# COMPUTER AIDED MANUFACTURE
# IN A DNC CELL

by

SEYED EBRAHIM RAZAVI

Thesis Submitted To

THE UNIVERSITY OF ASTON IN BIRMINGHAM

For the Degree of

DOCTOR OF PHILOSOPHY

Department of Production Technology & Production Management

July, 1981

## ACKNOWLEDGEMENTS

The author would like to thank:-

Dr. D. A. Milner for his guidance and supervision of the project.

Mr. J. Hopton, to whom I am indebted for his valuable assistance throughout the course of this project.

Dr. J. D. Brindley for his technical advice.

Thanks are also due to Professor R. H. Thornley for permitting this project to be carried out in the Department of Production Technology & Production Management.

The work described in this thesis has been carried out independently and submitted for no other degree.

S E Razavi

S. E. RAZAVI

THE UNIVERSITY OF ASTON IN BIRMINGHAM

# COMPUTER AIDED MANUFACTURE IN A DNC CELL

by

SEYED EBRAHIM RAZAVI

A thesis submitted for the degree of
Doctor of Philosophy

July, 1981.

## SUMMARY

The work reported in this thesis is concerned firstly with the evaluation of computer control in a Direct Numerically Controlled manufacturing cell. For this purpose, an NC machining centre was connected in a Behind the Tape Reader Configuration to a mini computer. This involved the construction of hardware interface between the computer and the machine tool, and the development of the necessary control software.

Part programs, FORTRAN based, have been evolved to test the system under practical applications and a range of two and three dimensional components have been machined.

The second phase investigated in this project was the implementation of a Computer Aided Design and Machining software package called BCSURF for production of sculptured or free-form surfaces. A vector-valued parametric method was adopted for shape description and a lofting technique has been used for construction of the surface.

Using BCSURF, design starts by defining the cross-sections which are represented by uniform B-spline curves as approximations to given polygons. Here the order of the curve, the position and the number of polygon vertices can be used as parameters for the modification to achieve the desired curves. When the definition of the sectional curves is complete, the surface is interpolated over them by cubic cardinal splines.

Thus in order to integrate design and machining, using the mathematical representation of the surface, the package provides facilities that generate tool offset positions for ball-nosed cutters and finally produces interference checked part-programs so that the objects can be cut in the DNC cell.

In addition to the details of developing the DNC cell and the BCSURF, the thesis includes a review of developments in Numerical Control of machine tools and a critical survey of the major existing methods in Computer Aided Geometric Design.

KEYWORDS: Computer Aided Manufacture
Computer Aided Design
B-splines
Behind the Tape Reader
Direct Numerical Control

CONTENTS PAGE

CHAPTER ONE

INTRODUCTION

CHAPTER TWO

A REVIEW OF DEVELOPMENTS IN NUMERICAL CONTROL

PAGE

CHAPTER THREE

A REVIEW OF COMPUTER AIDED GEOMETRIC DESIGN OF FREE FORM SURFACES

CHAPTER FOUR

MINI COMPUTER - NC MACHINE INTERFACE

PAGE

CHAPTER FIVE

DEVELOPMENT OF THE COMPUTER AIDED SURFACE DESIGN PACKAGE "BCSURF"

CHAPTER SIX

RESULTS

PAGE

PAGE

CHAPTER SEVEN

CONCLUSIONS AND REMARKS

CHAPTER EIGHT

LIST OF ABBREVIATIONS

| | Abbreviation |
|---|---|
| Computer Aided Design | CAD |
| Computer Aided Manufacture | CAM |
| Numerical Control | NC |
| Computer Numerical Control | CNC |
| Direct Numerical Control | DNC |
| Behind the Tape Reader | BTR |
| Modified Cost Unit | MCU |
| Flexible Manufacturing Cells | FMC |
| Digital to Analogue Conversion | D/A |
| Analogue to Digital Conversion | A/D |
| Integrated Circuit | IC |
| Transistor Transistor Logic | TTL |

LIST OF FIGURES

PAGE

PAGE

PAGE

# CHAPTER ONE

# INTRODUCTION

## 1.1. Outline of Research

The last two decades have seen rapid advances in the field of Integrated Circuit electronics and computer technology. As a result, integrated circuit based electronic components in general, and computers in particular, hold an exceptional status in that they are some of the few products these days that while becoming more effective, are cheaper even in absolute values. Computers, for example, are now a thousand times faster than 20 years ago, while the computation costs are 99% cheaper (1). Therefore, the attractions of economical viability coupled with almost unlimited power and versatility offered by computers, have culminated in them being incorporated into every stage of industrial process and manufacturing technique (2). Popular terms such as Computer Aided Design (CAD) and Computer Aided Manufacture (CAM) are the outcome of such developments. CAD/CAM though well established in many areas of manufacturing technology, are still the subjects of lively research works and every day find new spheres of application. The department of Production Technology and Production Management at the University of Aston with its highly technological oriented interest and capabilities could not consequently remain indifferent and had the necessary resources to undertake new research and development in this field.

Computer Aided Manufacturing is defined as the use of computers to aid in the manufacture or production of a part and in this very broad sense, CAM could include the design stage of manufacturing process too. This of course implies the use of computers as aids to the design and manufacturing processes themselves as well as providing the link between them (3). The development of such a total computer aided system was envisaged in this work and therefore, two main areas of computer involvement in production processes were investigated.

### 1.1.1. Development of a DNC System

As the first step towards establishing the basis for Computer Aided Manufacture, it was decided to incorporate a $2\frac{1}{2}$ axis Cincinnati Machining Centre in a Behind the Tape Reader (BTR) configuration (4,5) into a DNC system (6) which would be controlled by a Data General Mini Computer available in the department.

The main concepts upon which DNC systems are based include: i) Eliminating troublesome tape handling by using computer memory, ii) reducing the NC controller cost by replacing hard-wired logic by software in the host computer, iii) providing management with the up-to-date production information, iv) increasing the overall flexibility and efficiency of groups of NC machines by improved work planning (5).

In interfacing the NC machine to the mini computer other advantages were also sought which were well worthy of being studied in a practical situation. It is a fact (7) that the basic mechanical hardware of the machine tool has a life of 10 to 20 years, whereas the life of the corresponding control system is very much shorter, both in terms of technology and component life. Therefore, many of the first and second generation of NC machine tools have either been discarded due to control problems or are being under-used because of control limitations. With a computer in line such machines will have a new lease of life, many control limitations could be overcome by software development and any updating can be accommodated in the software as a kind of protection against premature obsolescence.

This work started by interfacing the Data General Mini Computer to the NC machine. Electronically the interface was divided into two sections; one at the computer end and the other at the machine end. Data transfer between the two parts takes place in serial form so that the effect of electrical noise on transmitted data is minimised. In order to relieve the computer to attending other peripherals and to ensure that the machine never has to pause while waiting for data which could cause dwell marks or broken tool, double buffer technique was used in which two equally sized data areas are provided, one at each end of the interface to make certain of a continuous flow of information.

In addition to the manufacture of the interface, it was also necessary to develop a set of control programs. The main function of these programs are: data transfer control, code conversion, format manipulation,and valid data check.

### 1.1.2. Development of a CAD/CAM Software

The design and manufacture of curved surfaces using traditional methods has always presented difficulties for both designer and pattern maker. The use of NC machines for machining of such parts, without computer assistance in preparation of control tapes is also expensive and tedious.

In recent years mathematical representation of curves and surfaces has been introduced as a way of overcoming these difficulties. This of course has been stimulated as a result of developments in computer technology, computer driven cathode ray tube displays and numerically controlled machine tools.

The mathematical representation of objects has the advantage over the conventional techniques such as engineering drawing, in that; i) it helps to minimise the ill defined areas of the component, ii) greatly facilitates production of NC part programs, iii) most of these methods provide analytical features (8), enabling calculation of volumes, areas, stresses, etc.

