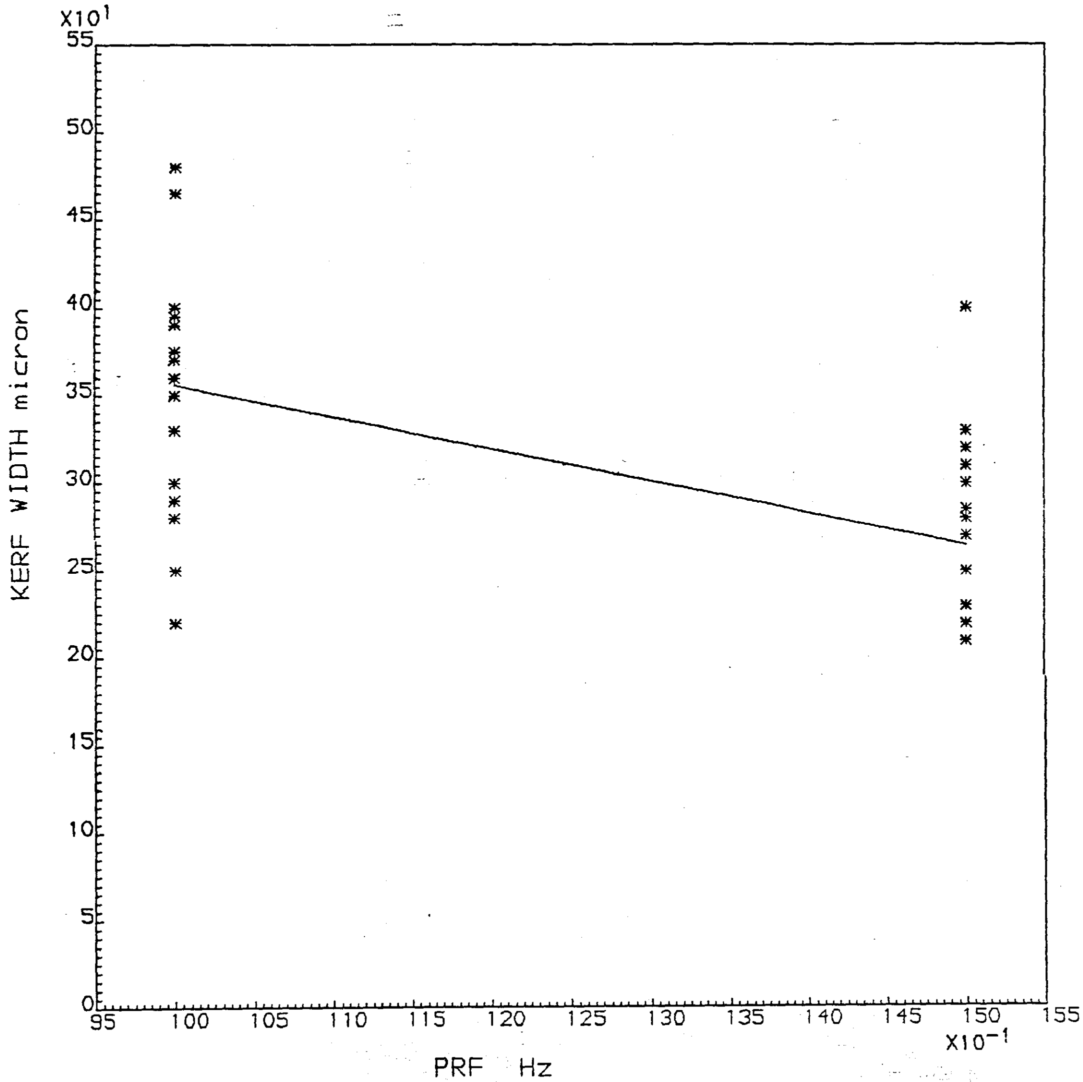
1) The limitation implied by the data form, on the maximum polynomial degree is shown in fig´s 6A1 and 6A2. In the first figure, only two distinct values occur on the abscissa, consequently, a straight line fit (degree 1) is the only choice. While in fig 6A2, a curve of degree 2 can be fitted, since there are three distinct values on the abscissa.

2) Fig 6A3 extends this situation, where the maximum degree that can applied is 4. A curve of this degree (C1) is rather unstable and gives misleading results. A curve of degree 3 (C2) is reasonably stable and accurate. The fluctuation of C1 is caused mainly by the relatively wide gaps between the groups of data.

3) Fig 6A4 demonstrates the opposite extreme to the situations of (1) and (2). Where a large number of data points, that are reasonably evenly scattered, are

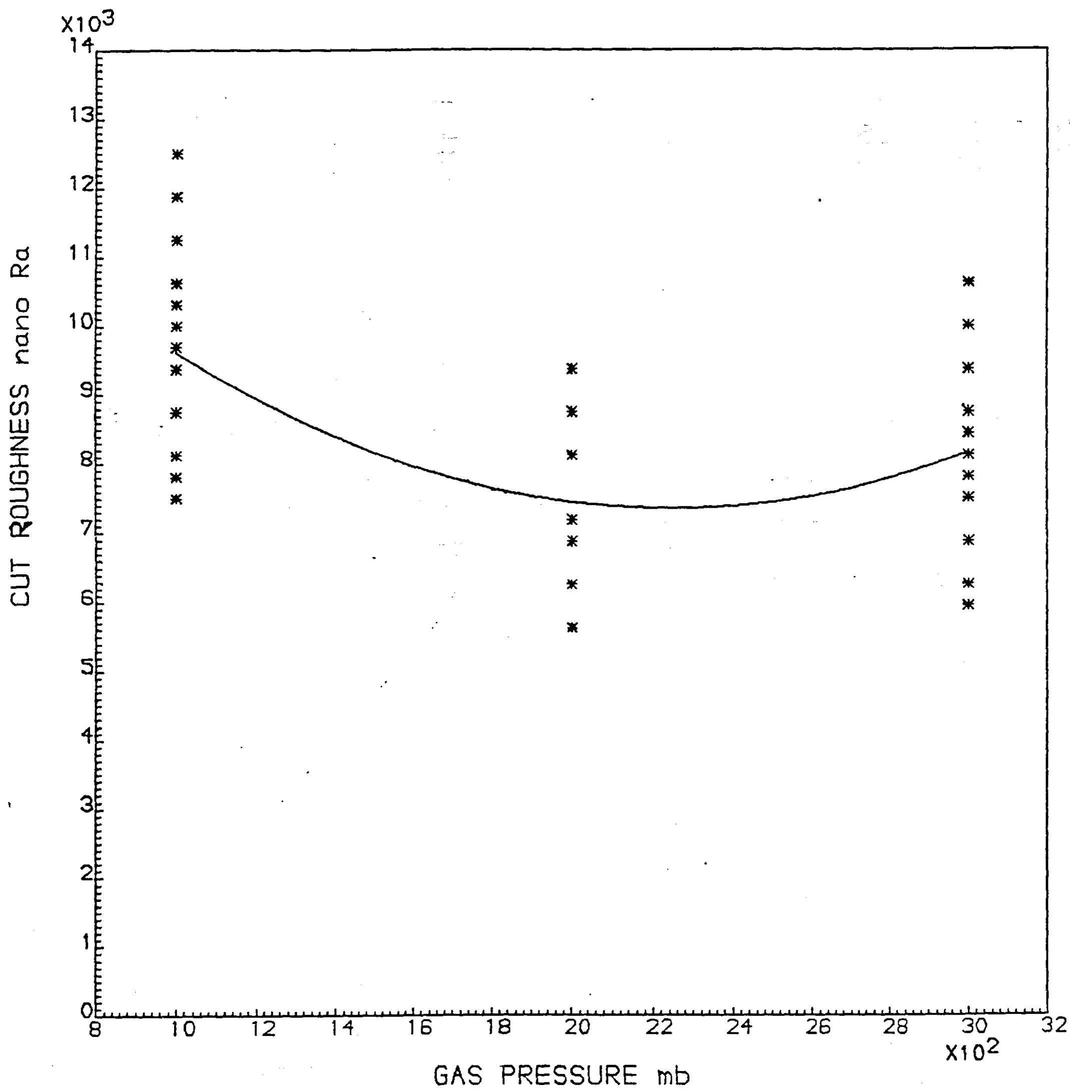
encountered. A high degree polynomial can be fitted to this data, due to the high number of distinct data on the abscissa. Though, no significant difference can be observed between a degree 8 (C1) and degree 2 (C2) curves.

It can, therefore, be concluded that low degree fitting is best suited the current application. Where the fit must be stable and reasonably accurate. An additional advantage gained is a reduction in computer processing and user waiting time, due to the fewer terms in the fitting equations.



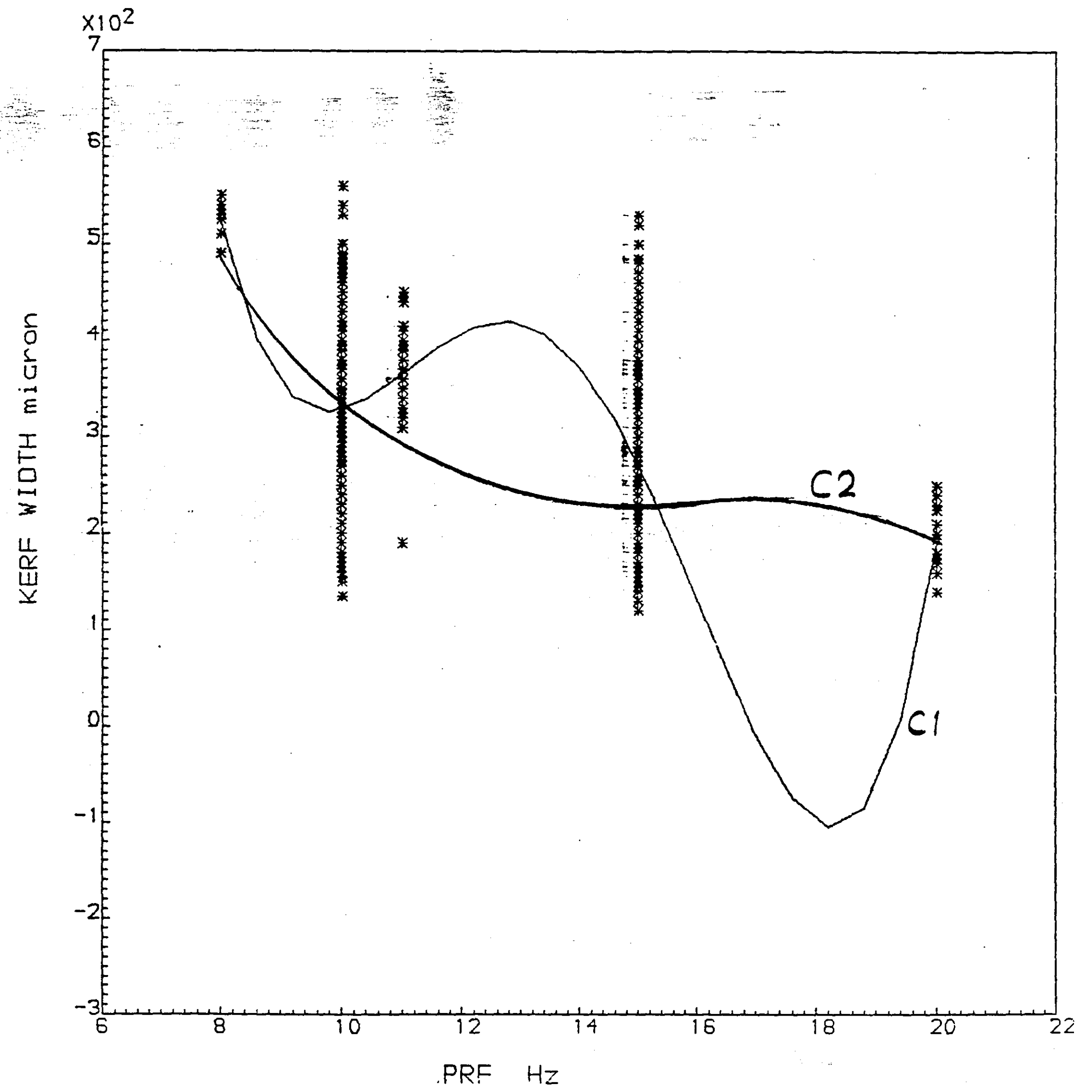
KERF WIDTH Vs PRF FOR TITANIUM 1.6mm

FIG 6A1



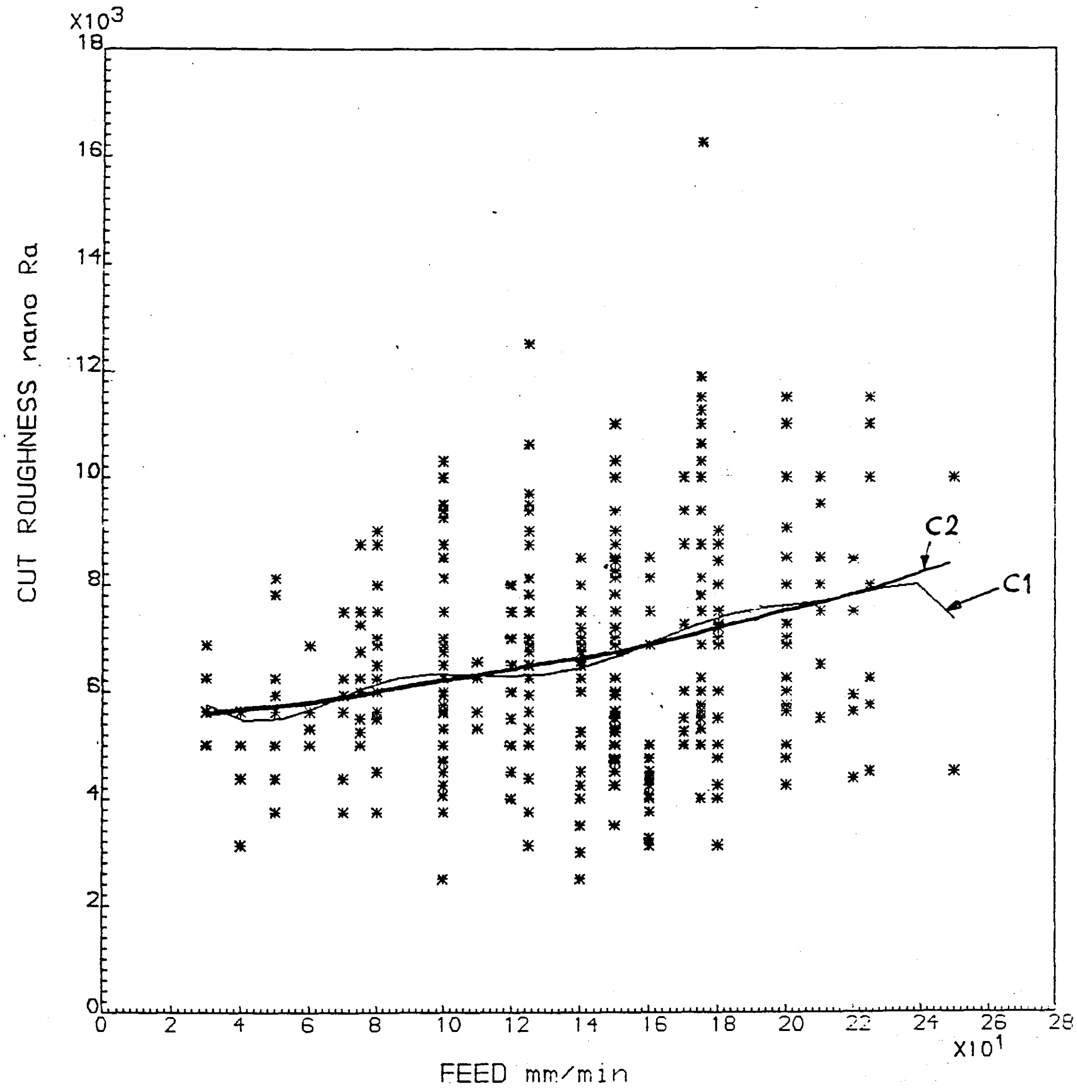
CUT ROUGHNESS Vs GAS PRESSURE FOR TITANIUM 1.6mm