In computer aided design systems based on mathematical representation, the choice of a mathematical method depends on criteria that vary according to the particular application (9). However, there are certain basic requirements that have to be satisfied in any successful system. In Forrest's words (10) "The mathematical representation must not only lend itself to computation, both to facilitate the production of drawings and to aid the analysis of the shape properties, but it must also be compatible with the particular computer peripherals such as displays, plotters and drafting machines, compensating where necessary on their inadequacies".

In the last several years the B-spline basis (9,11) due to its favourable properties has gained widespread popularity. It satisfies many of the major requirements of computer aided design and offers effective "handles" to aid the designer in shape manipulation.

In order to study some of the problems associated with computer aided design and manufacture and in an attempt to accomplish an integrated CAD/CAM system, an experimental software package has been developed for computer aided design and machining of sculptured surfaces.

The system uses a combination of spline curves and is based on lofting techniques in which sectional curves are represented by m-degree ($1 \leqslant m$) uniform parametric B-spline curves and the surface is interpolated between the sections using cubic cardinal splines. The system was developed to allow ab-initio design of curves and surfaces. However, in order to include constraints as well as the existing curves in the surface design elements, provisions for curve fitting have also been added.

The link between design and manufacture in the system is achieved by using the mathematical representation to compute normals at the surface which will then be used to off-set the tool by its radius over the surface. The package also includes "look-ahead" routines, which check and prevent any cutter interference while machining with the adjacent area. The latter routines, after interference checking, produce NC machine formatted output for machining of the part in the DNC cell.

The package is written in FORTRAN IV and is implemented on the ICL 1904 computer. The programs do not run interactively, but are interfaced with plotting routines for generation of graphical outputs.

# CHAPTER TWO

# A REVIEW OF DEVELOPMENTS IN NUMERICAL CONTROL

## 2.1. Milestones in Numerical Control

From the turn of the Century when mass production methods evolved in the U.S., there was virtually no change in the method of machining a component until the 1950's when the NC machine was evolved, again in the U.S.

In the years that followed World War II, the needs of the aerospace industries made demands on the capabilities of machine tools which could not be met by the conventional designs. The initial need was for a machine tool to machine the curved surfaces of large components in the aircraft industry (12).

By 1952 a prototype NC milling with three axes control was built in the Massachusetts Institute of Technology (13) under a project sponsored by U.S. airforce. This new technique aroused great interest and programmes based on the MIT projectwere followed through development works by major machine tool manufacturers.

Early machine control systems were based on various vacuum tubes of large size and required considerable space. The second generation control systems appeared in 1959 and were based on discrete electronic components. The technology of small and then medium scale integrated circuits had a great impact on design of control systems and the third generation of NC using Integrated circuitrywas introduced in 1965. Technical advances in production of integrated circuit chips and rapid fall in the cost of their production also sharply reduced the price of mini computers and gave economical viability to the use of mini computers for control purposes. Although since 1964 general purpose mini computers were incorporated in control systems, 1970 machine tool show in Chicago heralded the start of the fourth generation (14).

The need to standardise the control systems and progress in the field of large scale Integrated Circuits resulted in the use of microprocessors in numerical control, (15) particularly open loop systems, in that positioning requirements are handled through stepping motors. This type if not an extention of the fourth generation, may be regarded as the fifth generation of numerical control systems and has been available since 1974.

## 2.2. Introduction to Numerical Control

If one takes an operator controlled machine; the information is fed into the machine via hand wheels, switches, etc. The operator monitors the cutter position during machining and makes any necessary corrections to ensure suitable output. Here the operator carrys out data processing, position input, position feed back and compensation functions.

In a numerically controlled machine position command information is fed to slideway transmission elements by the control unit. The component drawing must therefore be translated into a form acceptable to the control.

Probably the most widely accepted definition of Numerical Control (NC) is that given by Electronic Industry Association (EIA), that is:

> "A system in which actions are controlled by the direct insertion of numerical data at some point, the system must automatically interpret at least some portion of this data".

The numerical data is introduced into the system by a stored input medium which may be punched card, punched paper tape, magnetic tape or magnetic card. Data could also be given to the system manually(15).

## 2.3. Methods of Tool Positioning Control

The main function to be performed by a control system on a machine tool is the displacement of the machine slides, since the slides are displaced to alter the position of the tool relative to the workpiece and produce the component to the required drawing dimensions.

Most machine tools have two or more slideways, disposed at right angles to one another, along which the slides are displaced. In addition to displacement along the axes, rotary displacement about the axes are also of normal feature of more complex NC machines.

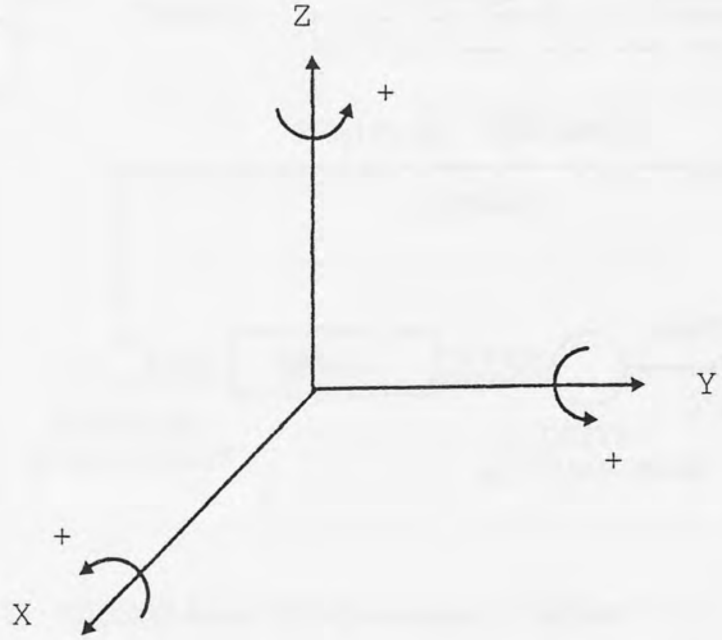


Fig. (2.1) Conventional Method of Defining Linear and Rotary Displacement of Machine Tools.

Control of slide displacement can be by either open or closed loop. In a closed loop system, Fig. (2.2), the command signal is constantly being compared with the actual position and the difference, or error, is fed through an amplifier to actuate the drive motor until the difference between the command signal and actual position reaches zero.

In most closed loop control systems for smooth running of the slides, input to the servo system is in analogue form. Therefore, digital information must be converted into analogue signals. Hence,