FIG 6A2



KERF WIDTH Vs PRF FOR ALL MATERIALS

FIG 6A3



CUT ROUGHNESS Vs FEED FOR ALL MATERIALS

FIG 6A4

APPENDIX 6B

LISTING OF THE PREDICTION PROGRAM CODE

```
      PROGRAM  PREDICT

C ... WRITTEN BY MOHAMMED A. HUSSAIN, during 1988.
C ... For prediction of laser cutting parameters
C ... using different routines from NAG-library.
C ... It evaluates in terms of the type of laser.
C ***********************************************

      IMPLICIT DOUBLE PRECISION (A-H),(K-Z)
C
      INTEGER    IMXFEED,IMXPWR,ITHKNS,ITHICK(900),
     1            IFEED(900),IPWR(900),IGASP(900),
     2            ISTDOF(900),IPRF(900),IPLSWD(900),
     3            IPLSEN(900),IKERF(900),IHAZ(900),
     4            ITPR(900),IRFNS(900),IDRSIZ(900),
     5            IDSTHK(30),IDSTFD(30),IDSTPWR(30),
     6            IDSTPRF(30),IDSTENGY(30),IFDWT,
     7            IKRFWT,IRFWT,IDRSWT,ITPRWT,IHAZWT,
     8            IFAIL,PRCSCOD,EVALCOD,KDST
C
C ... Reals
      DIMENSION  THICK(900),FEED(900),PWR(900),
     2            STDOF(900),PRF(900),PLSWD(900),
     3            PLSEN(900),KERF(900),HAZ(900),
     4            TPR(900),RFNS(900),DRSIZ(900),
     5            DSTHK(30),DSTFD(30),DSTPWR(30),
     6            DSTPRF(30),DSTENGY(30),GASP(900)
C
      CHARACTER  MTRLCLS*25,MTRLTYP*25,PLSTAT*6
C
C .... Local parameters ...
C
      INTEGER    NCASE,IN,OUT,VIN,VOUT,FLG1,FLG2,FLG3,
     1            FLG4,FLG5,FLG6,FGL7,FLG8,FLG9,FLG10,FLG11,
     2            FLG12,FLG13,LOGF,IDEG,KTER,IRANK(150)
      DIMENSION  THKF(101),FEEDF(101),PWRF(101),
     *            PRFF(101),PLSEF(101),TMP(101,2)
C
C ... These are dummy arguments for calling subroutines.
C
      DIMENSION   D1(101),D2(101),D3(50),D4(50)
C
      CHARACTER  INDAT*35,OUTDAT*35,CTHIK*9,CFEED*4,
     1            CPWR*5,CKERF*10,CHAZ*8,CRFNS*13,BEL,
     2            CTPR*5,CDROS*10,CPRF*4,CPLSWD*11,
     3            CGASP*12,CPLSEN*12,CSTDF*9,LOGFL*40,
     4            STMT*70,DAT*9,TIM*8
C ...............................................
C ... External modules...
C ... PROCS1,PROCS2,GRFSTT,GRAF1,GRAF2,GRAF3
C ...............................................
      DATA   VIN,VOUT,IN,LOGF,OUT,BEL /5,6,10,15,20,7/
      DATA   CTHIK,CFEED,CPWR,CKERF,CHAZ,CRFNS,CPRF,
     1       CPLSEN,CPLSWD,CDROS,CGASP,CTPR,CSTDF
     2       /'THICKNESS','FEED','POWER','KERF WIDTH',
     3          'H.A.ZONE','CUT ROUGHNESS','PRF',
```

```
     4           'PULSE ENERGY','PULSE WIDTH','DROSS SIZE'
     4           ,'GAS PRESSURE','TAPER','STAND OFF'/
C ..................................................
        D5 = 0.0000                !Dummy
        D6 = 0.0000                !arguments
C ... Display introductory message
        WRITE (VOUT,00001)
C ... Ask for input & output files.
20      WRITE (VOUT,99999)  BEL
        READ (VIN,99998)  INDAT
        IF (INDAT.EQ. ' ')    GOTO 20
        OPEN (UNIT=10,FILE=INDAT,STATUS='OLD',ERR=25)
        GOTO 30
25      WRITE (VOUT,99997)   BEL
        GOTO 20
C ...
30      WRITE (VOUT,99996)  BEL
        READ (VIN,99998)  OUTDAT
        IF (OUTDAT.EQ.' ')    GOTO 30
C ...
        LOGFL = OUTDAT // '.LOG'
        OPEN (UNIT=15,FILE=LOGFL,STATUS='NEW')
        OPEN (UNIT=20,FILE=OUTDAT,STATUS='NEW')
C ...
        WRITE (OUT,99993) !File heading
C ... Ask for the desired degree of the polynomial.
        WRITE (VOUT,99995) BEL
        READ (VIN,*) IDEG
C ... Ask for graphical OUTPUT device
        WRITE (VOUT,99994)  BEL
        READ (VIN,*) KTER
C       ***************************************
C       *        ACQUIRE EXTERNAL DATA        *
C       This is done by reading the data files
C       which are output from the database .
C       ***************************************
        READ (IN,99998) MTRLCLS
        READ (IN,99998) MTRLTYP
        READ (IN,*) ITHKNS
        READ (IN,111) PRCSCOD,KDST
        READ (IN,*) IMXPWR,IMXFEED
        READ (IN,*) EVALCOD
        READ (IN,113)IFDWT,IKRFWT,IRFWT,IDRSWT,
     *              ITPRWT,IHAZWT
        READ (IN,99998) PLSTAT
C ... Check for thickness range existance....
        IF (PRCSCOD.NE.1)     GOTO 40
        READ (IN,*)           !Dummy line
        DO 40 I=1,KDST
          READ (IN,115) IDSTHK(I),IDSTFD(I),
     *        IDSTPWR(I),IDSTPRF(I),IDSTENGY(I)
          DSTHK(I) = FLOAT(IDSTHK(I))
          DSTFD(I) = FLOAT(IDSTFD(I))
          DSTPWR(I) = FLOAT(IDSTPWR(I))
          DSTPRF(I) = FLOAT(IDSTPRF(I))
          DSTENGY(I) = FLOAT(IDSTENGY(I))
```

```
C ... Continue reading the data file ...
40      CONTINUE
        READ (IN,*)          ! Dummy line
        READ (IN,117,END=50) (ITHICK(I),IPWR(I),
     1        IFEED(I),IGASP(I),ISTDOF(I),IPRF(I),
     2        IPLSWD(I),IPLSEN(I),IKERF(I),
     3        IHAZ(I),ITPR(I),IRFNS(I),IDRSIZ(I),
     4        I=1,900)
111     FORMAT (2I6)
113     FORMAT (6I6)
115     FORMAT (5I6)
117     FORMAT (13I6)
C ..........................................
50      NCASE=I-1           ! NO. OF CASES
C ...
C ... Transfer the input parameters from integers
C ... into reals,and get the max. &min. of them.
C ...
        THKNS = FLOAT (ITHKNS)
        MXPWR = FLOAT (IMXPWR)
        MXFEED = FLOAT (IMXFEED)
        FDWT = (FLOAT (IFDWT))/100.
        KRFWT = (FLOAT (IKRFWT))/100.
        RFWT = (FLOAT (IRFWT))/100.
        DRSWT = (FLOAT (IDRSWT))/100.
        TPRWT = (FLOAT (ITPRWT))/100.
        HAZWT = (FLOAT (IHAZWT))/100.
C ...
        CALL MINMX(NCASE,ITHICK,THICK,THKMAX,THKMIN,FLG1)
        CALL MINMX(NCASE,IPWR,PWR,PWRMAX,PWRMIN,FLG2)
        CALL MINMX(NCASE,IFEED,FEED,FDMAX,FDMIN,FLG3)
        CALL MINMX(NCASE,IGASP,GASP,GSPMAX,GSPMIN,FLG4)
        CALL MINMX(NCASE,ISTDOF,STDOF,STDFMAX,STDFMIN,FLG5)
        CALL MINMX(NCASE,IPRF,PRF,PRFMAX,PRFMIN,FLG6)
        CALL MINMX(NCASE,IPLSWD,PLSWD,PLSWMX,PLSWMN,FLG7)
        CALL MINMX(NCASE,IPLSEN,PLSEN,PLSEMX,PLSEMN,FLG8)
        CALL MINMX(NCASE,IKERF,KERF,KRFMAX,KRFMIN,FLG9)
        CALL MINMX(NCASE,IHAZ,HAZ,HAZMAX,HAZMIN,FLG10)
        CALL MINMX(NCASE,ITPR,TPR,TPRMAX,TPRMIN,FLG11)
        CALL MINMX(NCASE,IRFNS,RFNS,RFNSMAX,RFNSMIN,FLG12)
        CALL MINMX(NCASE,IDRSIZ,DRSIZ,DRSMAX,DRSMIN,FLG13)
C       ********************************************
C ... Write material class & type.
        WRITE (OUT,99992) MTRLCLS,MTRLTYP,ITHKNS
C       ........................................
C       *   SELECT GRAPHICAL OUTPUT DEVICE - *
C       ........................................
       IF (KTER .EQ. 1)THEN
            CALL T4010
            CALL SCALE(0.45)
        ELSE IF (KTER .EQ. 2)THEN
            CALL B1302
        ELSE IF (KTER .EQ. 3) THEN
            CALL V1200
            CALL SCALE(0.9)
        ELSE IF (KTER .EQ. 4) THEN
```

```
            CALL LA100
        ENDIF
C ...............................................
C ... Check for pulse status .......
        IF (PLSTAT.EQ.'PULSED')  GOTO  707
C ***********************************************
C ... Start evaluation for CW lasers...
C ... Evaluate for thickness ...
        IF (PRCSCOD.NE.1)   THEN
          WRITE (OUT,99990) THKMAX
          GOTO 60
        ELSEIF (FLG1.EQ.1)   THEN
          WRITE (OUT,99985) THKMAX
          GOTO 60
        ELSE
          WRITE(OUT,99988)
        END IF
C ...............................................
C ... Check for existence of FEED & POWER ranges ...
        IF ((FLG2.EQ.1).AND.(FLG3.EQ.1)) THEN
          WRITE (OUT,99975) FDMAX,PWRMAX
          GOTO 90
        ELSEIF (FLG3.EQ.1)  THEN
          WRITE (OUT,99980) FDMAX
          GOTO 55
        ENDIF
        CALL PROCS1(THICK,THKMAX,THKMIN,FEED,FDMAX,
     1                 FDMIN,CTHIK,CFEED,MTRLTYP,
     2                 THKF,FEEDF,DSTHK,DSTFD,
     3                 THKNS,EXFD,NCASE,IDEG,KDST,0)
C ... Check if no power range ...
55      IF (FLG2.EQ.1)    THEN
          WRITE (OUT,99965) PWRMAX
          GOTO 67
        ENDIF
        CALL PROCS1(THICK,THKMAX,THKMIN,PWR,PWRMAX,
     1              PWRMIN,CTHIK,CPWR,MTRLTYP,
     2              THKF,PWRF,D3,D4,THKNS,
     3              EXPWR,NCASE,IDEG,1,0)
C ... Call for the surface fitting routine...
        CALL PROCS2(THKF,PWRF,FEEDF,THKNS,CTHIK,CPWR,
     1              CFEED,MTRLCLS,MTRLTYP,101,KTER)
        GOTO 67
C ... End of thickness range operations ...
C *****************************************************
C ... Evaluations in terms of FEED for process parms.
60      IF ((FLG2.EQ.1).AND.(FLG3.EQ.1)) THEN
          WRITE (OUT,99975) FDMAX,PWRMAX
          GOTO 90
        ELSEIF (FLG3.EQ.1)    THEN
          WRITE(OUT,99980)  FDMAX
          GOTO  70
        ELSE IF (FLG2.EQ.1)  THEN
          WRITE (OUT,99965) PWRMAX
          GOTO 67
        ENDIF
```

```
          CALL PROCS1(FEED,FDMAX,FDMIN,PWR,PWRMAX,
     1                PWRMIN,CFEED,CPWR,MTRLTYP,
     2                FEEDF,PWRF,D3,D4,D5,D6,
     3                NCASE,IDEG,1,0)
          GOTO 67
C ************************************************
C ... Start evaluations for PULSED lasers ...
707       IF (PRCSCOD.NE.1)   THEN
            WRITE (OUT,99990) THKMAX
            GOTO 64
          ELSEIF (FLG1.EQ.1)   THEN
            WRITE (OUT,99985) THKMAX
            GOTO 64
          ENDIF
          WRITE (OUT,99988)
          IF ((FLG8.EQ.1).AND.(FLG6.EQ.1)) THEN
            WRITE (OUT,99955) PRFMX,PLSEMX
            IF (FLG3.NE.1)   THEN
            CALL PROCS1(THICK,THKMAX,THKMIN,FEED,FDMAX,
     1                  FDMIN,CTHIK,CFEED,MTRLTYP,
     2                  THKF,FEEDF,DSTHK,DSTFD,THKNS,
     3                  EXFD,NCASE,IDEG,KDST,0)
            ENDIF
            GOTO 67
          ELSEIF (FLG8.EQ.1)   THEN
            WRITE (OUT,99945) PLSEMX
            GOTO  66
          ENDIF
          CALL PROCS1(THICK,THKMAX,THKMIN,PLSEN,
     1                PLSEMX,PLSEMN,CTHIK,CPLSEN,
     2                MTRLTYP,THKF,PLSEF,D3,D4,
     3                THKNS,EXENGY,NCASE,IDEG,1,0)
C ... Find a suitable pulse energy ...
          DO 62 I=1,101
            IF (THKF(I).GE.THKNS)     THEN
              WRITE (OUT,99943)  PLSEF(I)
              GOTO 66
            ENDIF
62        CONTINUE
64        IF (PRCSCOD.NE.1)  THEN
C ... Quote mean & min.pulse energy  ....
            AVENGY = 0.0
            DO 65 I=1,NCASE
               AVENGY = AVENGY+PLSEN(I)
65          CONTINUE
            AVENGY = AVENGY / NCASE
            WRITE (OUT,99935) AVENGY,PLSEMN
          ENDIF
66        IF (FLG6.EQ.1)  THEN
            WRITE (OUT,99925)  PRFMAX
            GOTO  67
          ELSEIF (FLG3.EQ.1)   THEN
            WRITE(OUT,99980) FDMAX
            GOTO  70
          ENDIF
C
```

```
      CALL PROCS1(FEED,FDMAX,FDMIN,PRF,PRFMAX,
     1             PRFMIN,CFEED,CPRF,MTRLTYP,FEEDF,
     2        PRFF,D3,D4,EXFD,EXPRF,NCASE,IDEG,1,0)
C ..............................................
C ... Evaluate in terms of FEED for quality features.
67    IF (FLG3.EQ.1)  GOTO 70
      WRITE (LOGF,99915)
      CALL GRFSTT(CFEED)
      CALL PROCS1(FEED,FDMAX,FDMIN,KERF,KRFMAX,
     1             KRFMIN,CFEED,CKERF,MTRLTYP,D1,D2,
     2             D3,D4,EXFD,FDKF,NCASE,IDEG,1,1)
      CALL PROCS1(FEED,FDMAX,FDMIN,RFNS,RFNSMAX,
     1             RFNSMIN,CFEED,CRFNS,MTRLTYP,D1,D2,
     2             D3,D4,EXFD,FDRF,NCASE,IDEG,1,2)
      CALL PROCS1(FEED,FDMAX,FDMIN,DRSIZ,DRSMAX,
     1             DRSMIN,CFEED,CDROS,MTRLTYP,D1,D2,
     2             D3,D4,EXFD,FDRS,NCASE,IDEG,1,3)
      CALL PROCS1(FEED,FDMAX,FDMIN,TPR,TPRMAX,
     1             TPRMIN,CFEED,CTPR,MTRLTYP,D1,D2,
     2             D3,D4,EXFD,FDTR,NCASE,IDEG,1,4)
      CALL PROCS1(FEED,FDMAX,FDMIN,HAZ,HAZMAX,
     1             HAZMIN,CFEED,CHAZ,MTRLTYP,D1,D2,
     2             D3,D4,EXFD,FDHZ,NCASE,IDEG,1,5)
C ... Compute the final answers ....
      IF ((PRCSCOD.EQ.1).AND.(PLSTAT.EQ.'CW')) THEN
C ... When thickness range & CW laser exist ...
        IK=0

        DO  I=1,101
          IF (FEEDF(I).GT.10.)   THEN
            IK=IK+1
            TMP(IK,1)=FEEDF(I)
            TMP(IK,2)=PWRF(I)
          ENDIF
        END DO
C ... Test for positive FEED ....
      IF (IK.LT.1)    THEN
        WRITE (OUT,99914)
        GOTO 70
      ENDIF
C ... Call the sorting routines ...
C ... Rank the data into ascending order ...
        IFAIL=1
        CALL M01DAF(TMP(1,1),1,IK,'A',IRANK,IFAIL)
C ... Sort the ranked data ...
        IFAIL=1
        DO  I=1,2
           CALL M01EAF(TMP(1,I),1,IK,IRANK,IFAIL)
        END DO
C ... Reset FEED & POWER arrays ....
        DO  I=1,IK
          FEEDF(I)=TMP(I,1)
          PWRF(I) =TMP(I,2)
        END DO
```

```
C ... Adjust max. & min limits of FEED ...
          IF (FDKF.LT.FEEDF(1))    FDKF=FEEDF(1)
          IF (FDKF.GT.FEEDF(IK))   FDKF=FEEDF(IK)
          IF (FDRF.LT.FEEDF(1))    FDRF=FEEDF(1)
          IF (FDRF.GT.FEEDF(IK))   FDRF=FEEDF(IK)
          IF (FDRS.LT.FEEDF(1))    FDRS=FEEDF(1)
          IF (FDRS.GT.FEEDF(IK))   FDRS=FEEDF(IK)
          IF (FDTR.LT.FEEDF(1))    FDTR=FEEDF(1)
          IF (FDTR.GT.FEEDF(IK))   FDTR=FEEDF(IK)
          IF (FDHZ.LT.FEEDF(1))    FDHZ=FEEDF(1)
          IF (FDHZ.GT.FEEDF(IK))   FDHZ=FEEDF(IK)
          FDMAX = FEEDF(IK)
          FDMIN = FEEDF(1)
C ... End of thick.range evaluation.
        ELSE
            IK=1 !When no thickness range
        ENDIF
        WRITE (OUT,99913)FDKF,FDRF,FDRS,FDTR,FDHZ
        IF (EVALCOD.LT.0)  THEN      !No quality optoins
          WRITE (OUT,99912)
          FNLFD = FDMAX             !Resultant Feed
          GOTO 69
        ENDIF
        WRITE (OUT,99911)KRFWT,RFWT,DRSWT,TPRWT,HAZWT,FDWT
C ... When preferences on quality items exists...
C ... Use D3 array to hold the sums of res.squares.
        D3(1) = 9.E20     !Initial value
        DO 68 I=2,50
          II=I-1
          FNLFD = FDMAX-(II*.02*(FDMAX-FDMIN))       !Initially
          D3(I)=((FDWT*ABS(FDMAX-FNLFD))**2  +
     1            (KRFWT*ABS(FDKF-FNLFD))**2  +
     2            (RFWT*ABS(FDRF-FNLFD))**2   +
     3            (DRSWT*ABS(FDRS-FNLFD))**2  +
     4            (TPRWT*ABS(FDTR-FNLFD))**2  +
     5            (HAZWT*ABS(FDHZ-FNLFD))**2  )
          IF (D3(I).GT.D3(II))       THEN
C ... Restore the previous value of resultant feed...
             FNLFD = FNLFD+(.02*(FDMAX-FDMIN))
             GOTO 69
          ENDIF
68      CONTINUE
69      WRITE (OUT,99910)FNLFD
C .............................................
C ... Evaluate in terms of laser POWER ...
70      IF (PLSTAT.EQ.'PULSED')  GOTO 80
        IF (FLG2.EQ.1)    GOTO 90
        WRITE (LOGF,99905)
        CALL GRFSTT (CPWR)
        CALL PROCS1(PWR,PWRMAX,PWRMIN,KERF,KRFMAX,
     1            KRFMIN,CPWR,CKERF,MTRLTYP,D1,D2,
     2            D3,D4,EXPWR,D5,NCASE,IDEG,1,1)
        CALL PROCS1(PWR,PWRMAX,PWRMIN,RFNS,RFNSMAX,
     1            RFNSMIN,CPWR,CRFNS,MTRLTYP,D1,D2,
     2            D3,D4,EXPWR,D5,NCASE,IDEG,1,2)
```

```
      CALL PROCS1(PWR,PWRMAX,PWRMIN,DRSIZ,DRSMAX,
   1              DRSMIN,CPWR,CDROS,MTRLTYP,D1,D2,
   2              D3,D4,EXPWR,D5,NCASE,IDEG,1,3)
      CALL PROCS1(PWR,PWRMAX,PWRMIN,TPR,TPRMAX,
   1              TPRMIN,CPWR,CTPR,MTRLTYP,D1,D2,
   2              D3,D4,EXPWR,D5,NCASE,IDEG,1,4)
      CALL PROCS1(PWR,PWRMAX,PWRMIN,HAZ,HAZMAX,
   1              HAZMIN,CPWR,CHAZ,MTRLTYP,D1,D2,
   2              D3,D4,EXPWR,D5,NCASE,IDEG,1,5)
C ... Search for suitable power of CW laser ...
      IF ((FLG3.EQ.1).OR.((IK.LT.1).AND.
   *      (PRCSCOD.EQ.1)))          GOTO 90
      DO 71 I=1,101
        IF (FEEDF(I).GE.FNLFD)   THEN
           FNLPWR = PWRF(I)
           GOTO 72
        ENDIF
71    CONTINUE
72    WRITE (OUT,99903)FNLPWR
C .............................................
75    IF (PLSTAT.EQ.'CW')  GOTO 90
C ... Evaluate in terms of PULSE ENERGY ...
80    IF (FLG8.EQ.1)  GOTO 85
      WRITE (LOGF,99898)
      CALL GRFSTT (CPLSEN)
      CALL PROCS1(PLSEN,PLSEMX,PLSEMN,KERF,KRFMAX,
   1              KRFMIN,CPLSEN,CKERF,MTRLTYP,D1,D2,
   2              D3,D4,EXENGY,D5,NCASE,IDEG,1,1)
      CALL PROCS1(PLSEN,PLSEMX,PLSEMN,RFNS,RFNSMAX,
   1              RFNSMIN,CPLSEN,CRFNS,MTRLTYP,D1,D2,
   2              D3,D4,EXENGY,D5,NCASE,IDEG,1,2)
      CALL PROCS1(PLSEN,PLSEMX,PLSEMN,DRSIZ,DRSMAX,
   1              DRSMIN,CPLSEN,CDROS,MTRLTYP,D1,D2,
   2              D3,D4,EXENGY,D5,NCASE,IDEG,1,3)
      CALL PROCS1(PLSEN,PLSEMX,PLSEMN,TPR,TPRMAX,
   1              TPRMIN,CPLSEN,CTPR,MTRLTYP,D1,D2,
   2              D3,D4,EXENGY,D5,NCASE,IDEG,1,4)
      CALL PROCS1(PLSEN,PLSEMX,PLSEMN,HAZ,HAZMAX,
   1              HAZMIN,CPLSEN,CHAZ,MTRLTYP,D1,D2,
   2              D3,D4,EXENGY,D5,NCASE,IDEG,1,5)
C ... Evaluate in terms of PRF....
85    IF (FLG6.EQ.1)  GOTO 90
      WRITE (logf,99895)
      CALL GRFSTT(CPRF)
      CALL PROCS1(PRF,PRFMAX,PRFMIN,KERF,KRFMAX,
   1              KRFMIN,CPRF,CKERF,MTRLTYP,D1,D2,
   2              D3,D4,EXPRF,D5,NCASE,IDEG,1,1)
      CALL PROCS1(PRF,PRFMAX,PRFMIN,RFNS,RFNSMAX,
   1              RFNSMIN,CPRF,CRFNS,MTRLTYP,D1,D2,
   2              D3,D4,EXPRF,D5,NCASE,IDEG,1,2)
      CALL PROCS1(PRF,PRFMAX,PRFMIN,DRSIZ,DRSMAX,
   1              DRSMIN,CPRF,CDROS,MTRLTYP,D1,D2,
   2              D3,D4,EXPRF,D5,NCASE,IDEG,1,3)
      CALL PROCS1(PRF,PRFMAX,PRFMIN,TPR,TPRMAX,
   1              TPRMIN,CPRF,CTPR,MTRLTYP,D1,D2,
   2              D3,D4,EXPRF,D5,NCASE,IDEG,1,4)
```

```
        CALL PROCS1(PRF,PRFMAX,PRFMIN,HAZ,HAZMAX,
     1              HAZMIN,CPRF,CHAZ,MTRLTYP,D1,D2,
     2              D3,D4,EXPRF,D5,NCASE,IDEG,1,5)
C ... Compute final answers for PULSED laser ...
        IF (FLG3.EQ.1)   GOTO 90
        DO 82 I=1,101
          IF (FEEDF(I).GE.FNLFD)   THEN
             FNLPRF = PRFF(I)
             GOTO  83
          ENDIF

82      CONTINUE

83      WRITE (OUT,99893)FNLPRF
C ..............................................
C ...Evaluate for GAS-PRESSURE.....
90      IF (FLG4.EQ.1)  THEN
          WRITE (OUT,99885) GSPMAX
          GOTO 300
        ENDIF

        WRITE (LOGF,99875)
        CALL GRFSTT(CGASP)

        CALL PROCS1(GASP,GSPMAX,GSPMIN,KERF,KRFMAX,
     1              KRFMIN,CGASP,CKERF,MTRLTYP,
     2              D1,D2,D3,D4,D6,D6,NCASE,IDEG,1,1)
        CALL PROCS1(GASP,GSPMAX,GSPMIN,RFNS,RFNSMAX,
     1              RFNSMIN,CGASP,CRFNS,MTRLTYP,
     2              D1,D2,D3,D4,D6,D6,NCASE,IDEG,1,2)
        CALL PROCS1(GASP,GSPMAX,GSPMIN,DRSIZ,DRSMAX,
     1              DRSMIN,CGASP,CDROS,MTRLTYP,
     2              D1,D2,D3,D4,D6,D6,NCASE,IDEG,1,3)
        CALL PROCS1(GASP,GSPMAX,GSPMIN,TPR,TPRMAX,