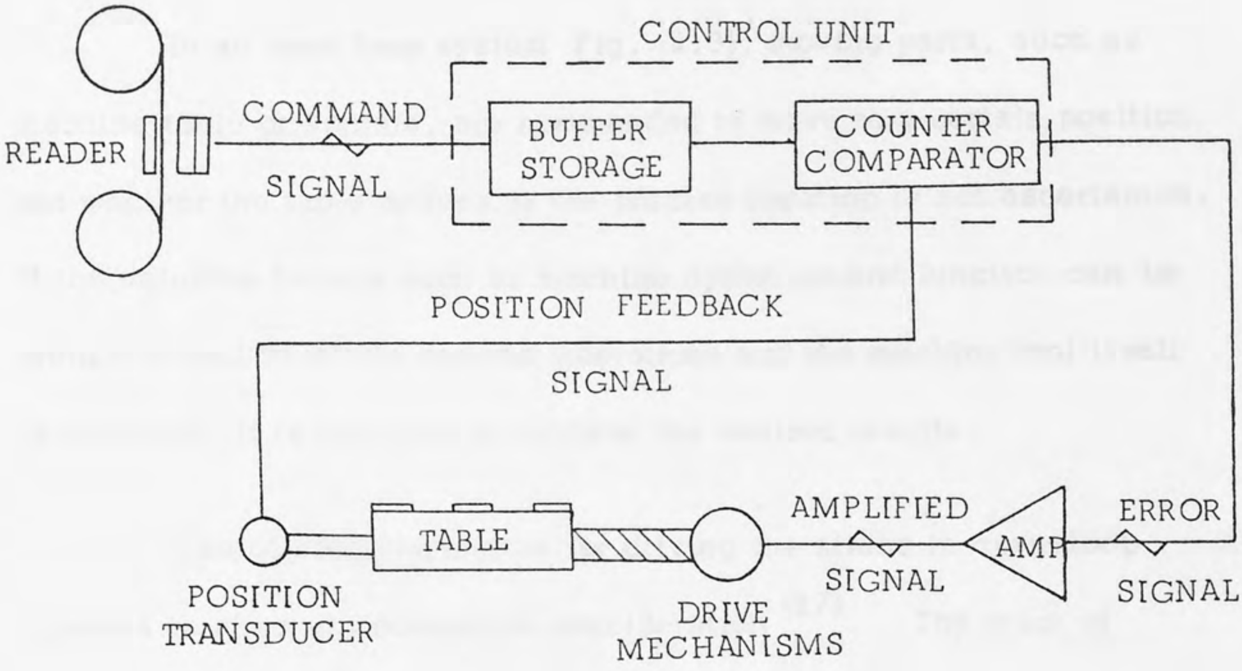


Fig. (2.2) Closed Loop Displacement Control

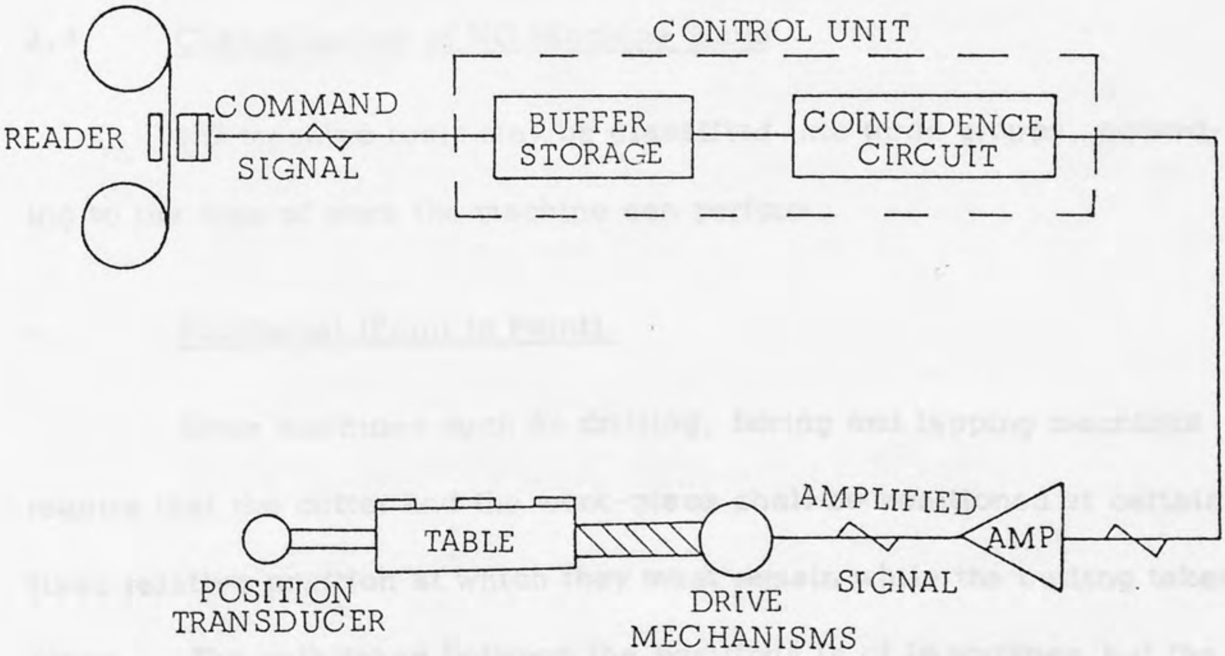


Fig. (2.3) Open Loop Displacement Control

A/D convertors are of essential parts of the system.

In an open loop system Fig. (2.3), moving parts, such as machine table or spindle, are commanded to move to a certain position, but whether the table arrives at the precise location is not ascertained. If the valuable factors such as machine dynamics and function can be predetermined to within desired tolerances and the machine tool itself is accurate, it is possible to achieve the desired results.

Use of stepping motors for driving the slides in open-loop systems is of great economical consideration (17). The price of positioning servos using stepping motors is lower owing to the fact that these can perform the functions of both driving and measuring. Furthermore, the application of such motors eliminates the need for any A/D or D/A convertors.(18)

## 2.4. Classification of NC Machine Tools

NC machine tools may be classified into three groups, according to the type of work the machine can perform.

### a. Positional (Point to Point)

Some machines such as drilling, boring and tapping machines require that the cutter and the work-piece shall be positioned at certain fixed relative position at which they must remain while the cutting takes place. The path taken between the positions is of importance but the speed of motion should be as high as possible to minimise the time.

### b. Paraxial (Straight Line)

The work-piece and the cutter move with respect to one another while cutting is carried out, but in which the motion, while cutting, lies along a straight line. Paraxial machines are normally used for surface milling or contouring - out of components with sides parallel to the machine axes.

### c. Continuous Path (Contouring)

Machines such as milling, routing machines, flame cutters etc., are known as contouring machines because they involve the motion of the work-piece with respect to the cutter while the cutting operation takes place. Obviously most contouring machines have some capabilities of the point to point machines, but it is generally uneconomical to use them only as point to point systems.

Contouring machine tools need to have independent control of the motors driving the individual axes. This will allow motion of the machine table in any direction (19).

## 2.5. Types of the Existing Control Systems

### 2.5.1. Conventional Numerical Control of Machine Tools

In conventional NC the automatic control of machine tools is achieved by a special purpose logic circuit designed as a hard-wired controller which translates small quantity of input data into a larger

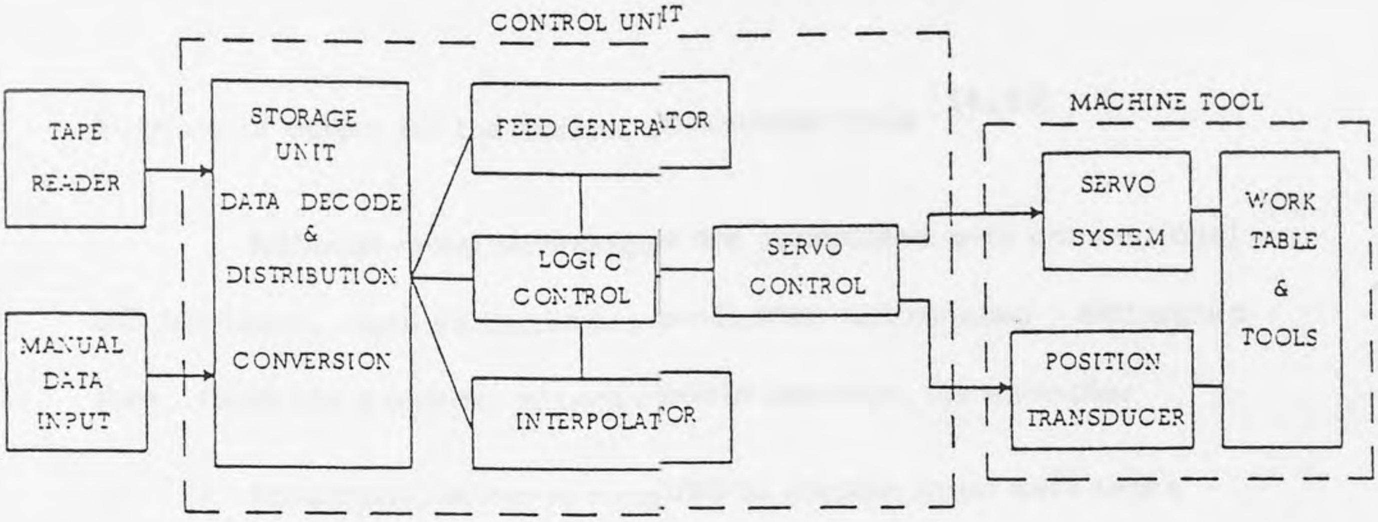


Fig. (2.4). Block Diagram of a Typical Hard-Wired NC System

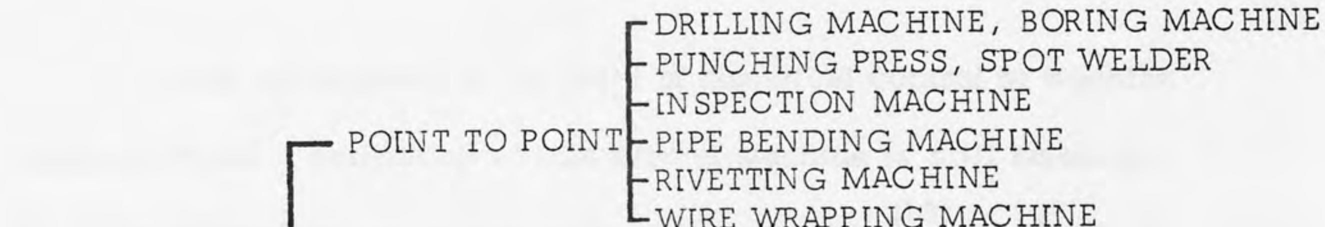


Fig. (2.5) Types of NC Machine Tools and Equipment quantity of output for the control of machine tools (14,20).