     1              TPRMIN,CGASP,CTPR,MTRLTYP,
     2              D1,D2,D3,D4,D6,D6,NCASE,IDEG,1,4)
        CALL PROCS1(GASP,GSPMAX,GSPMIN,HAZ,HAZMAX,
     1              HAZMIN,CGASP,CHAZ,MTRLTYP,
     2              D1,D2,D3,D4,D6,D6,NCASE,IDEG,1,5)
C ...................................................
C ... Write the results to the grafic screen ....
300     REWIND  OUT
        CALL PICCLE
        CALL MOVTO2 (30.,30.)
        CALL LINTO2 (30.,280.)
        CALL LINTO2 (200.,280.)
        CALL LINTO2 (200.,30.)
        CALL LINTO2 (30.,30.)
        CALL CHASIZ (2.,2.)
        CALL MOVTO2 (40.,270.)
        CALL DATE (DAT)
        CALL CHASTR (DAT // '*.')
        CALL MOVTO2 (175.,270.)
        CALL TIME (TIM)
        CALL CHASTR (TIM // '*.')
        CALL CHASIZ (2.5,2.5)
```

```
      DO 310 I=1,30
        CALL MOVTO2 (50.,(260.-(I*6.)))
        READ (OUT,320,END=400) ITST,STMT
        CALL CHASTR (STMT // '*.')
310   CONTINUE
320   FORMAT (I2,A)
C ... Close the graphics device.
400   CALL DEVEND
      STOP
C ..................................................
00001 FORMAT (/////////,12X,' *** LASER PARAMETERS PREDICTION'
     1 ' PROGRAM ***'//,17X,'THIS PROGRAM IS FOR THE ANALYSIS'
     2 'AND'/,17X,'PREDICTION OF PROCESS PARAMETERS OF'/,17X,
     3 'LASER CUTTING.   IT WORKS WITH THE'/,17X,'DATA'
     4 'SUPPLIED BY THE COLLECTION PROGRAM.'/,14X,
     5 ' ******************************************'/,17X,
     6 'IT WAS WRITTEN BY MOHAMMED A. HUSSAIN,'/,17X,
     7 'IN THE DEPT OF MECHANICAL ENGINEERING,'/,17X,
     8 'THE UNIVERSITY OF GLASGOW, IN 1988.'//////)
99999 FORMAT (/' * INPUT DATA FILE IS ....... ',A,$)
99998 FORMAT (A)
99997 FORMAT (/' *** INPUT DATA FILE NOT FOUND ***',A)
99996 FORMAT (/' * OUTPUT DATA FILE (WITHOUT',
     *         ' EXTENSION) IS ... ',A,$)
99995 FORMAT (/' * THE DESIRED DEGREE OF THE ',
     *         'POLYNOMIAL IS ....',A,$)
99994 FORMAT (/' * SELECT OUTPUT DEVICE * 1=4010, ',
     1        '2=BENSON, 3=VERSATEC, 4=TTB6',A,//
     2        '  * ENTER OPTION CODE...',$)
99993 FORMAT (18X'SUMMARY OF RESULTS'/,18X,
     *            '------------------')
99992 FORMAT (/'  MATERIAL CLASS IS ',A25,/
     1          '  MATERIAL TYPE IS ',A25,/
     2          '  MATERIAL THICKNESS IS',I7,' micron')
99990 FORMAT (/'  ALL THE DATA THAT WAS COLLECTED',
     1         ' FROM THE'/'  DATABASE ARE OF THE ',
     2         'SAME THICKNESS AS THE'/'  TARGET',
     3        ' MATERIAL WHICH IS',F6.0,' microns.')
99988 FORMAT (/'  AS THE TARGET MATERIAL THICKNESS HAS',
     1         ' NOT BEEN FOUND'/'  IN THE DATABASE, ',
     2         'INTERPOLATION/EXTRAPOLATION WILL BE'/
     3         '  USED TO PREDICT APPROXIMATE SETTINGS.')
99985 FORMAT (/'  CANNOT PREDICT FOR THE REQUIRED ',
     1         'THICKNESS SINCE THE'/'  SUPPLIED DATA ARE ',
     2         'FOR ONE THICKNESS ONLY, i.e',F6.0,' micron.')
99980 FORMAT (/'  CANNOT EVALUATE FOR FEED RATE ',
     *       'SINCE IT IS CONSTANT AT ',F6.0,' mm/min.')
99975 FORMAT (/'  CANNOT EVALUATE IN TERMS OF FEED',
     1         ' NOR POWER SINCE'/'  THEY ARE CONSTANT ',
     2     'AT',F6.0,'mm/min AND',F6.0,'W. RESPECTIVELY.')
99965 FORMAT (/'  CANNOT EVALUATE IN TERMS OF POWER'/
     1         ,'  SINCE IT IS CONSTANT AT',F6.0,'W.')
99955 FORMAT (/'  CANNOT EVALUATE FOR PRF NOR FOR ',
     1         'PULSE ENERGY'/'  SINCE BOTH ARE ',
     2         'CONSTANT AT',F5.0,'HZ AND',F6.0,'mJ.')
```

```
99945   FORMAT (/'  CANNOT EVALUATE FOR PULSE ENERGY'/
     1           ,'  SINCE IT IS CINSTANT AT',F7.0'mJ.')
99943   FORMAT (/'  THE RECOMMENDED PULSE ENERGY IS',
     1           F7.0,'mJ.')
99935   FORMAT (/'  THE MEAN PULSE ENERGY IS',F7.0,'mJ.'
     *   /'  WHILE THE MIN. RECORDED ENERGY IS',F7.0,'mJ.')
99925   FORMAT (/'  CANNOT EVALUATE FOR PRF SINCE ',
     *           'IT IS CONSTANT AT',F4.0,' HZ.')
99915   FORMAT (/'  EVALUATION FOR FEED RATE ')
99914   FORMAT (/'  CUT IS IMPOSSIBLE ACCORDING TO THIS ',
     1           'ANALYSIS.'/'  REVIEW (FEED X THICK.) GRAPH.'
     2           ,' LOWER DEGREE'/'  POLYNOMIAL COULD BE ',
     3           'MORE HELPFUL.')
99913   FORMAT (/'  FEEDS THAT CAN ACTUATE MIN. KERF,',
     1           'ROUGHNESS,DROSS,'/'  TAPER & HAZ ARE ',
     2           ',RESPECTIVELY, AS FOLLOWS : '/'  ',5F9.0)
99912   FORMAT (/'  NO PREFERENCE HAS BEEN GIVEN TO ',
     *           'QUALITY FEATURES.')
99911   FORMAT (/'  WITH PREFERENCE WEIGHTS OF :'/,2X,5F9.2,
     *           /'  WHILE THE PREFERENCE WEIGHT FOR MAXIMUM',
     2           ' FEED IS ',F4.2)
99910   FORMAT (/'  THE RECOMMENDED FEED IS ',F6.0,'mm/min.')
99905   FORMAT (/'  EVALUATION FOR LASER POWER ')
99903   FORMAT (/'  THE RECOMMENDED POWER IS ',F6.0,'Watts.')
99898   FORMAT (/'  EVALUATION IN TERMS OF PULSE ENERGY')
99895   FORMAT (/'  EVALUATION FOR PULSE REP. RATE ')
99893   FORMAT (/'  THE RECOMMENDED PRF IS ',F6.0,'Hz.')
99885   FORMAT (/'  CANNOT EVALUATE FOR NOZZLE GAS ',
     1           'PRESSURE'/'  SINCE IT IS CONSTANT AT',
     2           F9.0,' mbar.')
99875   FORMAT (/'  EVALUATION FOR NOZZLE GAS PRESSURE ')
        END
```

```
      SUBROUTINE PROCS1(X,XMAX,XMIN,Y,YMAX,YMIN,TTLX,
     1                  TTLY,MTRLTYP,XI,F,DSTX,DSTY,
     2                  BEXT,AEXT,NCASE,IDEG,KDST,COD)
C ******************************************************
C ... Routine for curve fitting to data points.
C ... Created on the 5th SEP. 1988
C ... External parameters
      INTEGER    NCASE,KDST,COD
      REAL*8     X(NCASE),XMAX,XMIN,Y(NCASE),YMAX,YMIN,
     1            DSTX(KDST),DSTY(KDST),XI(101),F(101),
     2            BEXT,AEXT
C ...
      CHARACTER*(*)    TTLX,TTLY,MTRLTYP
C ... Local parameters.
      INTEGER    VIN,VOUT,IN,LOGF,IFAIL,KPLUS1,NROWS,
     *            IDEG,M,IRANK(900)
C ...
      REAL*8     TEMP(900,2),W(900),XCAP,P,S(10),MRGN,
     1            A(10,10),DELTAX,DELTAY,RAV,R(30),
     2            AA(10),WORK1(3000),WORK2(10),AMAX,
     3            AMIN,BMAX,BMIN,XMISS,YMISS,RESULT(101)
C ...
      CHARACTER   BEL,UNITX*10, UNITY*10
C ...
      DATA  VIN,VOUT,IN,LOGF,BEL/5,6,10,15,7/
C ... Initiating local parameters.
      M=NCASE       ! No. of data rows .
      IDEG1=IDEG
      XMISS = 0.0000000
      YMISS = 0.0000000
C ... Check for variations in Y-ordinates....
      IF (YMAX.EQ.YMIN)     THEN
        CALL UNITS(TTLX,TTLY,UINTX,UNITY)
        WRITE (LOGF,9999)   TTLY,YMAX,UNITY
        AEXT = XMAX
C ... Write a message to the grafics screen if
C ... all Y-axis data is constant.
      CALL CHASIZ (2.5,2.5)
      CALL MOVTO2 (60.,(17.+(COD*47.)))
      CALL CHASTR ('THIS FEATURE IS CONSTANT AT*.')
      CALL MOVTO2 (132.,(17.+(COD*47.)))
      CALL CHAFIX (YMAX,-7,0)
      CALL MOVTO2 (150.,(17.+(COD*47.)))
      CALL CHASTR (UNITY // '*.')
      CALL MOVTO2 (40.,(7.+(COD*47.)))
      CALL CHAANG (90.)
      CALL CHASTR (TTLY //'*.')
      CALL MOVTO2 (50.,((40.+COD*47.)))
      CALL LINTO2 (50.,((COD*47.)-7.))
      CALL LINTO2 (190.,((COD*47.)-7.))
      CALL CHAANG (0.)
        RETURN
      ENDIF
C ...
```

```
C ... Check for out-of-range X-data ...
        IF (BEXT.LE.0)    GOTO 8
        IF ((BEXT.GT.XMAX).OR.(BEXT.LT.XMIN)) THEN
          CALL G02CCF(M,X,Y,XMISS,YMISS,RESULT,IFAIL)
        ELSE
          GOTO 8
        ENDIF
        IF (IFAIL)3,5,3
3       WRITE (LOGF,9977) IFAIL
        RETURN
5       M = M+1
        X(M) = BEXT
        Y(M) = RESULT(7)+(RESULT(6)*X(M))
        AEXT = Y(M)
C ... Convert input data into single 2D array for
C ... sorting by the M01... routines.
8       DO 10 I=1,M
           TEMP(I,1)=X(I)
           TEMP(I,2)=Y(I)
10      CONTINUE
C ... Rank the data into ascending order ...
        IFAIL=1
        CALL M01DAF(TEMP(1,1),1,M,'A',IRANK,IFAIL)
        IF (IFAIL.EQ.0)   GOTO 20
        WRITE (LOGF,9998) IFAIL
        RETURN
C ... Sort the ranked data ...
20      IFAIL=1
        DO 30 I=1,2
           CALL M01EAF(TEMP(1,I),1,M,IRANK,IFAIL)
30      CONTINUE
C ... Identify the additinal data point,and reduce
C ... its weight so as not to affect the fit badly.
        DO 32 I=1,M
           W(I) = 1.0              !Initiation
           IF ((TEMP(I,1).LT.XMIN).OR.
     *        (TEMP(I,1).GT.(XMAX)))   W(I)=0.5
           IF ((TEMP(I,2).LE.0.).AND.(COD.NE.4))  W(I)=0.01
C ... Specify Y-axis bounries ....
           IF (TEMP(I,2).GT.YMAX)  AMAX=TEMP(I,2)
           IF (TEMP(I,2).LE.YMAX)  AMAX=YMAX
           IF (TEMP(I,2).LE.YMIN)  AMIN=TEMP(I,2)
           IF (TEMP(I,2).GT.YMIN)  AMIN=YMIN
32      CONTINUE
C ... Specify X-axis boundaries.
        BMAX = TEMP(M,1)
        BMIN = TEMP(1,1)
C ... Call the polynomial curve fitting routines.
50      KPLUS1=IDEG1+1
        NROWS=10
C ................................................
        WRITE (LOGF,9996) IDEG1
        IFAIL=1
        CALL E02ADF (M,KPLUS1,NROWS,TEMP(1,1),TEMP(1,2),
     *               W,WORK1,WORK2,A,S,IFAIL)
        IF (IFAIL.NE.0)      GOTO 120
```

```
C ... Set the polynomial coefficients for the E02AEF.
        DO 60 K=1,KPLUS1
           AA(K)=A(KPLUS1,K)
60      CONTINUE
C ... Create 101 new X values for evaluation
C ... by the E02AEF routine.
        DELTAX=BMAX-BMIN
        DO 80 I=1,101
          II=I-1
          XI(I)=BMIN+(II*0.01*DELTAX)
          XCAP=((XI(I)-BMIN)-(BMAX-XI(I)))/DELTAX
          IFAIL=1
          CALL E02AEF (KPLUS1,AA,XCAP,P,IFAIL)
          IF (IFAIL.NE.0)   GOTO 100
          F(I)=P
80      CONTINUE
C ... Check for FEED Vs THICK fitting ...
        IF ((TTLX.EQ.'THICKNESS').AND.
     *     (TTLY.EQ.'FEED'))    GOTO  85
        GOTO  200
C ... Compute the margin of the max.Feed ...
85      RAV=0.0
        DO 90 I=1,KDST
          XCAP=((DSTX(I)-BMIN)-(BMAX-DSTX(I)))/DELTAX
          IFAIL=1
          CALL E02AEF (KPLUS1,AA,XCAP,P,IFAIL)
          IF (IFAIL.NE.0)   GOTO 100
          R(I)=(DSTY(I)-P)/P
          RAV=RAV+R(I)
90      CONTINUE
        RAV=RAV/KDST      !The average ratio
C ... Add the marginal value to the originally fitted value.
        DO 95 I=1,101
          F(I) = F(I)+(RAV*F(I))
95      CONTINUE
        GOTO 200
100     WRITE (LOGF,9992) IFAIL
        RETURN
120     GOTO (140,140,140,160,140), IFAIL
140     WRITE (LOGF,9990) IFAIL,BEL
        RETURN
C ... If ifail=4 reduce the polynomial degree and restart
160     WRITE (LOGF,9988) IFAIL,BEL
        IDEG1=IDEG1-1
        GOTO 50
C ..............................................
C ... Identify the current process ...
200     IF (COD.EQ.0)  GOTO 300
C ... Call the plotting routine.
        CALL GRAF3(TEMP(1,2),YMAX,YMIN,TEMP(1,1),
     *       XMAX,XMIN,F,XI,M,COD)
C ... Test for target FEEDs ....
        IF (TTLX.NE.'FEED')  GOTO 250
C ... Convert input data into single 2D array for
C ... sorting by M01...routines.
```

```
        DO 210 I=1,101
           TEMP(I,1)=XI(I)
           TEMP(I,2)=F(I)
210     CONTINUE
C ... Rank the data into ascending order ...
        IFAIL=1
        CALL M01DAF(TEMP(1,2),1,101,'A',IRANK,IFAIL)
        IF (IFAIL.EQ.0)   GOTO 220
        WRITE (LOGF,9998) IFAIL
        RETURN
C ... Sort the ranked data ...
220     IFAIL=1
        DO 230 I=1,2
           CALL M01EAF(TEMP(1,I),1,101,IRANK,IFAIL)
230     CONTINUE
C ... Test the first 10 points for target FEED ...
        AEXT = TEMP(1,1)  !Initiate
        MRGN =.01*(TEMP(101,2)-TEMP(1,2))  !Tolarence
        DO 240 I=2,10
          IF (TEMP(I,2).LE.(TEMP(I-1,2)+MRGN)) THEN
            IF (TEMP(I,1).GT.TEMP(I-1,1))
     *           AEXT = TEMP(I,1)
          ENDIF
240     CONTINUE
250     RETURN
C ..............................................
300     CALL GRAF1(TEMP(1,2),AMAX,AMIN,TEMP(1,1),
     *       BMAX,BMIN,F,XI,TTLY,TTLX,MTRLTYP,M,KTER)
C ..............................................
320     RETURN
C ...
9999    FORMAT (/'  *THE ',A12,' IS CONSTATNT AT ',F8.0,
     *           ' ',A10,/'    THROUGHOUT THE INPUT DATA*'/)
9998    FORMAT (/' *M01DAF FAILURE CODE   ',I4)
9996    FORMAT (/' *THE DEGREE OF THE POLYNOMIAL= ',I2,/)
9992    FORMAT (/' *E02AEF FAILURE CODE   ',I4,A)
9990    FORMAT (/' *E02ADF FAILURE CODE   ',I4,A)
9988    FORMAT (/' *E02ADF FAILURE CODE   ',I3,
     *         '  RESTARTING WITH LOWER DEGREE',A)
9977    FORMAT (/' *G02CCF FAILURE CODE   ',I3)
        END
```

```
      SUBROUTINE PROCS2(X,Y,F,XEXT,TTLZ,TTLX,TTLY,
     1                   MTRLCLS,MTRLTYP,NCASE,KTER)
C *******************************************************
C ...Routine for surface fitting to data points.
C ... External parameters
      INTEGER   NCASE
      REAL*8    X(NCASE),Y(NCASE),F(NCASE),XEXT
      CHARACTER*(*)   TTLX,TTLY,TTLZ,MTRLCLS,MTRLTYP
C ... Local parameters.
      INTEGER   IADRES,IFAIL,NC,NP,NWS,PX,PY,
     1          QX,QY,RANK,ADRES(50),POINT(500),
     2          VIN,VOUT,IN,LOGF
C ...
      REAL*8    EPS,SIGMA,BMAX,BMIN,TMP1(21),
     1          DELTAA,DELTAB,XI,AMAX,AMIN,
     2          C(80),DL(300),FF(400),LAMDA(15),
     3           MU(15),W(450),WS(15000),XVAL(4),
     4          SX(400),SY(400),Y1(21),Y2(21),
     5           Y3(21),Y4(21),Z(21),TMP2(21),
     6           AX(441),AY(441),AZ(441),TMP3(21)
C ...
      CHARACTER   BEL
C ...
      DATA   VIN,VOUT,IN,LOGF /5,6,10,15/
      DATA   NPNT,EPS,W /500,1.E-12,450*1.0/
      DATA   NX,NY,BEL /0,0,7/   !Initialized
C ... Initialising the local parameters.
      NWS=15000
      M=21*21         ! No. of data rows .
      PX=8
      PY=8
      SIGMA=0.0
      RANK=0
      AMAX=-1.0E15
      BMAX=-1.0E15
      AMIN=-AMAX
      BMIN=-BMAX
C ... Reverse the F-Axis, and reduce No. of points.
      DO 2 I=1,21
         II = I-1
         TMP1(I)=X(1+(II*5))
         TMP2(I)=Y(1+(II*5))
         TMP3(I)=F(1+(II*5))
2     CONTINUE
      DO 3 I=1,21
         F(I)=TMP3(22-I)
         X(I)=TMP1(I)
         Y(I)=TMP2(I)
3     CONTINUE
C ... Compose new data array (21*21) for evaluation..
      K=1
      DO 4 I=1,21
         II = I-1
        DO 4 J=1,21
          AX(K)=X(I)
          AY(K)=Y(J)
```

```
            KK=(K-(21*II))
            IF (KK.LT.I)    THEN
              AZ(K)=0.0
            ELSE
              AZ(K)=F(J-II)
              IF (AZ(K).LT.0.)  AZ(K)=0.0
            ENDIF
            K=K+1
4       CONTINUE
C ... Compute the max. & min. of the input data.
        DO 10 I=1,441
          IF (AX(I).GT.BMAX)  BMAX=AX(I)
          IF (AX(I).LT.BMIN)  BMIN=AX(I)
          IF (AY(I).GT.AMAX)  AMAX=AY(I)
          IF (AY(I).LT.AMIN)  AMIN=AY(I)
10      CONTINUE
C ... Compute the X & Y intervals...
        DELTAA=AMAX-AMIN   ! Real interval of X
        DELTAB=BMAX-BMIN   ! Real interval of Y
C ...............................................
C .. This block currently not active because both PX
C .. & PY are set to 8 (i.e the whole X-Y plane is one
C .. panel.  This block is included in case the user
C .. wants to try multi-panel evaluation, by assigning
C .. higher values to PX and PY.
C
C ... This is a measure to avoid failure 2 at the NAG
C ... routines E02DAF & E02DBF...
        IF (DELTAA.EQ.0.0)   PY=8
        IF (DELTAB.EQ.0.0)   PX=8
C ...  Compute LAMDA'S so that  the full  interval
C ...  of X is devided into  ? portins .
C ..
20      IF (PX.GT.8)    THEN
           NX=PX-8             !No. of subintervals.
           DO 25 I=1,NX
              LAMDA(I+4)=BMIN+(I*DELTAB/(NX+1))
25         CONTINUE
        ENDIF
C ...  Compute MU's so that  the full  interval
C ...  is devided into a number of subintervals.
C ..
        IF (PY.GT.8)    THEN
           NY=PY-8
           DO 30 I=1,NY
              MU(I+4)=AMIN+(I*DELTAA/(NY+1))
30         CONTINUE
        ENDIF
C ...............................................
        QX=PX-4
        QY=PY-4
        NC=QX*QY
        NP=(PX-7)*(PY-7)
C
```

```
      WRITE (LOGF,7775)PX,PY
      WRITE (LOGF,9986)
      WRITE (LOGF,9988)  (LAMDA(I), I=5,QX)
      WRITE (LOGF,9985)
      WRITE (LOGF,9987)  (MU(I), I=5,QY)
      IFAIL=1
      CALL E02ZAF(PX,PY,LAMDA,MU,M,AX,AY,POINT,NPNT,
     *             ADRES,NP,IFAIL)
      IF(IFAIL.EQ.0) GO TO 40
      WRITE (LOGF,9993) IFAIL,BEL
      STOP
C ..................................................
C ... Test for empty panels.
C ...
40       J=0
      DO 45 I=1,NP
         IADRES=I+M
42       IADRES=POINT(IADRES)
         IF (IADRES.EQ.0)  J=J+1
45    CONTINUE
C     WRITE (LOGF,*) (' ** No.of empty panels =',j)
C ... If No. empty panels one or nil.
      IF (J.LE.1)      GOTO 60
C ... But if there are some .
      IF (J.GT.5)     THEN
         PX=PX-1
         PY=PY-1
         GOTO   20
      ELSEIF ((J.LE.5).AND.(PX.GT.PY)) THEN
             PX=PX-1
             GOTO  20
      ELSEIF ((J.LE.5).AND.(PX.LE.PY)) THEN
             PY=PY-1
      ENDIF
C ... Do not allow PY to be less than 8 .
      IF (PY.LE.8)    PY=8
      GOTO 20  !Repeat the panelizing process.
C ..................................................
60    DO 70 I=1,NP
        WRITE (LOGF,7777) I
        IADRES=I+M
68      IADRES=POINT(IADRES)
        IF (IADRES.GT.0) THEN
          WRITE (LOGF,7776) AX(IADRES) , AY(IADRES),
     *            AZ(IADRES)
          GOTO 68
        ENDIF
70    CONTINUE
C ...................................................
      IFAIL=1
      CALL E02DAF(M,PX,PY,AX,AY,AZ,W,LAMDA,MU,POINT,NPNT,
     *            DL,C,NC,WS,NWS,EPS,SIGMA,RANK,IFAIL)
      IF (IFAIL.EQ.0) GO TO 85
      WRITE (LOGF,9995) IFAIL,BEL
      WRITE (VOUT,9995) IFAIL,BEL
C ..................................................
```

```
C ... Creating new array of data points for evaluation
C ... by E02DBF.
85       WRITE (LOGF,9992) SIGMA
         WRITE (LOGF,9991) RANK
         K=1
         DO 110 I=1,3
            II=I-1
            XI=BMIN+(II*0.5*DELTAB)
              DO 100 J=1,21
                  JJ=J-1
                  SY(K)=AMIN+(JJ*0.05*DELTAA)
                  SX(K)=XI
                  K=K+1
100           CONTINUE
110      CONTINUE
C ... Evaluate for thickness range .
         DO 120 I=1,101
           SX(I+63) = XEXT
           SY(I+63) = AMIN+((I-1)*.01*DELTAA)
           Y(I) = SY(I+63)
120      CONTINUE
C .........................................
         NCOL=((3*21)+101)    ! No. of the data points.
         IFAIL=1
         CALL E02ZAF(PX,PY,LAMDA,MU,NCOL,SX,SY,POINT,NPNT,
     *               ADRES,NP,IFAIL)
         IF(IFAIL.EQ.0) GO TO 220
         WRITE (LOGF,9993) IFAIL,BEL
         WRITE (VOUT,9993) IFAIL,BEL
         STOP
C ..........................................
C ..     220      DO 222 I=1,NP
C          WRITE (LOGF,7777) I
C          IADRES=I+NCOL
C        221         IADRES=POINT(IADRES)
C          IF (IADRES.GT.0) THEN
C            WRITE (LOGF,7774) SX(IADRES) , SY(IADRES)
C            GOTO 221
C         ENDIF
C        222      CONTINUE
220      IFAIL=1
         CALL E02DBF(NCOL,PX,PY,SX,SY,FF,LAMDA,MU,POINT,
     *               NPNT,C,NC,IFAIL)
         IF (IFAIL.EQ.0) GO TO 240
         WRITE (LOGF,9994) IFAIL,BEL
         WRITE (VOUT,9994) IFAIL,BEL
         GOTO 320
C  240   WRITE (LOGF,*) ' ******************************'
C        WRITE (LOGF,*) ' ORIGINAL DATA MATRIX :'
C        WRITE (LOGF,9989)  (AX(I),AY(I),AZ(I),I=1,M)
C        WRITE (LOGF,*) ' ******************************'
C ... Impose limits on the fitted data.
240      DO 250 I=1,NCOL
            IF (FF(I).LT.(1.0))  FF(I)=0.0
            IF (FF(I).GT.(15.E03))  FF(I)=15.E03
250      CONTINUE
```

```
C ... Target thickness curve points ...
      DO 260 I=1,101
        F(I) = FF(I+63)
        IF (F(I).LE.0.)  CALL DPNOIS (F,I)
260   CONTINUE
C ... Construct the graphical input data.
      DO 300 I=1,21
        XVAL(I)=SY(I)
        Y1(I)=FF(I)
        IF (Y1(I).LE.0.) CALL DPNOIS (Y1,I)
        Y2(I)=FF(I+21)
        IF (Y2(I).LE.0.) CALL DPNOIS (Y2,I)
        Y3(I)=FF(I+42)
        IF (Y3(I).LE.0.) CALL DPNOIS (Y3,I)
        Y4(I)=F(1+((I-1)*5))
300   CONTINUE
      DO I=1,4
         Z(I)=SX(I*21)
      END DO
C ... Output data as a matrix...
      WRITE (LOGF,9978)TTLZ,(Z(I),I=1,3)
      WRITE (LOGF,9976)  TTLX,TTLY,TTLY,TTLY
      WRITE (LOGF,8882) (XVAL(I),Y1(I),Y2(I),Y3(I),
     *       I=1,21)
C ..............................................
C ... Call the plotting routine.
      CALL GRAF2(Y1,Y2,Y3,Y4,XVAL,Z,TTLY,TTLX,TTLZ,
     *          MTRLCLS,MTRLTYP,KTER)
C ..............................................
320   RETURN
9995  FORMAT (23H E02DAF FAILURE NUMBER , I4,A)
9994  FORMAT (23H E02DBF FAILURE NUMBER , I4,A)
9993  FORMAT (23H E02ZAF FAILURE NUMBER , I4,A)
9992  FORMAT (/' SIGMA = ', E16.4)
9991  FORMAT (/' RANK = ', I5,//)
9989  FORMAT (3F9.0)
9988  FORMAT (<NX>F9.0)
9987  FORMAT (<NY>F9.0)
9986  FORMAT (/'  LAMDA VALUES ARE '/)
9985  FORMAT (/'  MU VALUES ARE '/)
9978  FORMAT (/,A9,5F9.0)
9976  FORMAT (4A9)
8888  FORMAT (I4)
8885  FORMAT (/'  FOR THE ',A,' OF ',F9.0,/1H,2A9)
8884  FORMAT (2F9.0)
8882  FORMAT (4F9.0)
7777  FORMAT (/' PANEL ',I4)
7776  FORMAT (' ',3F9.0)
7775  FORMAT (/' PX= ',I4,'    &   PY= ',I4)
7774  FORMAT (' ',2F9.0)
      END
```

```
      SUBROUTINE GRAF1(Y,YMAX,YMIN,X,XMAX,XMIN,
     *         A,B,TTLY,TTLX,MTRLTYP,NCASE,KTER)
C ***************************************************
C ... Routine to plot one curve and the original
C ... points. Created on the 5th SEP. 1988.
C ... External parameters.
      INTEGER   NCASE,KTER
      REAL*8    B(101),A(101),X(NCASE),XMAX,XMIN,
     *           Y(NCASE),YMAX,YMIN
C ...
      CHARACTER*(*) TTLY,TTLX,MTRLTYP
C ... Local parameters.
      INTEGER    NPOINT
      REAL*4     BPOSX, BPOSY, APOSX,APOSY,
     *           BTPOSX, BTPOSY, ATPOSX, ATPOSY
      REAL*4     AR(21),BR(21),XR(900),XRMAX,XRMIN,
     1            YR(900),YRMAX,YRMIN,AMAX,AMIN,
     2            BMAX,BMIN,DELTAA,DELTAB
C ...
      CHARACTER   TTLM*80,TTLX1*30,TTLY1*30,
     1             MTRLTYP1*27,BEL,STREND*2,UNITX*10,
     2             UNITY*10
C ....
      DATA BPOSX/50./     ! X coord of start of X-axis
      DATA BPOSY/90./     ! Y coord of start of X-axis
      DATA APOSX/50./     ! X coord of start of Y-axis
      DATA APOSY/90./     ! Y coord of start of Y-axis
      DATA BTPOSX/100./ ! X coord of start of X-title
      DATA BTPOSY/80./  ! Y coord of start of X-title
      DATA ATPOSX/40./  ! X coord of start of Y-title
      DATA ATPOSY/150./ ! Y coord of start of Y-title
      DATA BEL,NPOINT /7, 21/
C ...
      CALL PICCLE       !Start a new page.
C ... Compose the arrays for plotting...
      DO 20 I=1,NCASE
         XR(I)=X(I)
         YR(I)=Y(I)
20    CONTINUE
      XRMIN=XMIN
      YRMIN=YMIN
      XRMAX=XMAX
      YRMAX=YMAX
C ... Reduce No. of points to 21 ....
      DO 55 I=1,21
         II = I-1
         BR(I)=B(1+(II*5))
         AR(I)=A(1+(II*5))
55    CONTINUE
      AMAX=-1.E20
      BMAX=-1.E20
      AMIN=-AMAX
      BMIN=-BMAX
```

```
C ... Test for the min.&max. of A & B.
        DO 65, I=1,21
           IF (AR(I).GE.AMAX) AMAX=AR(I)
           IF (AR(I).LE.AMIN) AMIN=AR(I)
           IF (BR(I).GE.BMAX) BMAX=BR(I)
           IF (BR(I).LE.BMIN) BMIN=BR(I)
65      CONTINUE
C ... Test for the absolute max. & min. of the axes.
        IF (AMAX.GE.YRMAX)  AMAXT=AMAX
        IF (AMAX.LT.YRMAX)  AMAXT=YRMAX
        IF (AMIN.LE.YRMIN)  AMINT=AMIN
        IF (AMIN.GT.YRMIN)  AMINT=YRMIN
        IF (BMAX.GE.XRMAX)  BMAXT=BMAX
        IF (BMAX.LT.XRMAX)  BMAXT=XRMAX
        IF (BMIN.LE.XRMIN)  BMINT=BMIN
        IF (BMIN.GT.XRMIN)  BMINT=XRMIN
C ... Specify the plot boundaries.
        DELTAB=BMAXT-BMINT
        BMAXT=BMAXT+(.05*DELTAB)
        BMINT=BMINT-(.05*DELTAB)
        DELTAA=AMAXT-AMINT
        AMAXT=AMAXT+(.05*DELTAA)
        IF(AMINT.LE.0.)   THEN
             AMINT=AMINT-(.05*DELTAA)
        ELSE
             AMINT=0.0
        ENDIF
C... Start of the graph.
        CALL CHASIZ (2.,2.)
        CALL AXIPOS (1,BPOSX,BPOSY,140.,1)
        CALL AXISCA (1,10,BMINT,BMAXT,1)
        CALL AXIDRA (-2,1,1)
        CALL AXIPOS (1,APOSX,APOSY,140.,2)
        CALL AXISCA (1,10,AMINT,AMAXT,2)
        CALL AXIDRA (2,-1,2)
C ... complete the graph box .
        CALL MOVTO2 (50.,230.)
        CALL LINTO2 (190.,230.)
        CALL LINTO2 (190.,90.)
C ... Plot the oringinal points...
        CALL GRASYM (XR,YR,NCASE,8,0)
C ... Plot the fitting curve...
        CALL GRAPOL (BR,AR,NPOINT)
C .......... Start lettering.
C ...Call for the  routine
        CALL UNITS(TTLX,TTLY,UNITX,UNITY)
        TTLX1=TTLX // UNITX
        TTLY1=TTLY // UNITY
C ... Terminate the string ...
        STREND='*.'
        TTLX1=TTLX1 // STREND
        TTLY1=TTLY1 // STREND
        MTRLTYP1= MTRLTYP // STREND
C ...
```

```
      CALL MOVTO2 (ATPOSX,ATPOSY)
      CALL CHASIZ (2.5,2.5)
      CALL CHAANG (90.)
      CALL CHASTR (TTLY1)
      CALL MOVTO2 (BTPOSX,BTPOSY)
      CALL CHAANG (0.)
      CALL CHASTR (TTLX1)
C ..............................................
C ... write class & type of the material.
C ...
C     CALL MOVTO2 (55.,55.)
C     CALL CHAHOL ('THE MATERIAL IS > *.')
C     CALL MOVTO2 (100.,55.)
C     CALL CHASTR (MTRLTYP1)
C ... Write the main title.
79    CALL CHASIZ (2.5,2.5)
      CALL MOVTO2 (75.,20.)
      TTLM=TTLY//' Vs '//TTLX//' FOR '//MTRLTYP//STREND
      CALL CHASTR (TTLM)
1000  RETURN
      END
```

```
      SUBROUTINE GRAF2(A1,A2,A3,A4,B,Z,TTLY,TTLX,
     *                     TTLZ,MTRLCLS,MTRLTYP,KTER)
C ******************************************************
C ... Routine to plot five curves.
C ... Created on the 5th SEP. 1988.
C ... External parameters.
      REAL*8   B(21),A1(21),A2(21),A3(21),
     *          A4(21),Z(21)

      CHARACTER*(*) TTLY,TTLX,TTLZ,MTRLCLS,MTRLTYP

C ... Local parameters.

      INTEGER   NPOINT(4)

      REAL*4    BPOSX, BPOSY, APOSX,APOSY,
     1          BTPOSX, BTPOSY, ATPOSX, ATPOSY,
     2           DELTAA,DELTAB,AR1(21),AR2(21),
     3           AR3(21),AR4(21),BR(21)
C ...
      CHARACTER   TTLM*80,TTLX1*15,TTLY1*15,TTLZ1*20,
     1            MTRLCLS1*27,MTRLTYP1*27,BEL,STREND*2,
     2             UNITY*10,UNITX*10
C ....
      DATA BPOSX/50./      ! X coord of start of X-axis
      DATA BPOSY/90./      ! Y coord of start of X-axis
      DATA APOSX/50./      ! X coord of start of Y-axis
      DATA APOSY/90./      ! Y coord of start of Y-axis
      DATA BTPOSX/100./ ! X coord of start of X-title
      DATA BTPOSY/80./  ! Y coord of start of X-title
      DATA ATPOSX/40./  ! X coord of start of Y-title
      DATA ATPOSY/150./ ! Y coord of start of Y-title
      DATA BEL,NPOINT /7, 4*21/
C .....................................................
      CALL PICCLE       !Start a new page.

      DO 55 I=1,21
         BR(I)=B(I)
         AR1(I)=A1(I)
         AR2(I)=A2(I)
         AR3(I)=A3(I)
         AR4(I)=A4(I)
55    CONTINUE

      AMAXT=-1.E20
      BMAXT=-1.E20
      AMINT=-AMAXT
      BMINT=-BMAXT

C ... Test for the min.&max. of x & y.

      DO 65, I=1,21
         IF (AR1(I).GT.AMAXT) AMAXT=AR1(I)
         IF (AR1(I).LT.AMINT) AMINT=AR1(I)
         IF (AR2(I).GT.AMAXT) AMAXT=AR2(I)
         IF (AR2(I).LT.AMINT) AMINT=AR2(I)
```

```
            IF (AR3(I).GT.AMAXT) AMAXT=AR3(I)
            IF (AR3(I).LT.AMINT) AMINT=AR3(I)
            IF (AR4(I).GT.AMAXT) AMAXT=AR4(I)
            IF (AR4(I).LT.AMINT) AMINT=AR4(I)
            IF (BR(I).GT.BMAXT) BMAXT=BR(I)
            IF (BR(I).LT.BMINT) BMINT=BR(I)
65      CONTINUE

C ... Specify the graph boudaries.
        DELTAA=AMAXT-AMINT
        DELTAB=BMAXT-BMINT
        AMAXT=AMAXT+(.05*DELTAA)
        AMINT=AMINT-(.05*DELTAA)
        BMAXT=BMAXT+(.05*DELTAB)
        BMINT=BMINT-(.05*DELTAB)

C ... Start of the graph.
        CALL CHASIZ (2.,2.)
        CALL AXIPOS (1,BPOSX,BPOSY,140.,1)
        CALL AXISCA (1,10,BMINT,BMAXT,1)
        CALL AXIDRA (-2,1,1)
        CALL AXIPOS (1,APOSX,APOSY,140.,2)
        CALL AXISCA (1,10,AMINT,AMAXT,2)
        CALL AXIDRA (2,-1,2)

C ... complete the graph box .
        CALL MOVTO2 (50.,230.)
        CALL LINTO2 (190.,230.)
        CALL LINTO2 (190.,90.)

C ... Plot the first curve.
        CALL GRASYM (BR,AR1,NPOINT(1),7,4)
        CALL GRAPOL (BR,AR1,NPOINT(1))
C ... Plot the second curve.
        IF(NPOINT(2).EQ.0)  GOTO 69
         CALL GRASYM (BR,AR2,NPOINT(2),6,4)
C        CALL DASHED (1,3.5,3.,0.)
         CALL GRAPOL (BR,AR2,NPOINT(2))
C ... Plot the third curve.
        IF(NPOINT(3).EQ.0)GOTO69
         CALL GRASYM (BR,AR3,NPOINT(3),5,4)
C        CALL DASHED (1,2.5,2.,0.)
         CALL GRAPOL (BR,AR3,NPOINT(3))
C ... Plot the forth curve.
        IF(NPOINT(4).EQ.0)GOTO69
         CALL GRASYM (BR,AR4,NPOINT(4),2,4)
C        CALL DASHED (1,1.5,1.,0.)
         CALL GRAPOL (BR,AR4,NPOINT(4))

C ... End of plotting.
69       CALL DASHED(0)

C ... Start lettering.
        CALL UNITS(TTLX,TTLY,UNITX,UNITY)
        STREND='*.'
        TTLX1=TTLX // UNITX // STREND
```

```
      TTLY1=TTLY // UNITY // STREND
      TTLZ1=TTLZ // ' = ' // STREND
C
      CALL MOVTO2 (ATPOSX,ATPOSY)
      CALL CHASIZ (2.5,2.5)
      CALL CHAANG (90.)
      CALL CHASTR (TTLY1)
      CALL MOVTO2 (BTPOSX,BTPOSY)
      CALL CHAANG (0.)
      CALL CHASTR (TTLX1)
C ...............................................
C ... Explain the curves symbols.
      CALL AXIPOS (1,0.,0.,210.,1)
      CALL AXISCA (1,10,0.,210.,1)
      CALL AXIPOS (1,0.,0.,280.,2)
      CALL AXISCA (1,10,0.,280.,2)
C
      CALL GRASYM (50.,72.,1,7,0)
      CALL MOVTO2 (55.,70.)
      CALL CHASTR (TTLZ1)
      CALL MOVTO2 (90.,70.)
      CALL CHAFIX (Z(1),-7,0)
C
      IF(NPOINT(2).EQ.0)  GOTO  79
      CALL GRASYM (120.,72.,1,6,0)
      CALL MOVTO2 (125.,70.)
      CALL CHASTR (TTLZ1)
      CALL MOVTO2 (160.,70.)
      CALL CHAFIX (Z(2),-7,0)
C
      IF(NPOINT(3).EQ.0)  GOTO  79
      CALL GRASYM (50.,67.,1,5,0)
      CALL MOVTO2 (55.,65.)
      CALL CHASTR (TTLZ1)
      CALL MOVTO2 (90.,65.)
      CALL CHAFIX (Z(3),-7,0)
C
      IF(NPOINT(4).EQ.0)  GOTO  79
      CALL GRASYM (120.,67.,1,2,0)
      CALL MOVTO2 (125.,65.)
      CALL CHASTR (TTLZ1)
      CALL MOVTO2 (160.,65.)
      CALL CHAFIX (Z(4),-7,0)
C .......................................
C ... write class & type of the material.
      MTRLCLS1= MTRLCLS // STREND
      MTRLTYP1= MTRLTYP // STREND
C ... Write the main title.
79    CALL CHASIZ (2.5,2.5)
      CALL MOVTO2 (75.,20.)
      TTLM=TTLY//' Vs '//TTLX//' FOR '//
     *     MTRLTYP // STREND
      CALL CHASTR (TTLM)
1000  RETURN
      END
```

```
      SUBROUTINE GRAF3(Y,YMAX,YMIN,X,XMAX,XMIN,
     *                 A,B,NCASE,COD)
C ***************************************************
C ... Routine to plot one curve and the original
C ... points. Created on the 6th SEP. 1988.

C ... External parameters.

      INTEGER  NCASE,COD
      REAL*8   B(21),A(21),X(NCASE),XMAX,XMIN,
     *          Y(NCASE),YMAX,YMIN
C
C ... Local parameters.

      INTEGER   NPOINT,IVAL

      REAL*4    BPOSX, BPOSY, APOSX,APOSY,
     *          ATPOSX, ATPOSY
      REAL*4    AR(21),BR(21),XR(900),XRMAX,XRMIN,
     1           YR(900),YRMAX,YRMIN,AMAX,AMIN,
     2           BMAX,BMIN,DELTAA,DELTAB
C
      CHARACTER   TTLM*80,TTLY*20,
     1             BEL,STREND*2
C
      DATA BPOSX/50./      ! X coord of start of X-axis
      DATA APOSX/50./      ! X coord of start of Y-axis
      DATA ATPOSX/40./  ! X coord of start of Y-title
      DATA BEL,NPOINT /7, 21/

C ... Specify the Axis Y-coordinates...
      KCOD=COD-1
      APOSY=40+(KCOD*47.)
      ATPOSY=APOSY+7.0
      BPOSY=APOSY

C ... Compose the arrays for plotting...
      DO 20 I=1,NCASE
         XR(I)=X(I)
         YR(I)=Y(I)
20    CONTINUE
      XRMIN=XMIN
      YRMIN=YMIN
      XRMAX=XMAX
      YRMAX=YMAX

C ... Reduce No.of points to 21 ...
      DO 55 I=1,21
         II = I-1
         AR(I)=A(1+(II*5))
         BR(I)=B(1+(II*5))
55    CONTINUE
      AMAX=-1.E20
      AMIN=-AMAX
```

```
C ... Test for the min.&max. of A & B.
      DO 65, I=1,21
         IF (AR(I).GE.AMAX) AMAX=AR(I)
         IF (AR(I).LE.AMIN) AMIN=AR(I)
65    CONTINUE
C ... Test for the absolute max. & min. of the axes.
      IF (AMAX.GE.YRMAX)  AMAXT=AMAX
      IF (AMAX.LT.YRMAX)  AMAXT=YRMAX
      IF (AMIN.LE.YRMIN)  AMINT=AMIN
      IF (AMIN.GT.YRMIN)  AMINT=YRMIN

C ... Specify the plot boundaries.
      DELTAB=XRMAX-XRMIN
      BMAXT=XRMAX+(.05*DELTAB)
      BMINT=XRMIN-(.05*DELTAB)
      DELTAA=AMAXT-AMINT
      AMAXT=AMAXT+(.05*DELTAA)
      AMINT=AMINT-(.05*DELTAA)

C... Start of the graph.
      IF (COD.EQ.1)   THEN
         IVAL = 1
      ELSE
         IVAL = 0
      ENDIF
      CALL CHASIZ (2.,2.)
      CALL AXIPOS (1,BPOSX,BPOSY,140.,1)
      CALL AXISCA (1,20,BMINT,BMAXT,1)
      CALL AXIDRA (-2,IVAL,1)
      CALL AXIPOS (1,APOSX,APOSY,40.,2)
      CALL AXISCA (2,5,AMINT,AMAXT,2)
      CALL AXIDRA (2,-1,2)

C ... Plot the oringinal points...
      CALL GRASYM (XR,YR,NCASE,8,0)

C ... Plot the fitting curve...
      CALL GRAPOL (BR,AR,NPOINT)

C .......... Start lettering.
      IF (COD.EQ.1)  TTLY='KERF mic.'
      IF (COD.EQ.2)  TTLY='Ra nano.'
      IF (COD.EQ.3)  TTLY='DROSS mic.'
      IF (COD.EQ.4)  TTLY='TAPER mic.'
      IF (COD.EQ.5)  TTLY='H.A.Z mic.'

C ... Terminate the string ...
      TTLY=TTLY // '*.'
C ... Write the Y-axis title...
      CALL MOVTO2 (ATPOSX,ATPOSY)
      CALL CHASIZ (2.5,2.5)
      CALL CHAANG (90.)
      CALL CHASTR (TTLY)
      CALL CHAANG (0.0)
1000  RETURN
      END
```

```
C ... AUXILIARY ROUTINES
C *****************************************************
      SUBROUTINE MINMX(NCASE,IA,A,AMAX,AMIN,IFLG)
C *****************************************************
C .. Subroutine for converting into real numbers
C ...and finding the max. & min of a parameter.
C
      INTEGER    NCASE,IA(NCASE),IFLG
      REAL*8     A(NCASE),AMAX,AMIN
C ...
      AMIN = 1.E15                ! Initiating max. & min.
      AMAX = -AMIN
      DO 10 I=1,NCASE
         A(I) = FLOAT(IA(I))
         IF (AMAX.LE.A(I)) AMAX=A(I)
         IF (AMIN.GE.A(I)) AMIN=A(I)
10    CONTINUE
      IF (AMAX.EQ.AMIN)  IFLG=1
      RETURN
      END
C *********************************************************
      SUBROUTINE UNITS(TTLX,TTLY,UNITX,UNITY)
C *********************************************************
C ... Subroutine to supply units for the
C ... perspectvie parameters ...
C ...
      CHARACTER*(*)    TTLX,TTLY,UNITX,UNITY
C ...
C ... Specifying units for the X-axis ...
      IF (TTLX.EQ.'THICKNESS') UNITX=' micron'
      IF (TTLX.EQ.'PRF')     UNITX=' Hz'
      IF (TTLX.EQ.'GAS PRESSURE') UNITX=' mbar'
      IF (TTLX.EQ.'FEED')    UNITX=' mm/min'
      IF (TTLX.EQ.'POWER')   UNITX=' Watt'
      IF (TTLX.EQ.'PULSE ENERGY') UNITX=' mJoule'
      IF (TTLX.EQ.'STAND OFF')  UNITX=' micron'
      IF ((TTLX.EQ.'KERF WIDTH').OR.(TTLX.EQ.'H.A.ZONE')
     1   .OR.(TTLX.EQ.'TAPER').OR.(TTLX.EQ.'DROSS SIZE'))
     2         UNITX=' micron'
      IF (TTLX.EQ.'CUT ROUGHNESS')  UNITX=' nano Ra'
      IF (TTLX.EQ.'FEED')    UNITX=' mm/min'
      IF (TTLX.EQ.'POWER')  UNITX=' Watt'
      IF (TTLX.EQ.'PULSE ENERGY') UNITX=' mJoule'
C ...
C ... Specifying units for the Y-axis ...
      IF ((TTLY.EQ.'KERF WIDTH').OR.(TTLY.EQ.'H.A.ZONE')
     1   .OR.(TTLY.EQ.'TAPER').OR.(TTLY.EQ.'DROSS SIZE'))
     2         UNITY=' micron'
      IF (TTLY.EQ.'CUT ROUGHNESS')  UNITY=' nano Ra'
      IF (TTLY.EQ.'FEED')    UNITY=' mm/min'
      IF (TTLY.EQ.'POWER')  UNITY=' Watt'
      IF (TTLY.EQ.'PULSE ENERGY') UNITY=' mJoule'
      IF (TTLY.EQ.'THICKNESS') UNITY=' micron'
      IF (TTLY.EQ.'PRF')     UNITY=' Hz'
      IF (TTLY.EQ.'GAS PRESSURE') UNITY=' mbar'
      IF (TTLY.EQ.'FEED')    UNITY=' mm/min'
```

```
      IF (TTLY.EQ.'POWER')  UNITY=' Watt'
      IF (TTLY.EQ.'PULSE ENERGY') UNITY=' mJoule'
      IF (TTLY.EQ.'STAND OFF')  UNITY=' micron'
C ...
      RETURN
      END
C ***********************************************
      SUBROUTINE DPNOIS (VAL,K)
C ***********************************************
C ... Subroutine to dump the noise that can
C ... occure at the 3D fitting routine,by
C ... dumping all non-zero values ,that are
C ... followed by zero values, to zero.
C ...
      INTEGER     K
      REAL *8      VAL(K)
C ...
      DO 10 I=1,K-1
         VAL(I) = 0.0
10    CONTINUE
      RETURN
      END
C ****************************************************
      SUBROUTINE GRFSTT (TTLX)
C ****************************************************
C ... Routine to start plotting a multi-graf.
C ... points. Created on the 6th SEP. 1988.
C ... External parameters.
C ...
      CHARACTER*(*) TTLX
C ... Local parameters.
      CHARACTER   TTLM*80,UNITX*10,D1*1,D2*1
C ...
      CALL PICCLE       !Start a new page.
C ... complete the graph box .
      CALL MOVTO2 (50.,275.)
      CALL LINTO2 (190.,275.)
      CALL LINTO2 (190.,40.)
C ... Write the X-axis title....
      CALL UNITS(TTLX,D1,UNITX,D2)
      CALL CHAANG (0.)
      CALL CHASIZ (2.5,2.5)
      CALL MOVTO2 (100.,30.)
      CALL CHASTR (TTLX//UNITX//'*.')
C ... Write the main title.
      CALL CHASIZ (2.5,2.5)
      CALL MOVTO2 (75.,20.)
      TTLM='QUALITY FEATURES Vs '//TTLX//'*.'
      CALL CHASTR (TTLM)
      RETURN
      END
```