Although many advantages are associated with conventional NC machines, such as increased production and reduced setting-up time, there are a number of undesirable features, for example:

a) Substantial effort is required to prepare input data tapes satisfactorily.

b) A tape reader recognised as the weakest link of the NC machines, is required for each machine tool.

c) An NC controller, relatively inflexible in operation, is also required for each machine (21).

With all progress in the field of numerical control of machine tools the World's population of this type of machine is still small but the range of application is very wide and impressive (13) as shown in Fig. (2.5).

### 2.5.2. Computer Numerical Control of Machine Tools

CNC is defined as a control system in which a mini computer replaces the electronics in a conventional numerically controlled system to perform some or all of the basic NC functions of one machine or a number of machine tools, Fig. (2.6). The control programs stored in the mini computer is used to implement these basic functions.

It has been argued (22) that the use of computers for control

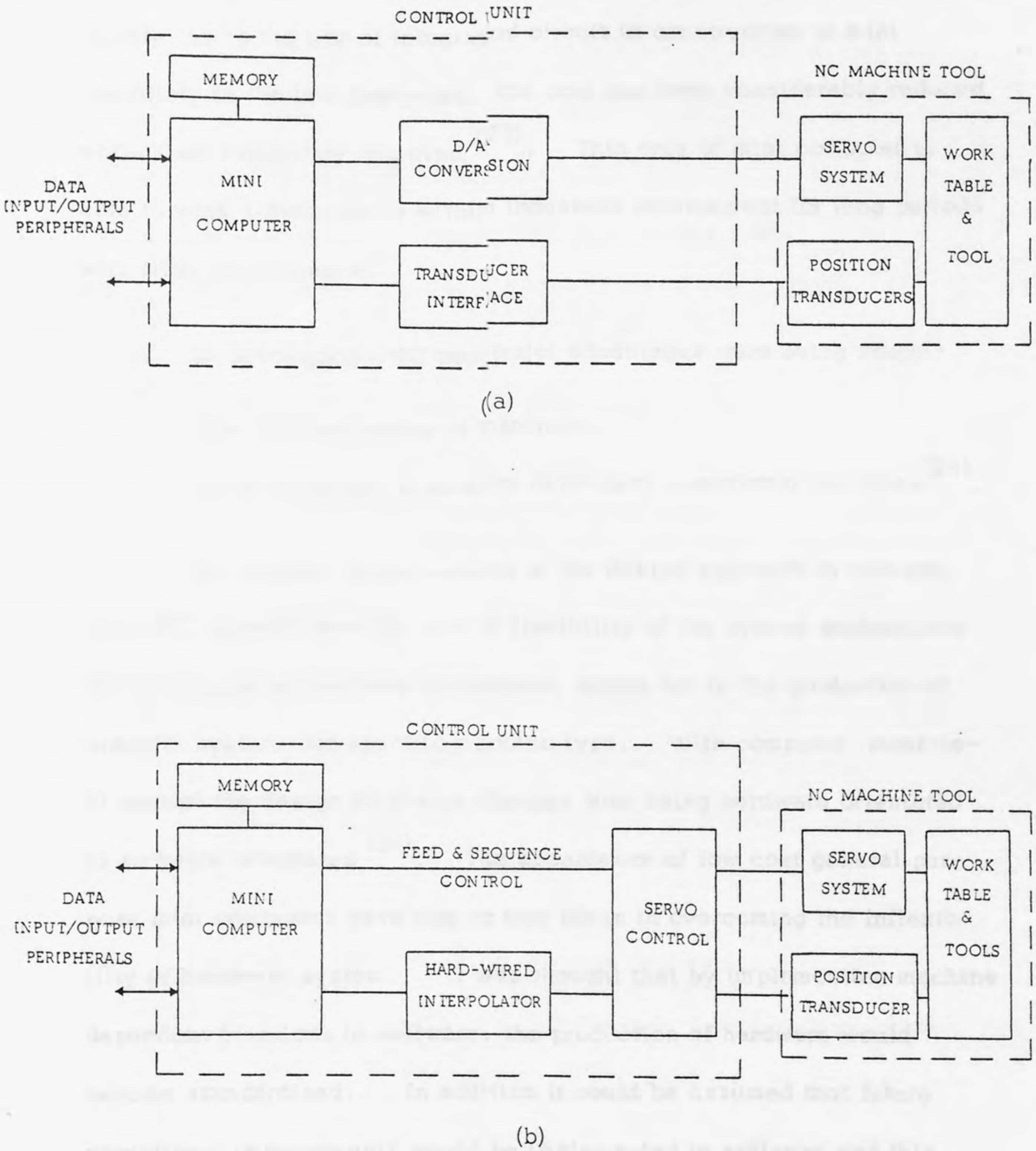


Fig. (2.6) Block Diagrams of CNC Systems

(a) Full Computer Control

(b) System Incorporating Hard-Wired Interpolator purposes was obvious for many years, but that it was the advent of small, low cost and highly modular computers that made CNC a possibility. Mainly due to the use of integrated circuit in construction of mini computers in the last few years, the cost has been considerably reduced while their reliability improved (23). This type of mini computer is able to work satisfactorily within industrial environment for long periods with little maintenance.

In developing CNC two major advantages were being sought:

1. The standardisation of hardware,
2. the availability of memory dependent operational facilities (24).

An inherent characteristic of the design approach in conventional NC systems was the lack of flexibility of the system architecture due to the use of hardware techniques, which led to the production of specific system for specific machine type. With computer numerical control the design emphasis changes from being hardware orientated to software orientated (25). The appearance of low cost general purpose mini computers gave rise to new ideas in overcoming the inflexibility of hardware system. It was thought that by implementing machine dependent functions in software, the production of hardware would become standardised. In addition it could be assumed that future operational improvements would be implemented in software and this was a guarantee against obsolescence.

The presence of mini computers and availability of relatively cheap memory could provide a number of operational facilities which in hardware system either due to technical impracticalities, or high cost was not possible. These features include:

a) Part program storage and shop-floor edit,

b) multiple number of tool offset and compensation,

c) large capacity operation message/instruction display,

d) input part program tape compatibility,

e) machine tool interface compatibility.

It has been claimed that the first generation of CNC systems shows that in general memory based features have materialised and hardware has indeed become standardised (24).

The edit facility is perhaps the most valuable feature of CNC, because it reduces the time needed to prove and correct the tape. Some CNC users say it increases the productivity of programmers up to five times (26).

Canned cycles, that is a sequence of movements for certain operations can be held in store and used only by adding primary dimensions. Also sections of computation for recurring details could be held in store and called up when required. The use of paper tape reader could also be minimised by storing the part program when machining a batch of parts.

Lee Cole (27) Marketing Manager of Cincinnati Milarcon (Process Control Division) argues that NC machines are actually cutting no more than 35 percent of the time and this using CNC could be increased to 43 percent, which represent a considerable saving.

The use of general purpose mini computers as controller of a CNC system is cost effective only if it can handle a wide range of functions; such as input data handling and distribution, feed rate computation, interpolation, servo computation, cutter compensation and machine logic. Therefore, a high performance and as a result expensive mini computer is needed, one with powerful byte and bit manipulation instruction for handling input data and logic processing and also double or triple length arithmetic instructions to carry out the calculations accurately. Probably the worst feature of all gneral purpose mini computers is that they are not ideally suited for machine tool control since they are basically word orientated or at least byte orientated. Their capability for addressing and manipulation of bit logic leaves a lot to be desired. In addition the design of the software logic is no simpler than a new design of hardware logic and initially will probably cost more (25). The price of high performance mini computers, cost of software preparation and hardwares inevitably needed in design of CNC systems, result in a non-optimum price/performance solution on machines other than for the most complex, for example, four axis lathes and five axis machining centres etc. (24)

### 2.5.3. Direct Numerical Control of Machine Tools

DNC is a system in which a set of machines are simultaneously controlled from a main computer. The main computer could take over the machine logic control or merely have a supervisory role in co-ordinating the activities and data distribution and storage. DNC is predominantly a system for communication of information and is capable of issuing up-to-date reports dealing with down time, cost of components, maintenance requirements, inventory control etc.

DNC first appeared in U.S.about 1968 (5). There has also been considerable interest in Japan and Germany in DNC systems. In Japan Fujitsu FanucLimited claims to have installed 70 DNC systems in that country alone.

Generally, these systems are restricted to large companies which can afford the large capital outlay and provide technical expertise to run the system.

The design of direct numerically controlled systems is still in the stage of flux and there is no standarisation. Therefore, the system of this kind available on the market or in design stage are totally different. The reason is that there are three different suppliers of DNC systems; machine tool manufacturers, machine tool users, and builders of control systems. Each group has its own idea about the design and a different criteria on the level of involvement of the contol computer of the system in control of machine tools.

In general there are two major approaches to the design of DNC systems, one is called "maximum flexibility" and the other, "minimum cost" (4, 28) which considering the role of computers are based on :

1. Behind the Tape Reader method (BTR), and
2. Systems with built-in Machine Control Unit (MCU) respectively.

### 2.5.3.1. DNC Behind the Tape Reader System (BTR)

In this system a main computer executes and transfers data in parallel to several machines. The tape reader is by-passed in the interfacing of the mini computers to the controller, but both the tape reader and the conventional controller, hardwired or otherwise, are retained, Fig. (2.7). This system offers maximum flexibility since each machine tool can be operated either as an independent NC system or as part of a DNC system.

The price that is paid for this kind of DNC is the cost of complete NC machine tool plus the cost of interfacing equipment to each machine in addition to the control data storage equipment. Thus, the only disadvantage of this system is the high initial capital investment, otherwise all the advantages of the conventional numerical control that have evolved over years, have been retained in addition to benefits that computer control can offer.

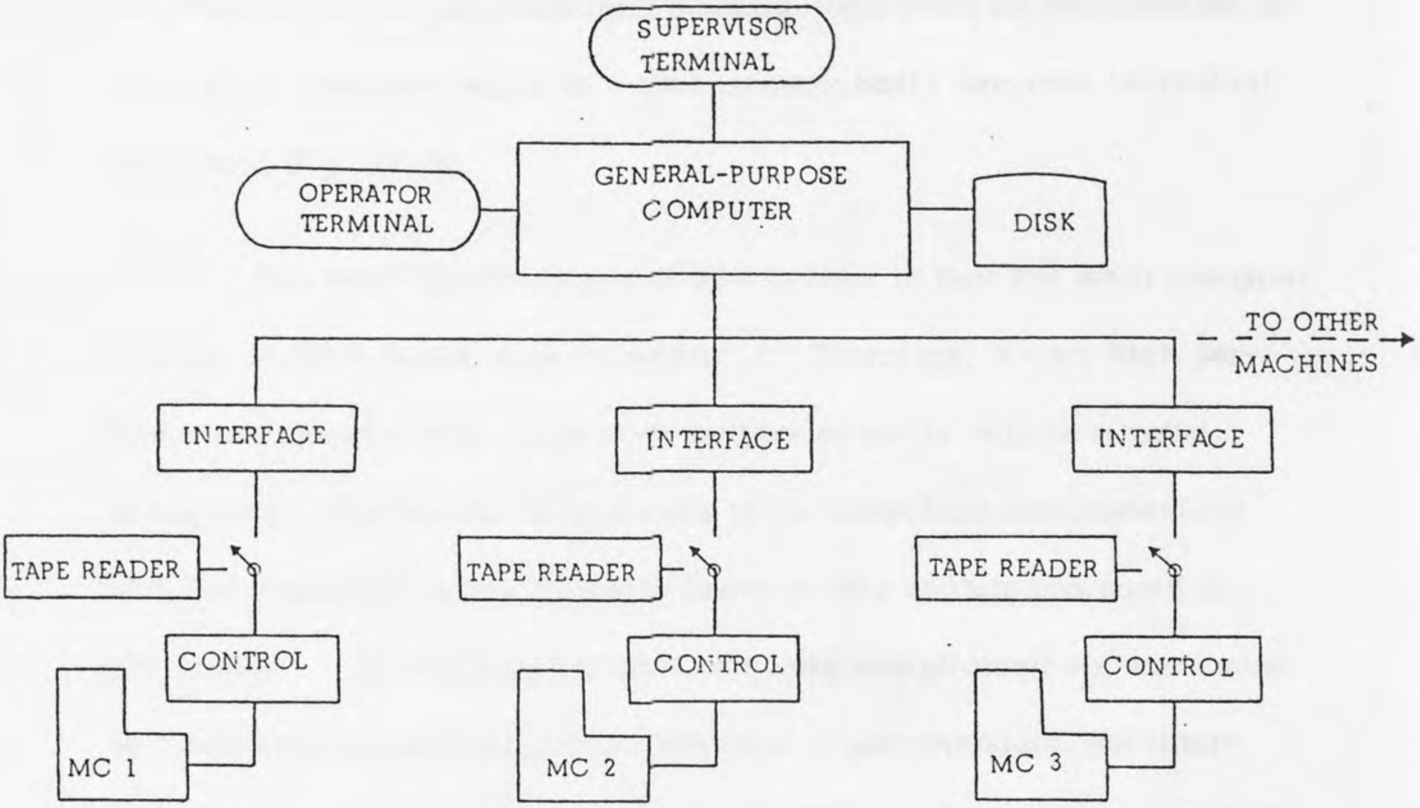


Fig. (2.7) Block Diagram of a DNC Behind the Tape Reader System

### 2.5.3.2. DNC Machine Control Unit (MCU)

In this system most of the control functions are performed in software and only few in the hardware, i.e. in machine control unit. In an extreme case, a minimum cost DNC can be designed if almost all electronic equipment at machine tool are eliminated and only necessary parts such as servo drive motors, feed back arrangements for a closed loop system and other equipment that may be required for data transmission and distribution are kept. In this case, all control functions and machine logics for individual machine tools must be implemented in the control computer which in a time sharing basis services individual machines, Fig. (2.8).

The main disadvantage of this system is that the main computer will have a very heavy task to perform. Therefore, a very high performance computer with large core memory capacity will be needed. In any case, the number of machines to be controlled simultaneously with one computer is significantly fewer in this system compared to BTR system. In addition the cost of a very complicated software must be taken into account as well as the cost of non-standardised interface that is necessary as machines with different designs would require different types of interface.

To solve the problems associated with this approach, the second category of DNC machine control unit has been developed. This is termed "MCU modified minimum cost" (4), Fig. (2.9).