APPENDIX 6C

THE PREDICTION PROGRAM EVALUATION RESULTS

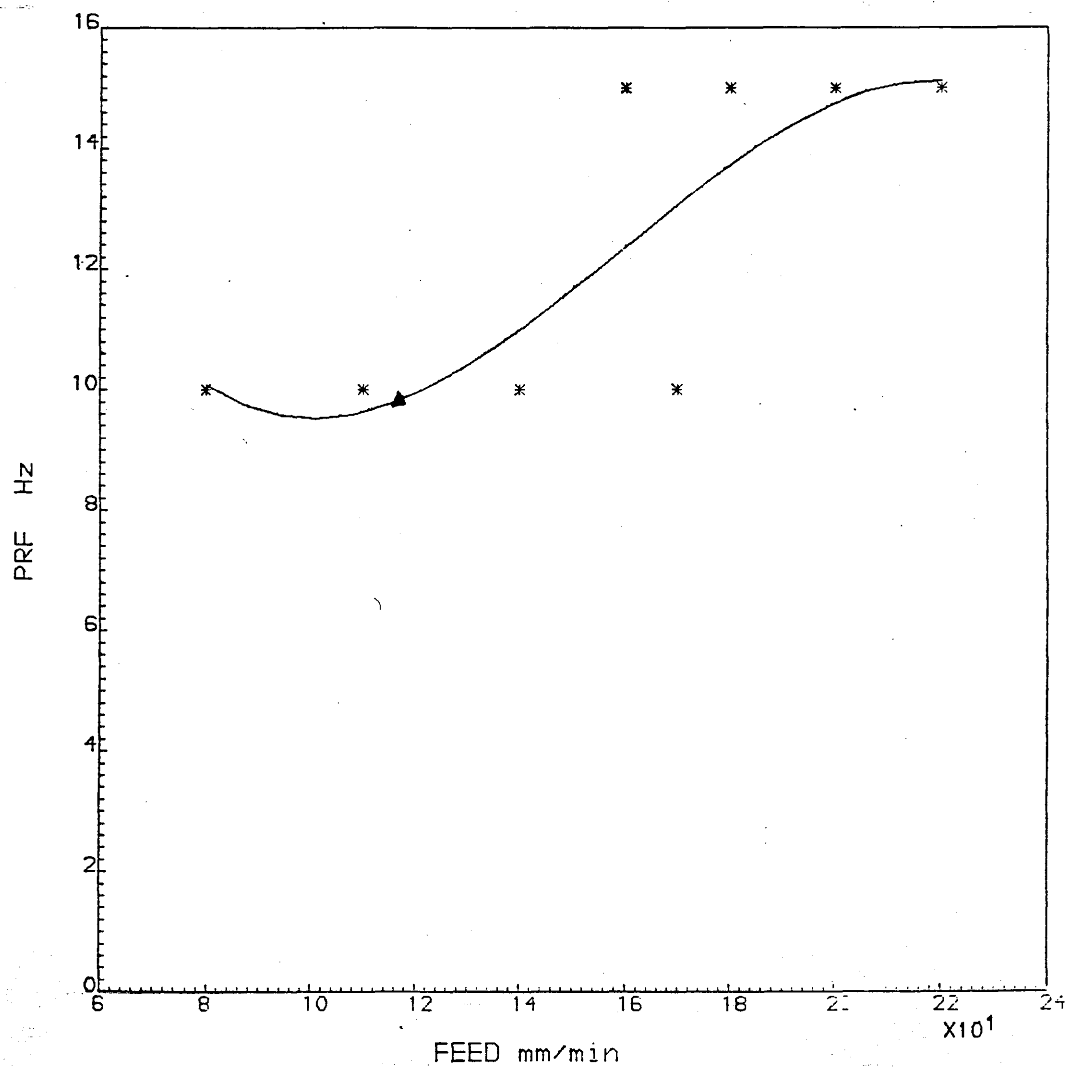
This Appendix consists of four groups of figures which are output from four runs of the prediction program. The experimental results are on these figures by triangular symbols. The figures are grouped as follows:

a) Fig´s 6C1 to 6C5 present the prediction for cutting with the YAG laser, when data for single thickness is found.

b) Fig´s 6C6 to 6C11 present the prediction for cutting with the YAG laser, when data for a range of thicknesses is encountered.

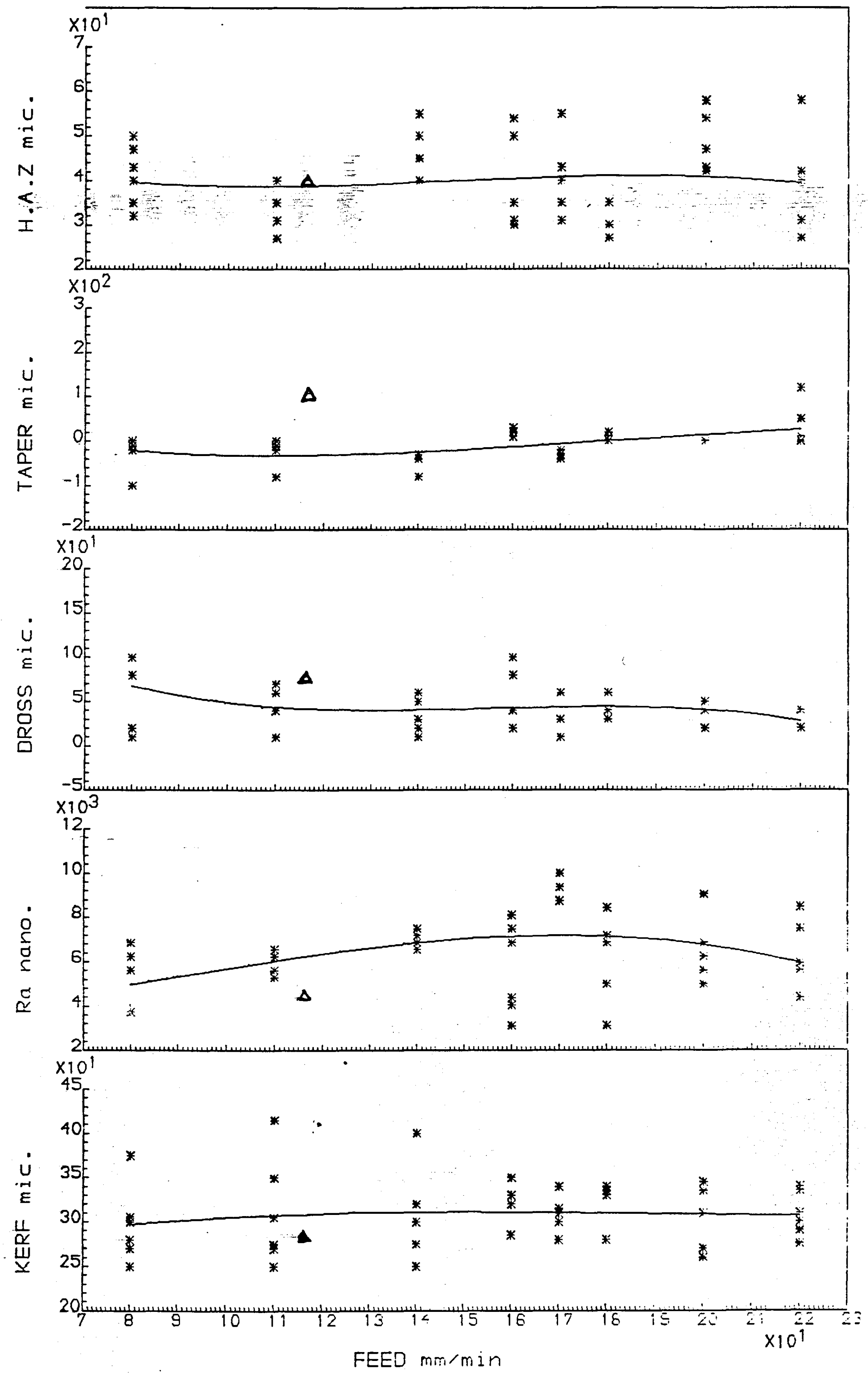
c) Fig´s 6C12 to 6C16 present the prediction for cutting with the CO_2 laser, when data for single thickness found

d) Fig´s 6C17 to 6C23 present the prediction for cutting with the CO_2 laser, when data for a range of thicknesses is encountered.