This is a less restricted system in that it allows some hardware to be kept at the machine end of the system and some control functions, for example, interpolation be performed out of the computer. This reduces the rate of data requirements and consequently, allows more machines to be connected to a single computer. In a system designed by Warner & Swasey Co. based on this approach (4), circular interpolation is carried out in two stages. First a coarse interpolation

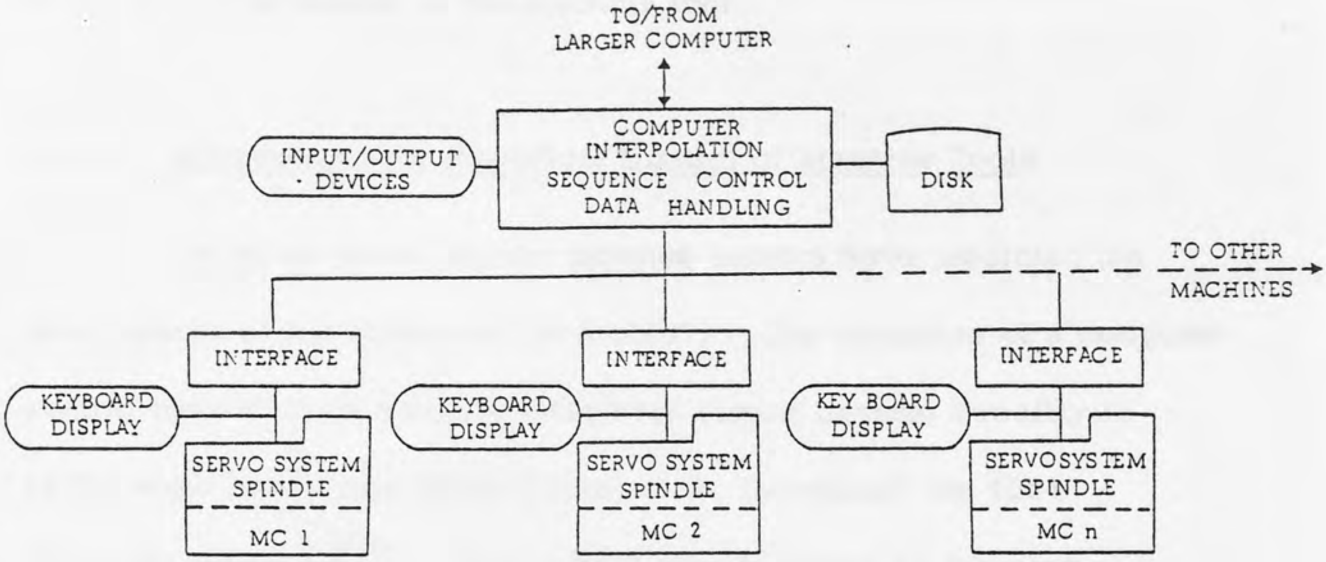


Fig. (2.8) A DNC Machine Control Unit System

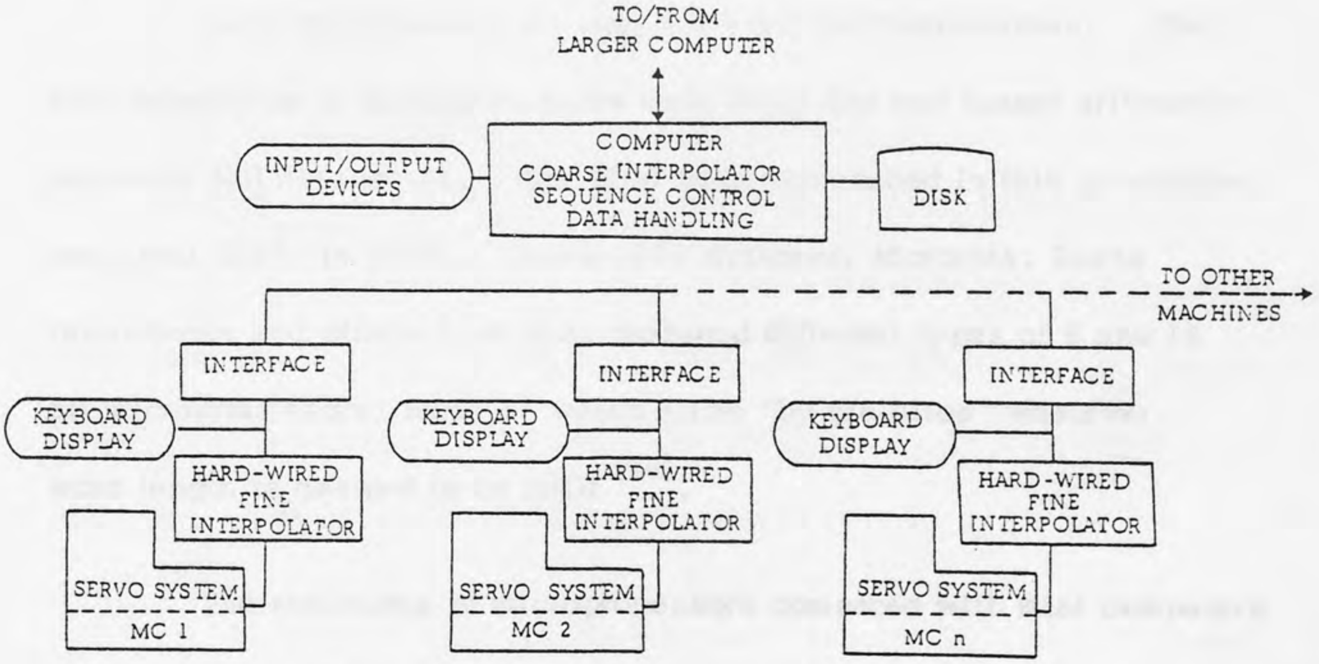


Fig. (2.9) MCU Modified Minimum Cost DNC System is carried out by the computer and the intermediate points are established, then the path of the cutting tool between these points is determined by the fine interpolator at the machine tool.

### 2.5.4. Microprocessor Numerical Control of Machine Tools

For many years popular science authors have predicted the development of the "computer on a chip". The objective of a computer central processor on a single integrated circuit became a reality in 1970, when Intel Corp. Santa Clora, C.A. introduced the 4004 "microprocessor".(29) This circuit was designed as the control element for a cash register, but was organised like a computer.

Early microprocessors were all 4-bit microprocessors. The next generation of microprocessors were 8-bit and had longer arithmetic registers and data words. The first to be introduced in this generation was Intel 8080, in 1973. Since 1973 National, Motorola, Texas Instruments and others have also produced different types of 8 and 16 bit microprocessors, some of which allow "by the slice" whatever word length is desired to be built (30).

The structures of microprocessors compared with mini computers though similar, there are a few differences which must be taken into account. The word length is normally shorter in microprocessors which limits the number of bits and the number of devices to be addressed. Microprocessors have longer execution time per instruction and their interrupt capability is simpler.

As discussed previously, the mini computer based CNC systems proved that the advantages of computer numerical control is achieved only at the most complex end of machine tool spectrum. The advent of microprocessors in early 70's has opened up large possibilities for cheapter and more effective NC systems. It also brought about new approach to system design.

Kean, G. C. & Savage, R, (24,31) analyse an NC system into three different areas: data input, computation, and sequence control. In these major areas the individual functions can be controlled using microprocessor, dedicated hardware,or other control techniques. However, each area shows somewhat different characteristics that will affect the design. The first area, data handling and distribution requires 8-bit byte manipulation capabilities, the second area needs powerful multi-byte arithmetic capabilities for interpolation and cutter compensation, and finally sequence control requires efficient bit manipulation in handling the machine logic functions. This analysis therefore, calls for a decentralised approach.

The power of this approach is that each function area becomes a specifiable entity within itself, communicating with other function areas over a universal data bus as indicated by Fig. (2.10). Thus, each function area design may be implemented in the optimum way for that function without regard to other function areas apart from ensuring interface compatibility with data bus. In addition this method enables different functions to take place simultaneously, contrary to the classical CNC's where all operations have to be performed sequentially as there is only one processor. Redesign only partly affects the control structure, and lastly, the physical separation of different functions can be important in integrated systems where eventually single modules can serve several sub-systems.

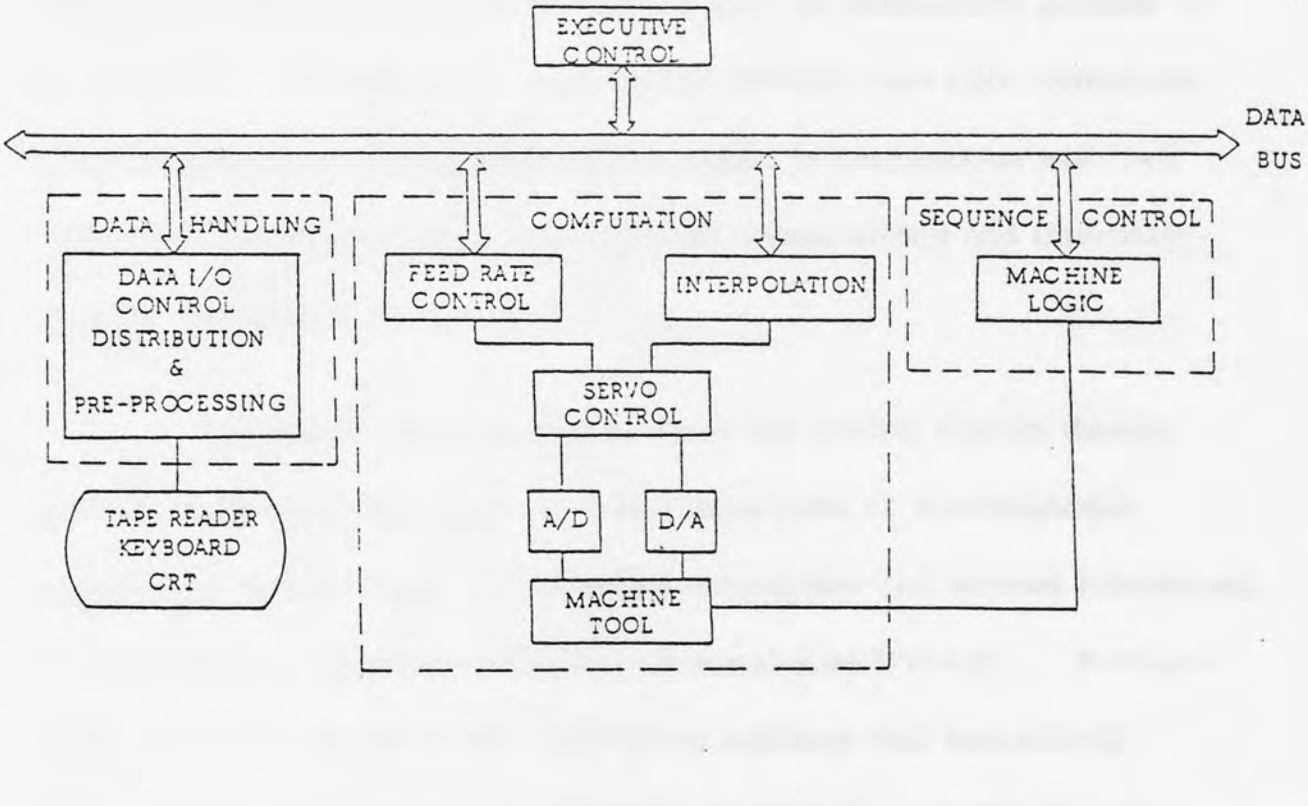


Fig. (2.10) Function Diagram of a Decentralised Approach in Design of NC Systems

The flexibility offered by the potential of the microprocessor made a variety of products in the field of machine tool control available on the market. Among these products the Philips 6600, Siemens System 5 and System 7, and Plessy's RUSC can be mentioned all of which are structured on a data highway.(31) ACTRON III System developed by McDonnell Douglas which is capable of controlling 2-5 axes is also a good example of systems based on decentralised design approach.(32)

In a CNC system a microprocessor can either be a master device or a few microprocessors could be dedicated to different functions. An Intel 8080 microprocessor used by Bolinger (30) proved to be capable of handling a relatively complete 3-axis point to point machine with a foreground program to control the drives and the background program to monitor I/O. On the other hand, in the Swedish new CNC controller, NUCON 400, a control microprocessor works in conjunction with two additional microprocessors which perform computations and interpolations for continuous path control (33).

The impact of microprocessors in the control system design was so great that some major manufacturers such as Westinghouse, stopped the manufacture of hard-wired controllers and instead introduced microprocessor controlled systems (Westinghouse W2560). Furthermore, on the evidence of the successive machine tool exhibitions since 1977 microprocessor based CNC systems are now the dominant form of control system. The introduction of new "play-back" systems is also a result of the impact of the advent of micro-electronic and particularly microprocessors. Here a skilled operator machines the first component in the batch using console controls. The instructions given are recorded and can be used in automatic mode to machine the subsequent components. One of the earliest systems of this kind was the Kremer 100 self-programming control using magnetic cassettes, while Siemens have introduced MATE-M 3-axis system using a semi-conductor memory which can store up to 230 instructions (1).

Finally, in addition to attractions such as more compact control design, reduction in hardware cost (33) and flexibility offered by the use of microprocessors, extensive diagnostic testing facilities are also of the main feature of recent systems (34).

## 2.6. Integrated Manufacturing Systems

Machining centres were among the first approaches to integrate all machining operations to be performed at one setting in an effort to reduce setting up time and to eliminate handling between machines. The main limitation of this method is that machining centres to be capable of completing all operations in one setting tend to be extremely expensive. In addition such expensive machines will be spending most of their time carrying out such operations as drilling and tapping, which could be done on much smaller, less complicated and cheapter machines at much lower cost.

There are strong indications that the trend in automation of machine tools and manufacturing would be towards realisation of the computer - integrated automatic factory.

Already in Japan, the U.S.A and elsewhere (35) direct numerically controlled machine tools, handling devices and transportation conveyors are efficiently integrated to constitute a manufacturing system in which the system control software both controls the timing relation among these equipment and gives advance production schedule.

DNC is the most promising unit to constitute the future of automated manufacturing system. It is worthy of attention that most of DNC systems are justifiable both economically and technologically.

It is desirable that numerically controlled machine tools should be fully loaded. Consequently, production components should be identified and channelled efficiently to the most economical machine tool for carrying out specific operation, the result has been the introduction of technique of group technology involving production and component systems of classification.

The most recent development in this field, taking advantage of the technique of group technology was the emergence of 'Flexible Machining Cell" (FMC) (1,36) which is a single CNC unit comprising controls, machine tools and work handling devices capable of performing all the required operations on a family of components. The cells are linked in a hierarchical manner to form a flexible manufacturing sytem Fig. (2.11).