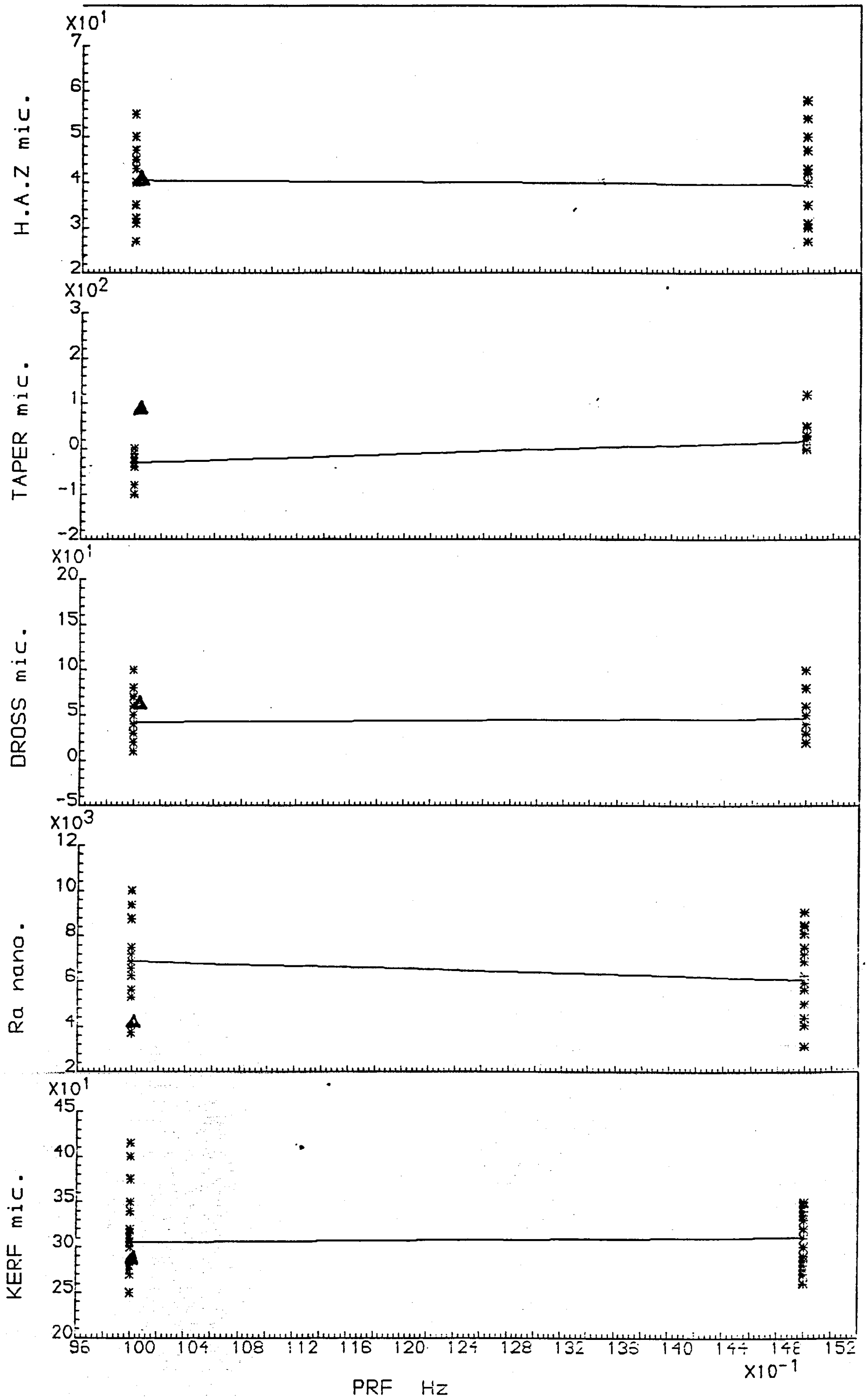
PRF Vs FEED FOR TOOL-STEEL

FIG 6C1



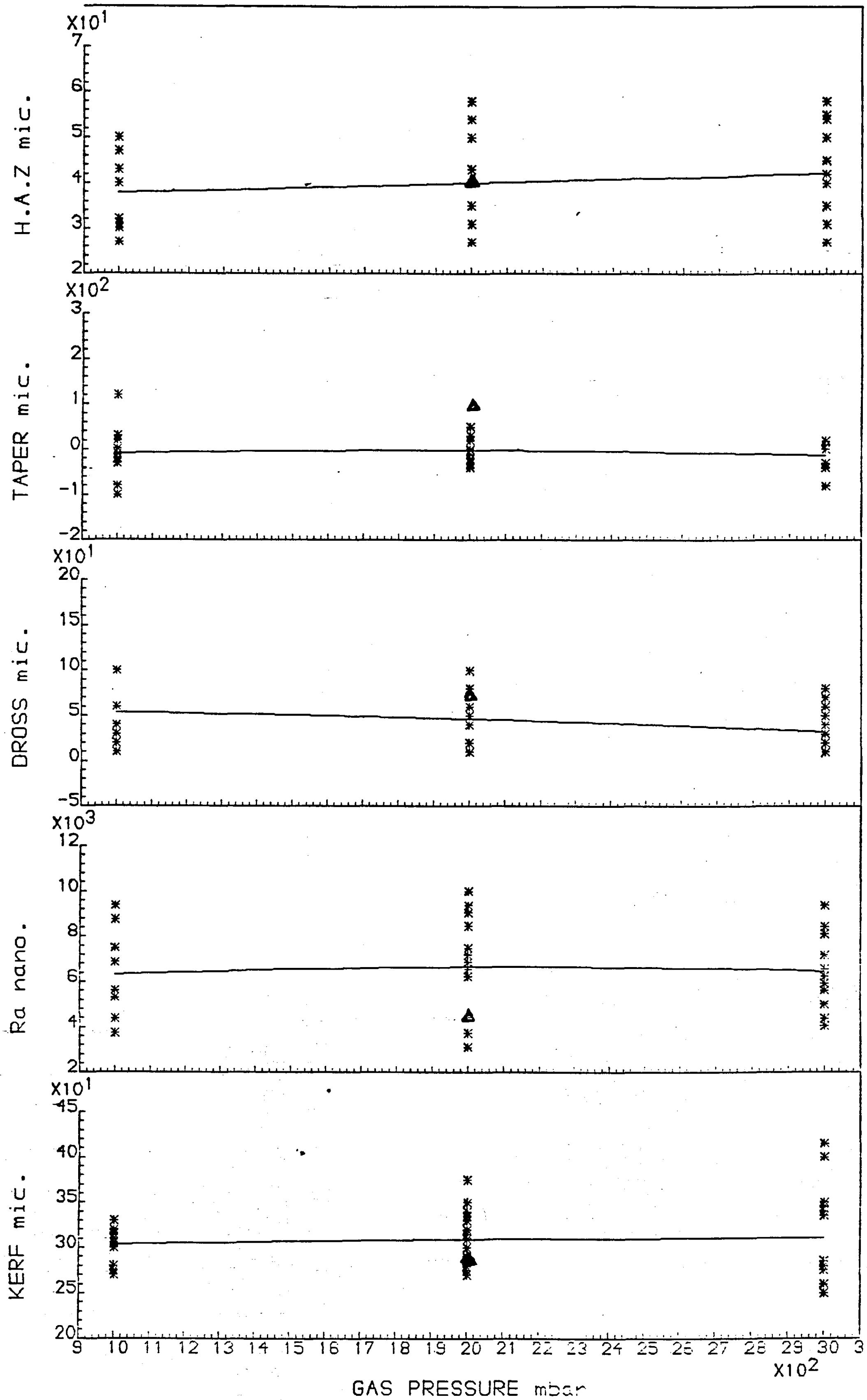
QUALITY FEATURES Vs FEED

FIG 6C2



QUALITY FEATURES Vs PRF

FIG 6C3



QUALITY FEATURES Vs GAS PRESSURE

FIG 6C4

4-OCT-88 16:23:03

SUMMARY OF RESULTS

MATERIAL CLASS IS ALLOY-STEEL
MATERIAL TYPE IS TOOL-STEEL
MATERIAL THICKNESS IS 1600 micron

ALL THE DATA THAT WAS COLLECTED FROM THE DATABASE ARE OF THE SAME THICKNESS AS THE TARGET MATERIAL WHICH IS 1600. microns.

THE MEAN PULSE ENERGY IS 4091.mJ.

FEEDS THAT CAN ACTUATE MIN. KERF,ROUGHNESS,DROSS, TAPER & HAZ ARE ,RESPECTIVELY, AS FOLLOWS :

80.	80.	220.	116.	114.

WITH PREFERENCE WEIGHTS OF :

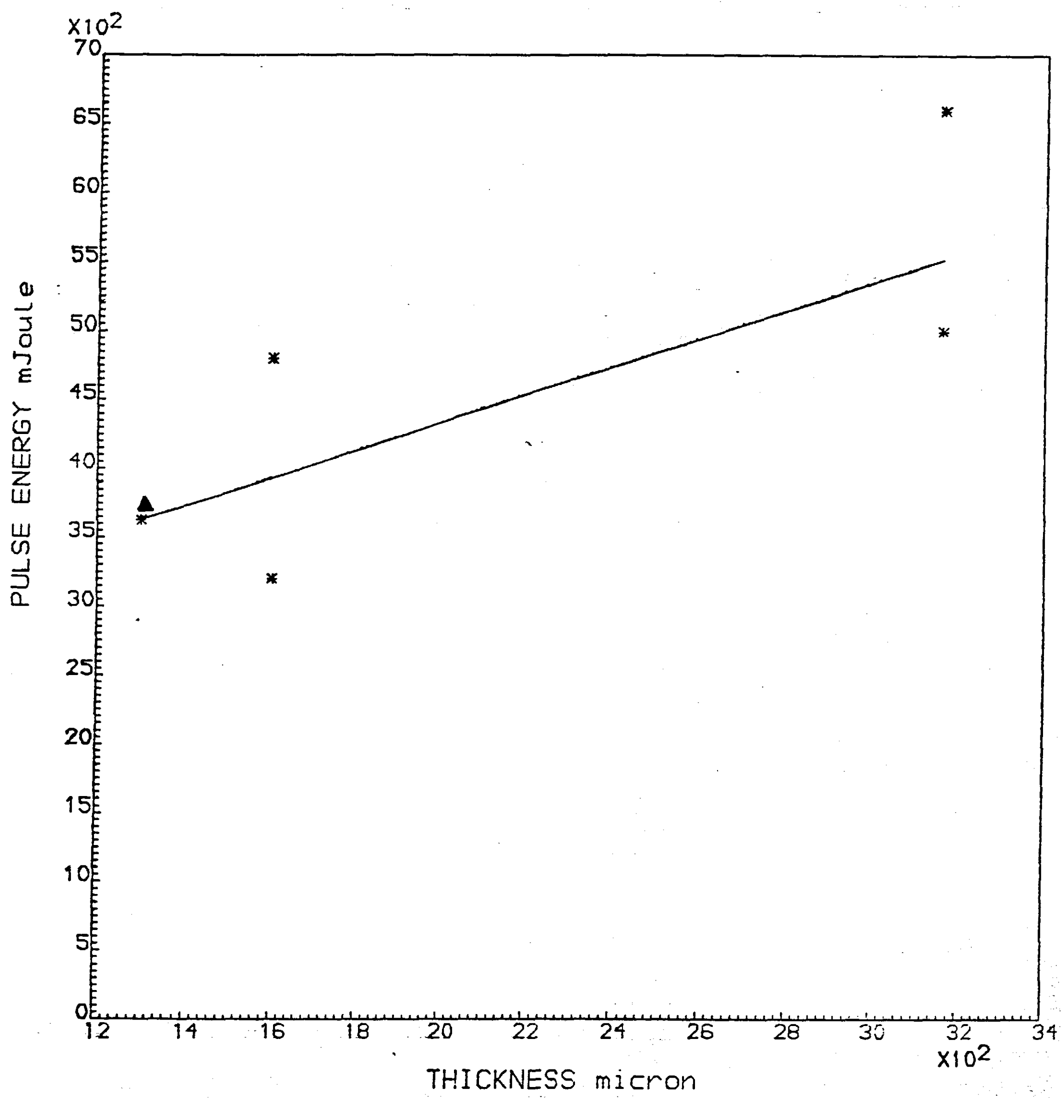
0.30	0.40	0.00	0.00	0.00

WHILE THE PREFERENCE WEIGHT FOR MAXIMUM FEED IS 0.30

THE RECOMMENDED FEED IS 116.mm/min.

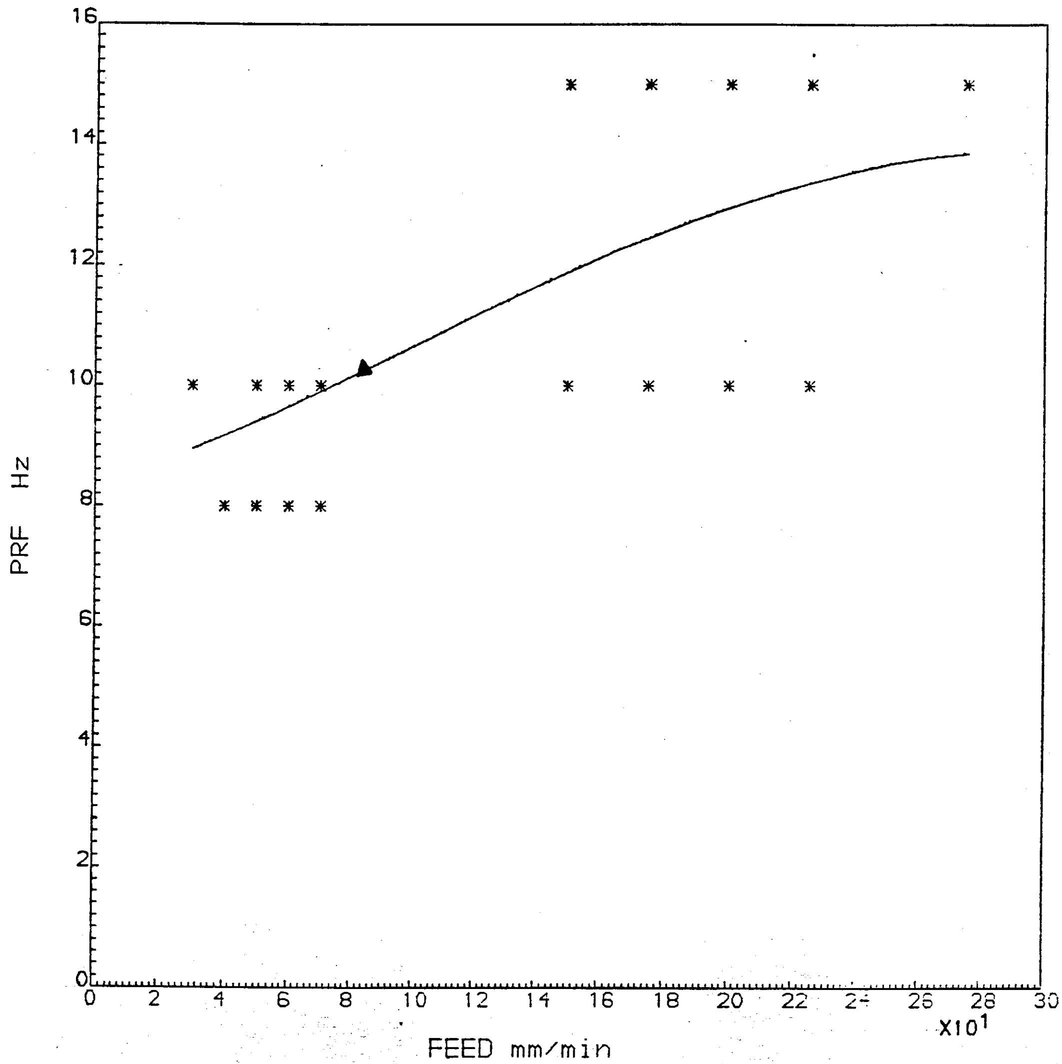
THE RECOMMENDED PRF IS 10.Hz.

FIG 6C5



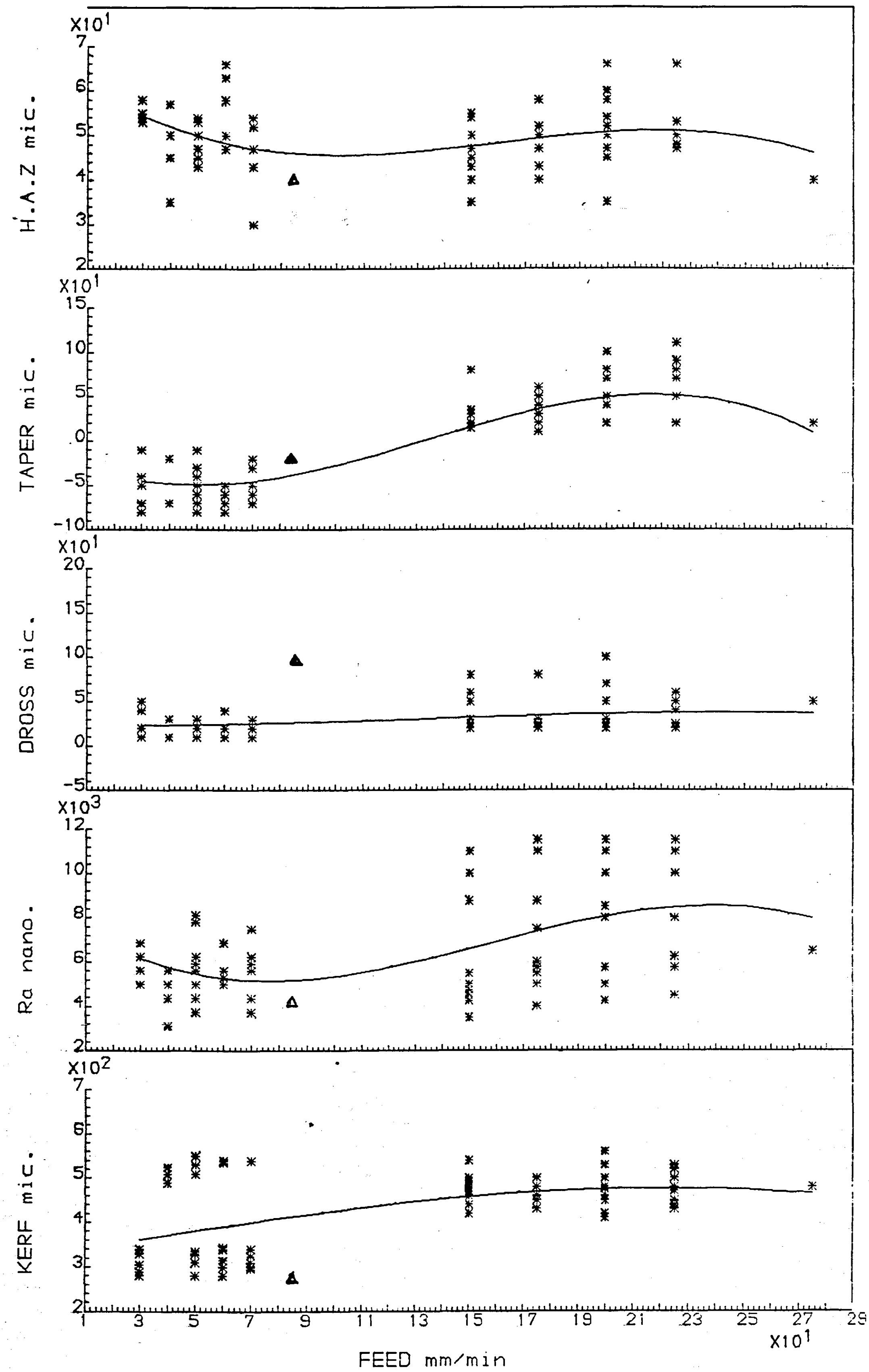
PULSE ENERGY Vs THICKNESS FOR MILD-STEEL

FIG 6C6



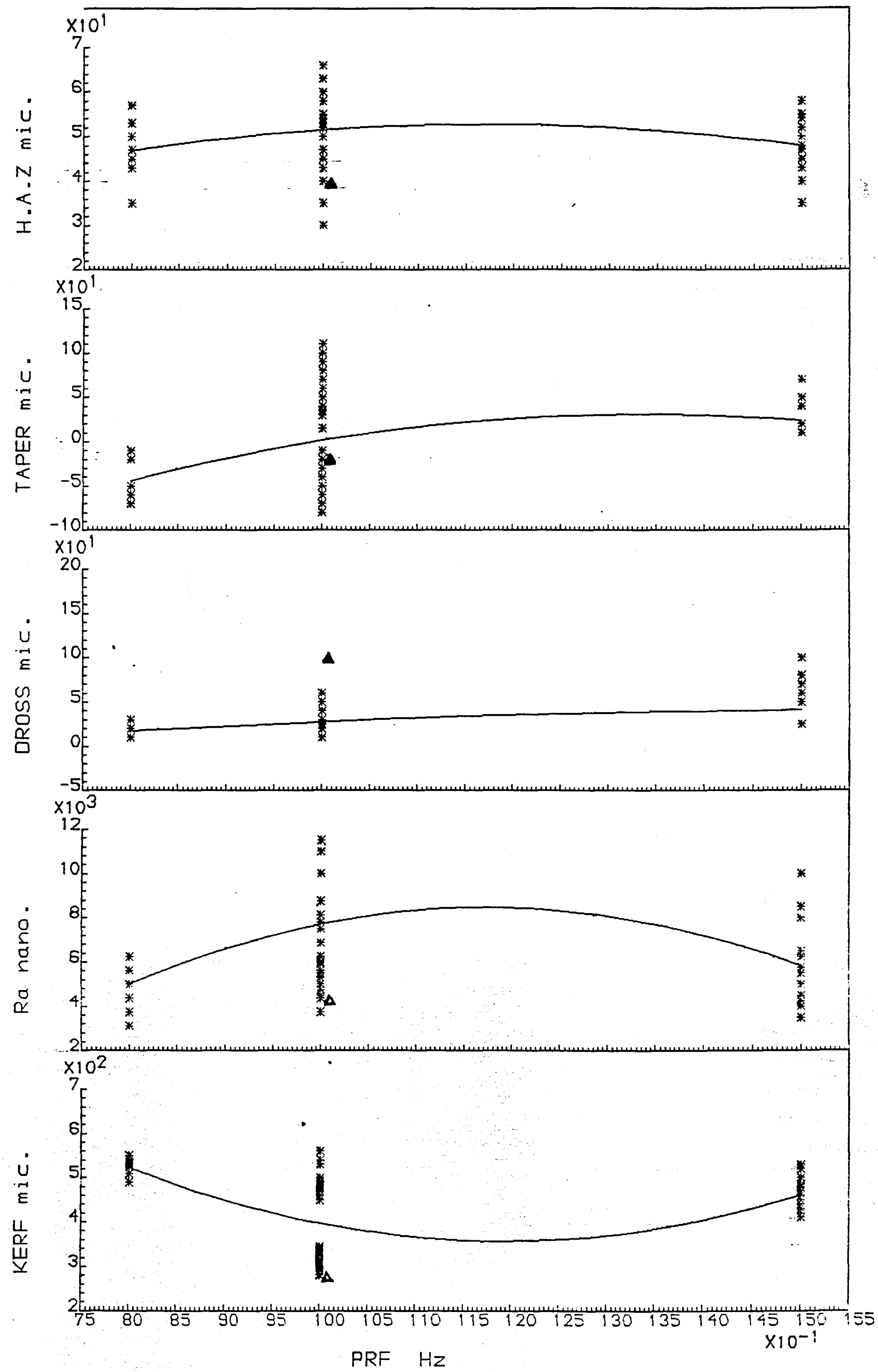
PRF Vs FEED FOR MILD-STEEL

FIG 6C7



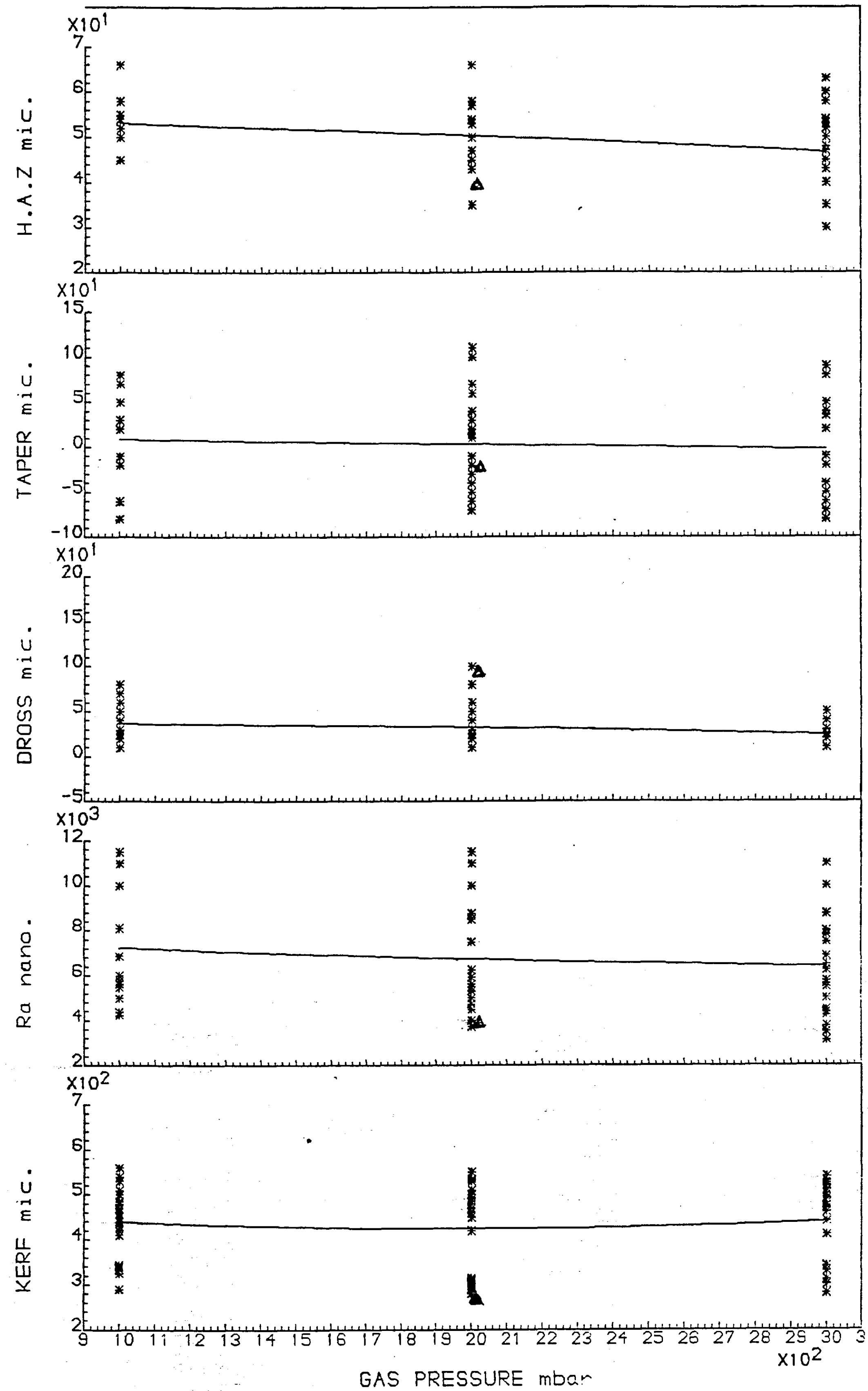
QUALITY FEATURES Vs FEED

FIG 6C8



QUALITY FEATURES Vs PRF

FIG 6C9



QUALITY FEATURES Vs GAS PRESSURE

FIG 6C10

5-OCT-88 12:57:30

SUMMARY OF RESULTS

MATERIAL CLASS IS MILD-STEEL
MATERIAL TYPE IS MILD-STEEL
MATERIAL THICKNESS IS 1300 micron

AS THE TARGET MATERIAL THICKNESS HAS NOT BEEN FOUND IN THE DATABASE, INTERPOLATION/EXTRAPOLATION WILL BE USED TO PREDICT APPROXIMATE SETTINGS.

THE RECOMMENDED PULSE ENERGY IS 3625.mJ.

FEEDS THAT CAN ACTUATE MIN. KERF,ROUGHNESS,DROSS, TAPER & HAZ ARE ,RESPECTIVELY, AS FOLLOWS :

30.	89.	52.	62.	113.

WITH PREFERENCE WEIGHTS OF :

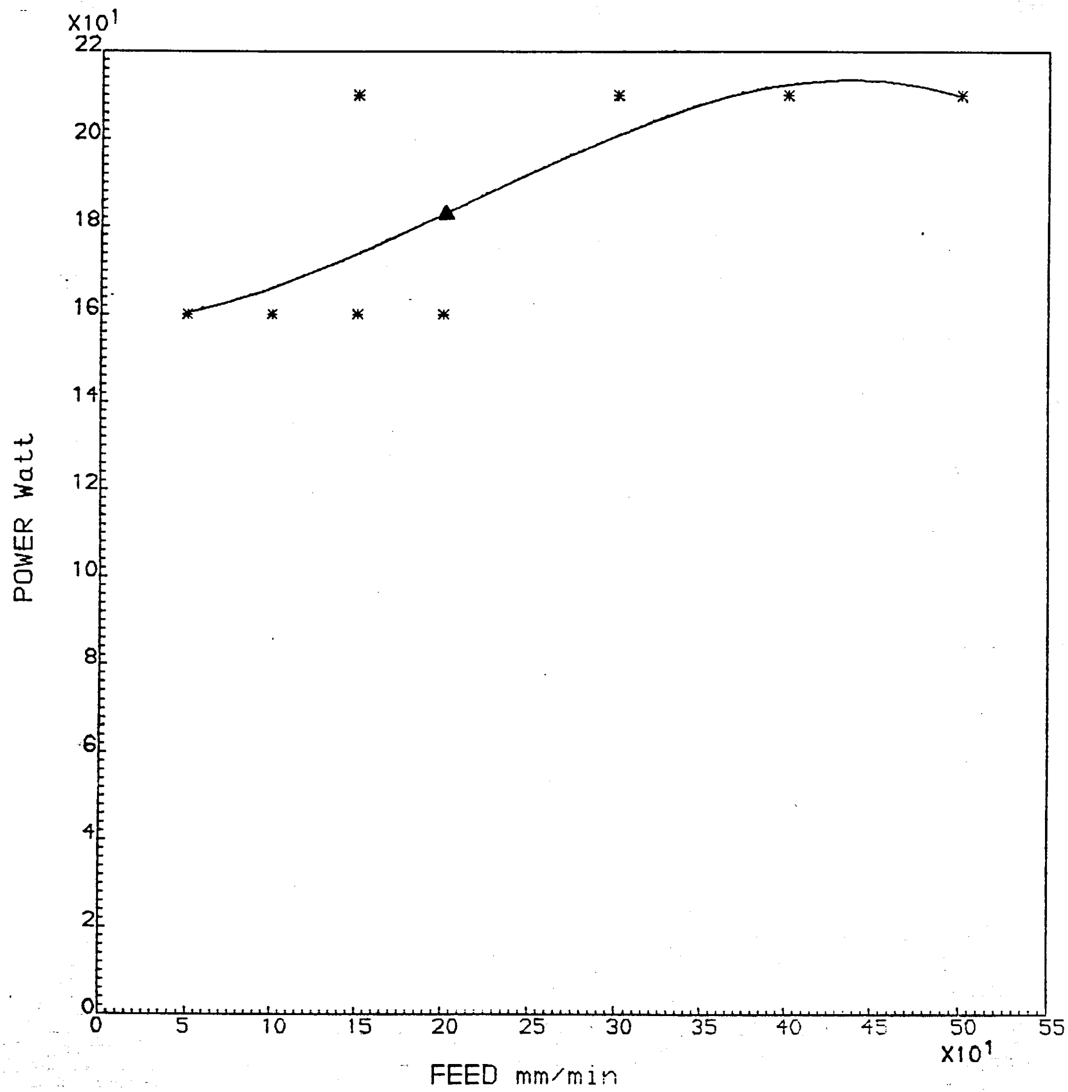
0.80	0.90	0.70	0.30	0.00

WHILE THE PREFERENCE WEIGHT FOR MAXIMUM FEED IS 0.50

THE RECOMMENDED FEED IS 84.mm/min.

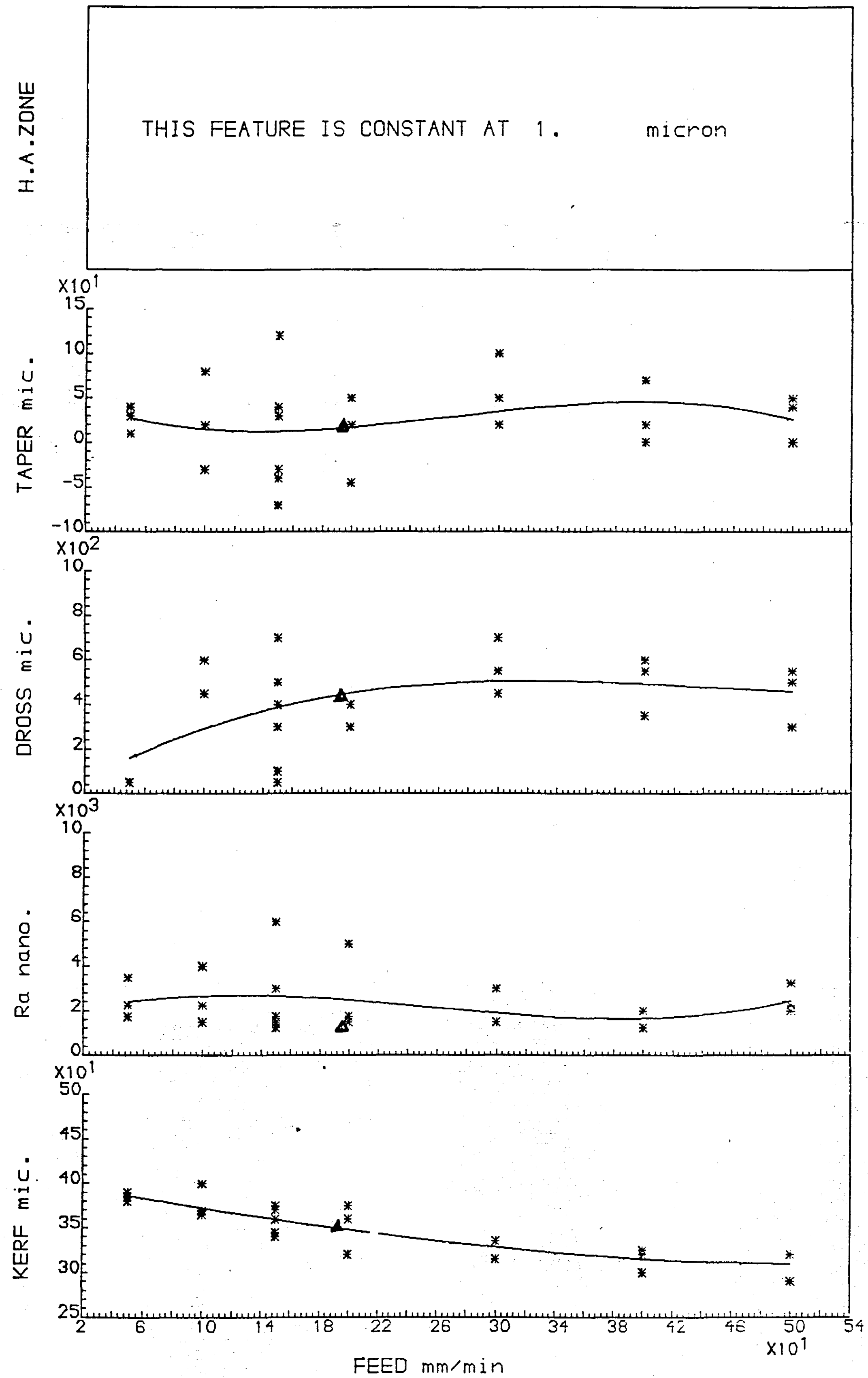
THE RECOMMENDED PRF IS 10.Hz.

FIG 6C11



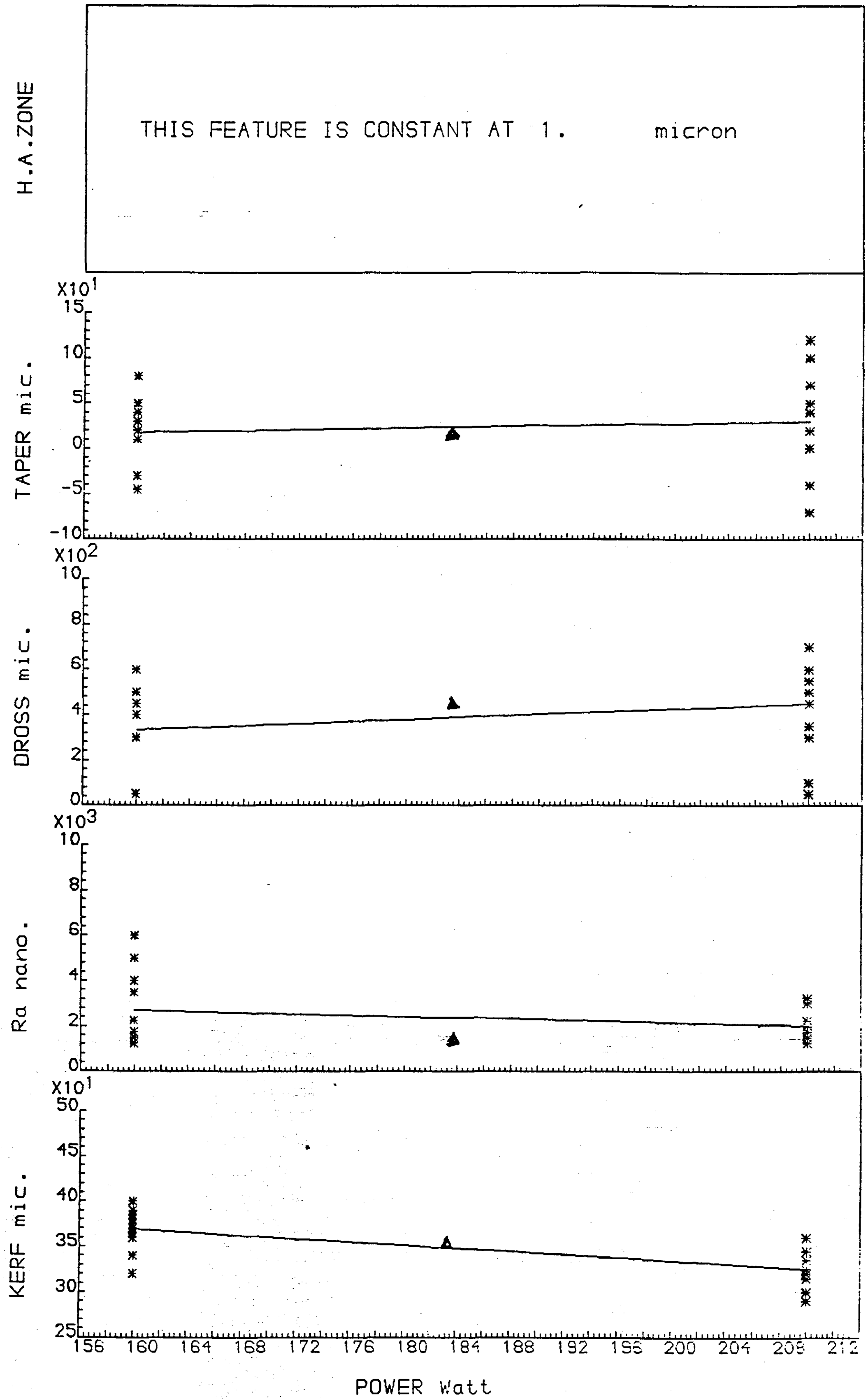
POWER Vs FEED FOR 1.6mm STAINLESS-STEEL

FIG 6C12



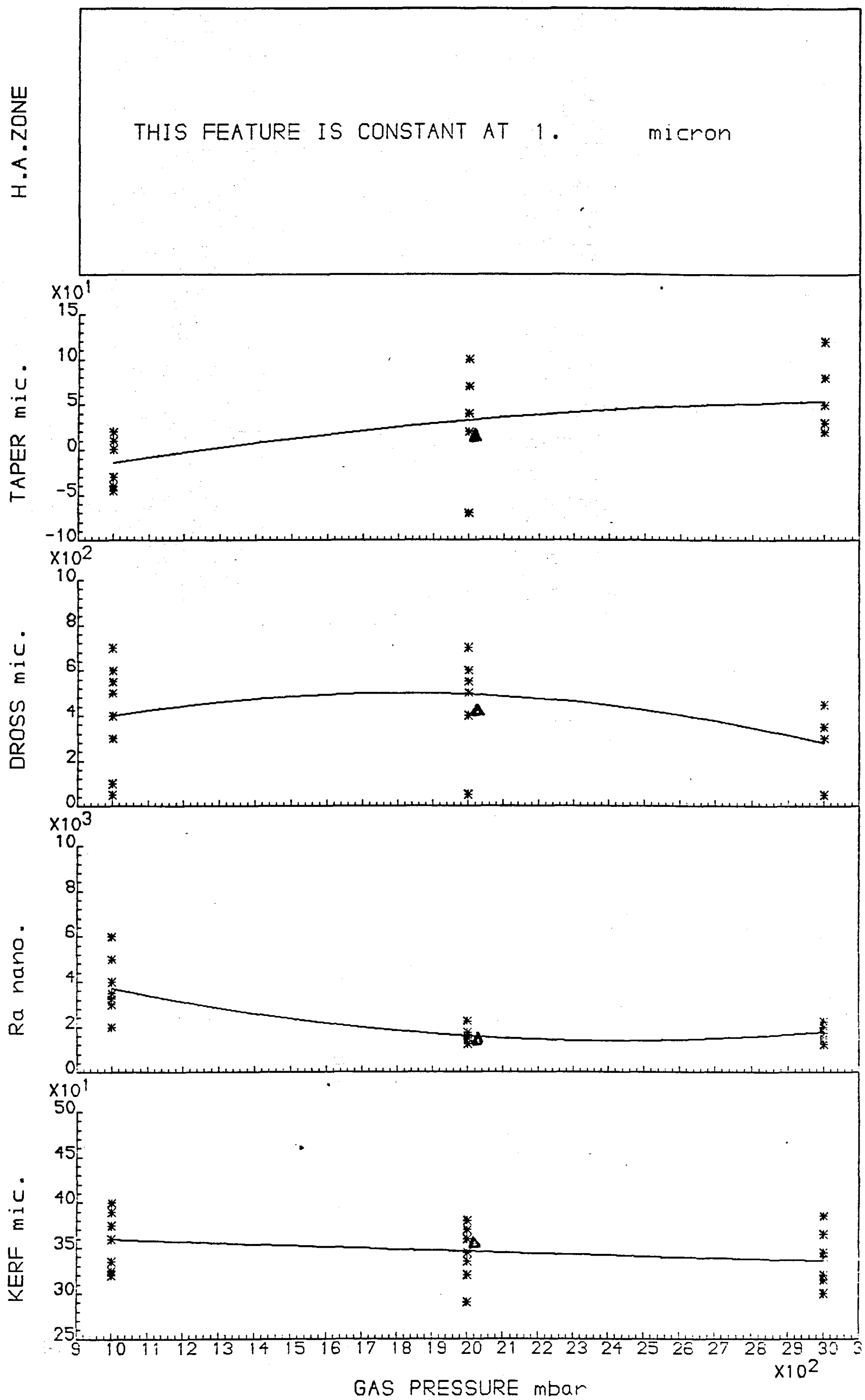
QUALITY FEATURES Vs FEED

FIG 6C13



QUALITY FEATURES Vs POWER

FIG 6C14



QUALITY FEATURES Vs GAS PRESSURE

FIG 6C15

5-OCT-88 09:08:28

SUMMARY OF RESULTS

MATERIAL CLASS IS STAINLESS-STEEL
MATERIAL TYPE IS 1.6mm STAINLESS-STEEL
MATERIAL THICKNESS IS 1600 micron

ALL THE DATA THAT WAS COLLECTED FROM THE DATABASE ARE OF THE SAME THICKNESS AS THE TARGET MATERIAL WHICH IS 1600. microns.

FEEDS THAT CAN ACTUATE MIN. KERF,ROUGHNESS,DROSS, TAPER & HAZ ARE ,RESPECTIVELY, AS FOLLOWS :

500.	405.	50.	163.	500.

WITH PREFERENCE WEIGHTS OF :

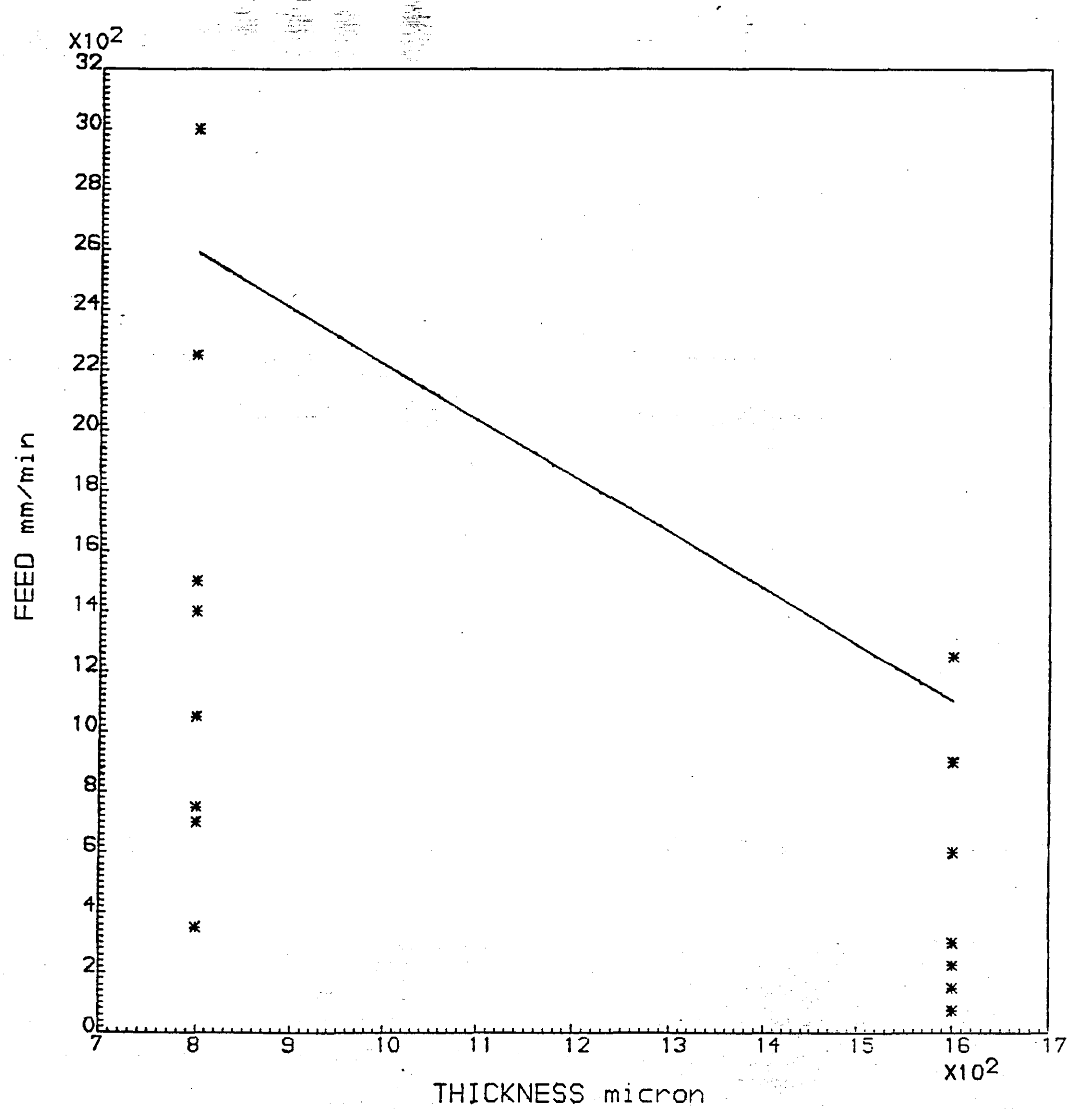
0.20	0.40	0.90	0.30	0.00

WHILE THE PREFERENCE WEIGHT FOR MAXIMUM FEED IS 0.50

THE RECOMMENDED FEED IS 194.mm/min.

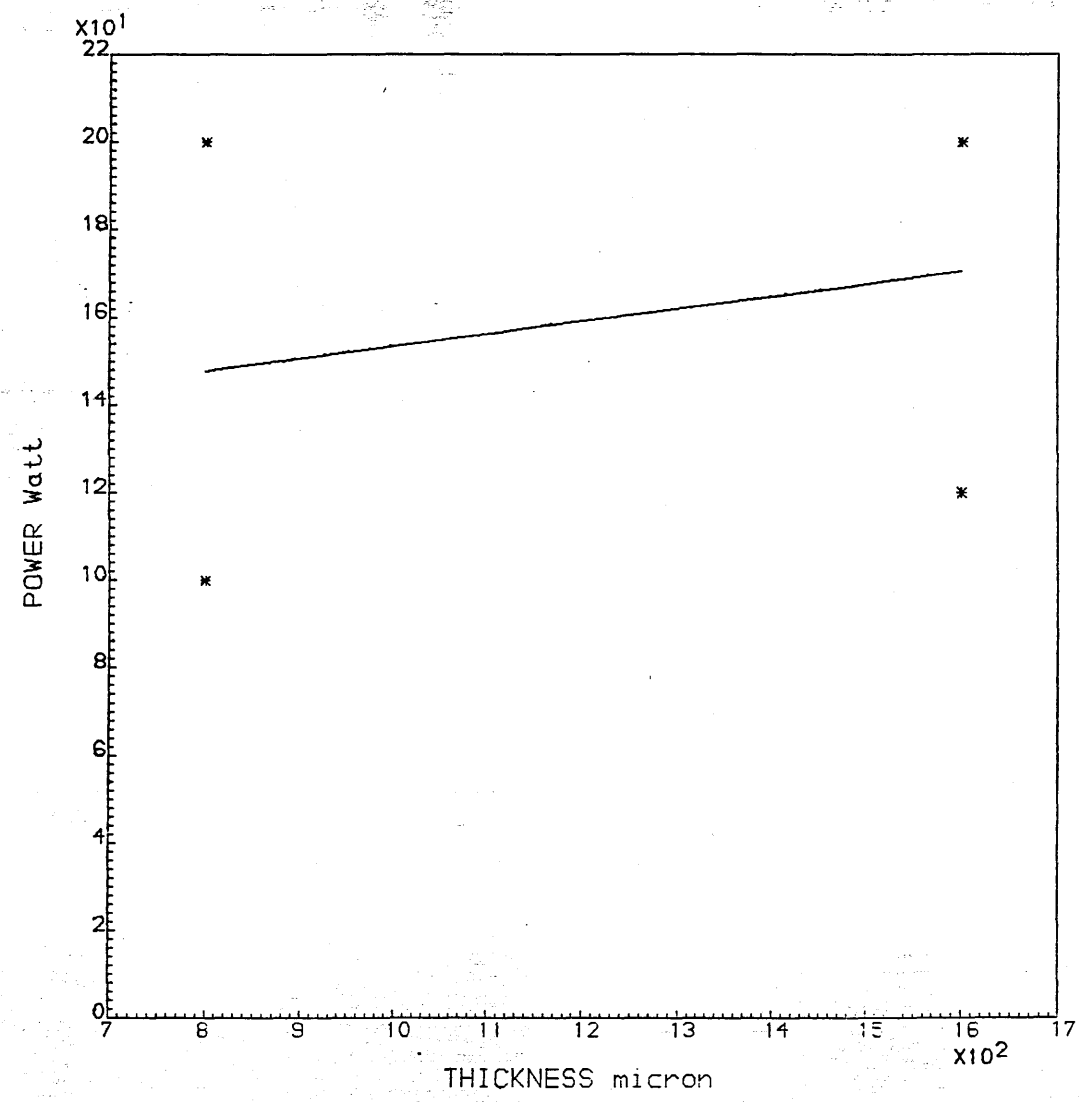
THE RECOMMENDED POWER IS 183.Watts.

FIG 6C16



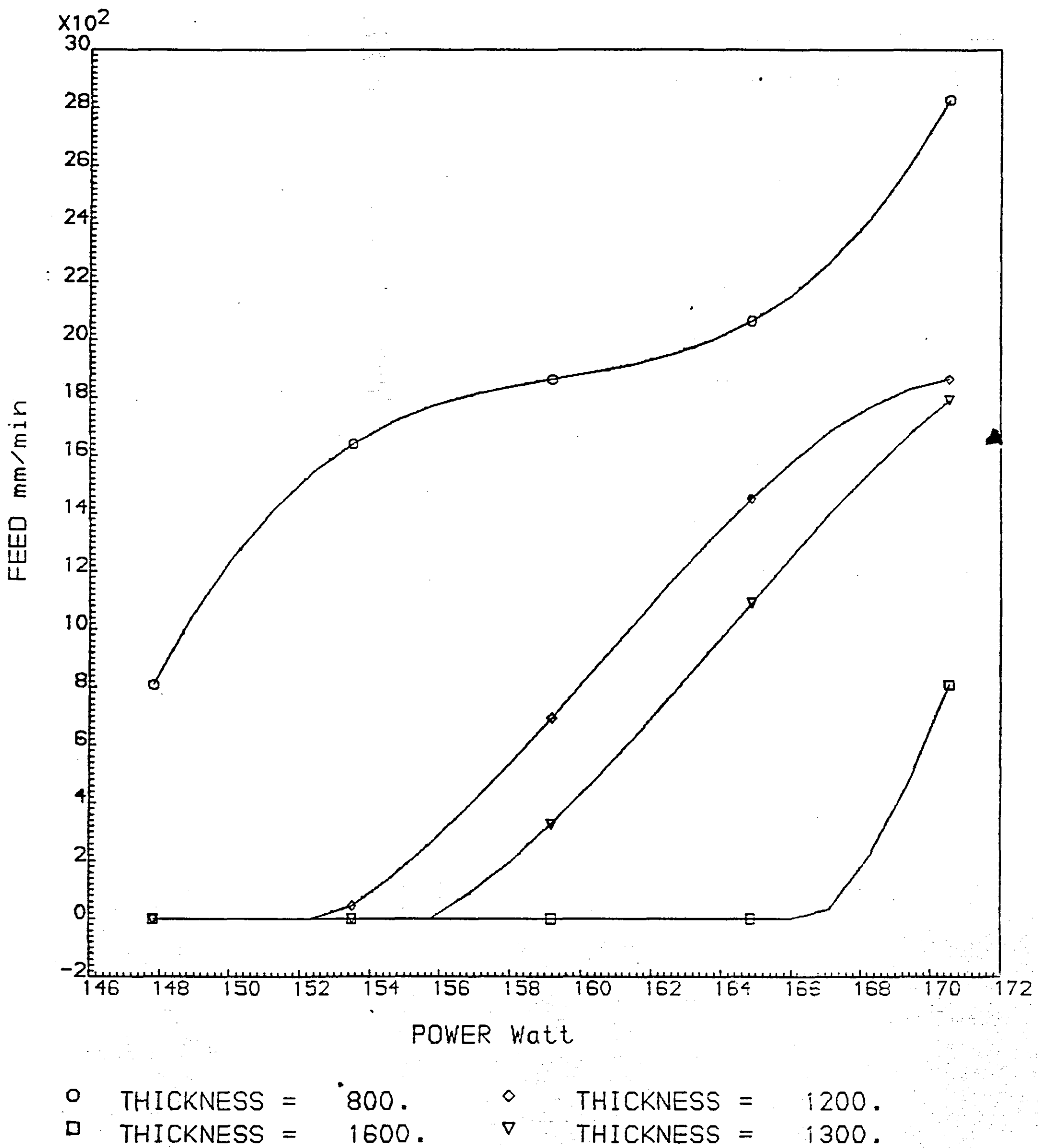
FEED Vs THICKNESS FOR MILD-STEEL

FIG 6C17



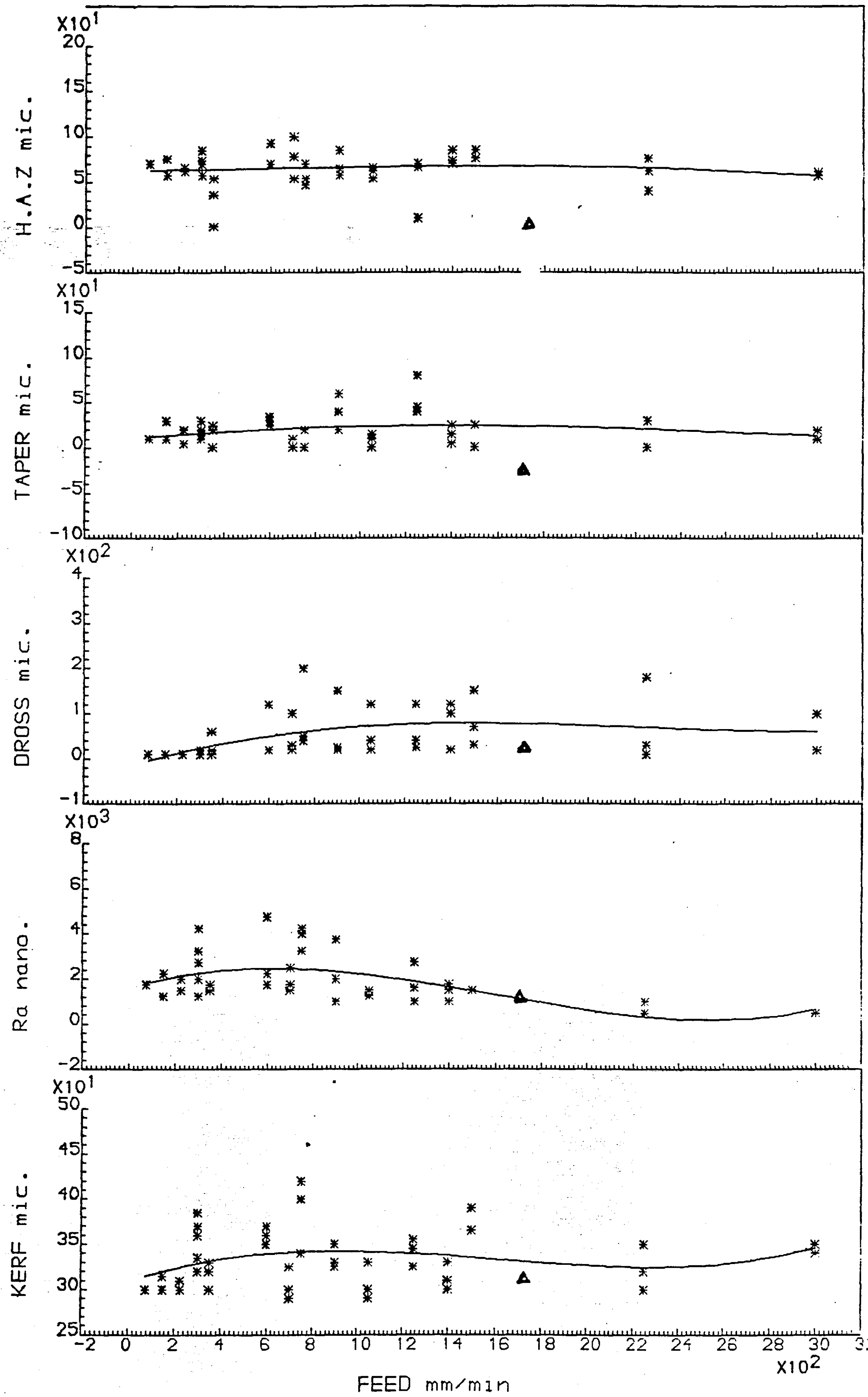
POWER Vs THICKNESS FOR MILD-STEEL

FIG 6C18



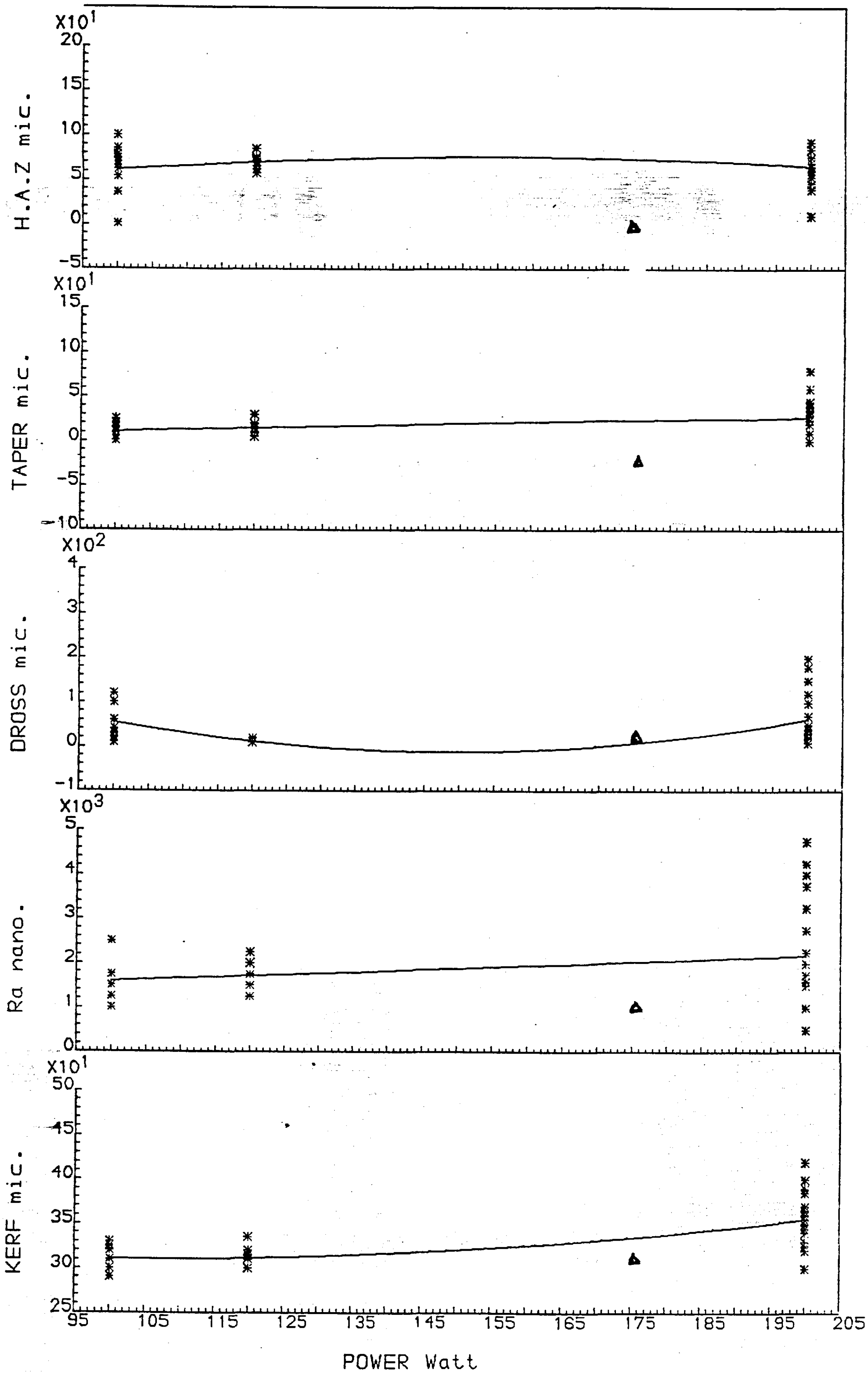
FEED Vs POWER FOR MILD-STEEL

FIG 6C19



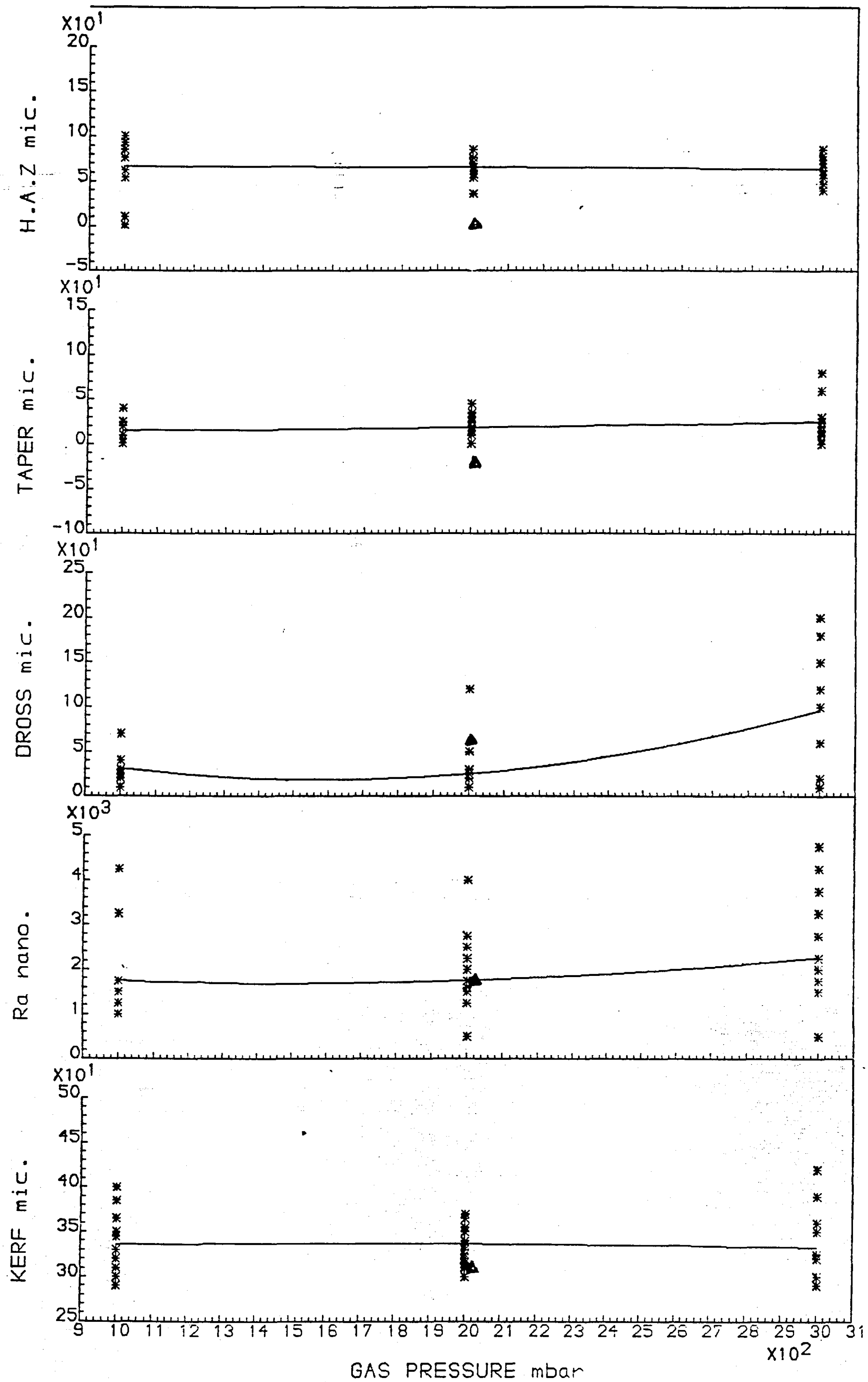
QUALITY FEATURES Vs FEED

FIG 6C20



QUALITY FEATURES Vs POWER

FIG 6C21



QUALITY FEATURES Vs GAS PRESSURE

FIG 6C22

5-OCT-88 12:45:00

SUMMARY OF RESULTS

MATERIAL CLASS IS MILD-STEEL
MATERIAL TYPE IS MILD-STEEL
MATERIAL THICKNESS IS 1300 micron

AS THE TARGET MATERIAL THICKNESS HAS NOT BEEN FOUND IN THE DATABASE, INTERPOLATION/EXTRAPOLATION WILL BE USED TO PREDICT APPROXIMATE SETTINGS.

FEEDS THAT CAN ACTUATE MIN. KERF,ROUGHNESS,DROSS, TAPER & HAZ ARE ,RESPECTIVELY, AS FOLLOWS :

1792. 1792. 75. 1792. 1792.

WITH PREFERENCE WEIGHTS OF :

0.20 0.80 0.00 0.00 0.00

WHILE THE PREFERENCE WEIGHT FOR MAXIMUM FEED IS 0.30

THE RECOMMENDED FEED IS 1756.mm/min.

THE RECOMMENDED POWER IS 170.Watts.

FIG 6C23