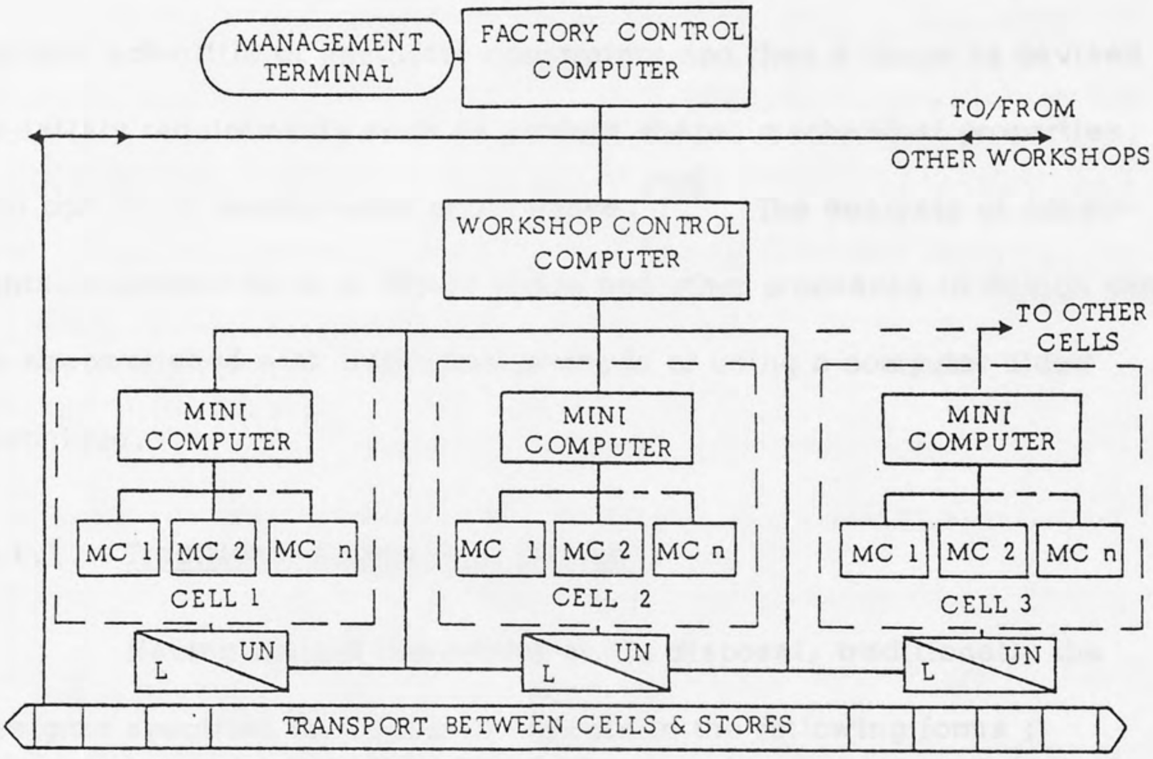


Fig. (2.11) Flexible Machining Cells in an Integrated Manufacturing System

# CHAPTER THREE

# A REVIEW OF COMPUTER AIDED GEOMETRIC DESIGN OF FREE FORM SURFACES

## 3.1. Considerations for Design

The design of mechanical objects starts with the study of the various scientific or aesthetic constraints and then a shape is devised to satisfy requirements such as product shape, mechanical properties, and optical or aerodynamic performance.(37) The analysis of constraints, representation of object shape and other processes in design can be accomplished with traditional methods or using a computer aided technique.

### 3.1.1. Traditional Approach to Design

Having limited component at his disposal, traditionally the designer specified the designed objects in the following forms :

a) in the form of a model,

b) in the form of an analytical surface,

c) in the form of a free hand sketch,

d) in the form of a conventional engineering drawing,

e) in the form of a tabular specification of points which lie on the devised surface,

f) in the form of displacement and slope requirements at selected points on the surface.

This method of design has major deficiencies, for it is time consuming and in many cases ambiguous, and may need special skills such as artistic and analytical ability.

### 3.1.2. Computer Aided Design

In general terms "Computer Aided Design is a technique in which man and machine are blended into a problem-solving team, intimately coupling the best characteristics of each, so that this team works better than either alone" (38). If computer is used for design of mechanical objects, the object is in a way modelled in the computer, consequently, a more comprehensive representation and study of design parameters can be carried out.

The overall system in computer aided design may be divided into hardware and software. The support computer and graphics devices, interactive or otherwise, constitute the hardware. The software can be divided into two parts, first the software system that provides the interface between the normal drawing functions and the graphics devices. This is a device dependent software and is written for use on a specific device.

The second is the data manipulating software and is used for geometrical description of the item to be designed. The mathematical basis for this part of CAD is the main topic of this Chapter. Due to the fact that graphics devices are an inseparable part of any computer aided design system, some important types of these devices are also briefly discussed.

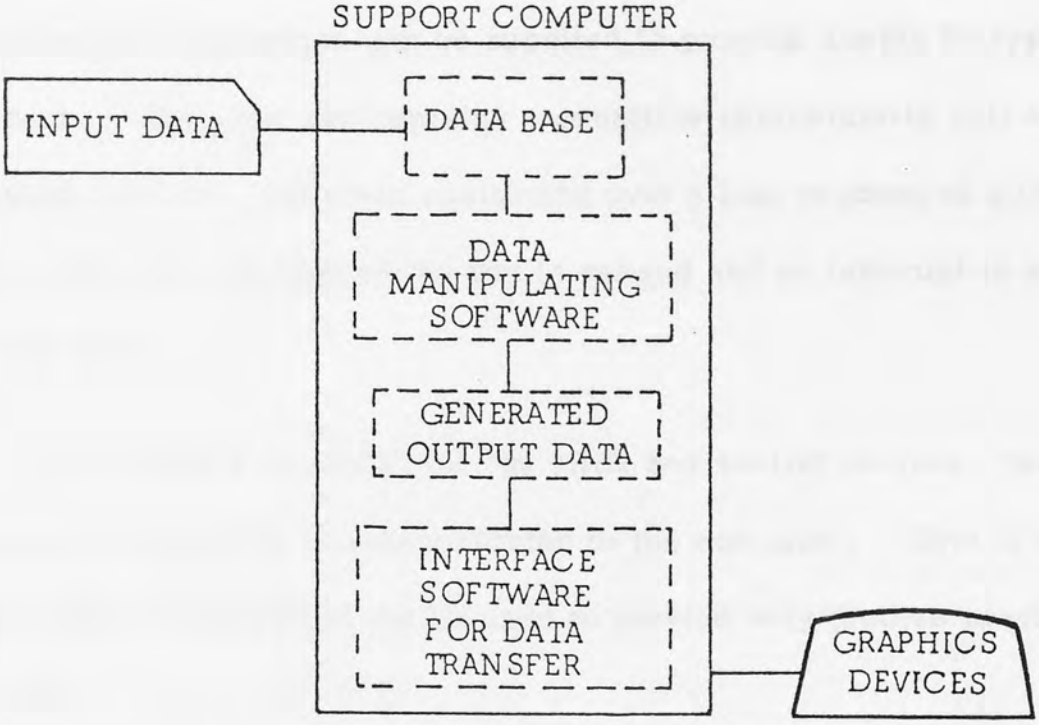


Fig. (3.1) Block Diagram of a Computer Aided Design System.

## 3.2. Computer Aided Design Hardware

### 3.2.1. Interactive Devices

Interactive devices communicate with the program while it is running and in effect this interrupts the program so that new or different information can be provided. Numerous devices have been used to accomplish this task, the simplest of which is of course, the alphanumeric keyboard. More sophisticated devices include light pens, joy sticks, control dials, and analogue tablets.

Using a keyboard as an interacting device, alphabetic, numeric and control information can be supplied to program simply by typing in the data. The light pen contains a sensitive photoelectric cell and associated circuitry, and when positioned over a line segment of a CRT and activated, the position of the pen is sensed and an interrupt is sent to the computer.

By moving a control in the joy stick and similar devices, two-dimensional positioning is communicated to the computer. This is an analogue device and should not be used to provide very precise positional information.

The analogue tablet is the most versatile and accurate device for communicating positional information to the computer. Associated with the tablet is a pen which can be moved over the surface and whose position can be sensed. The position of the pen and its relative location in the picture-display area are tied together by means of a cursor whose motion on the picture display area is in concert with that of the pen on the tablet. The analogue tablet is used to perform a pointing function, the indication occurs in the data base and not in the display file. Thus, the programming is simplified. Also, drawing or sketching or pointing using an analogue tablet which is on a horizontal surface is more natural than performing the same action with a light pen in a vertical orientation.

### 3.2.2. Types of Graphics Devices

There are a large number of different types of graphics devices available. Here only a few important types, in particular, three different types of CRT (storage tube, refresh and rastar scan), pen and ink plotter, and dot matrix plotter are described.

#### 3.2.2.1. Cathode Ray Tube Graphics Devices

The direct view storage tube display, also called a bistable storage tube, can be considered as similar to an oscilloscope with a very long persistence phosphor. A line or character will remain visible until erased (up to an hour). Storage tube displays have several advantages ; some of the advantages are: the display is flicker free, cost is low, further, with them it is easy and fast to obtain an acceptable hard copy, and conceptually they are easier to program and more suited to time sharing applications than refresh and rastar scan displays. The main disadvantage is that the screen can not be selectively erased; in order to change any element of a picture the entire picture must be redrawn. Because of this the display of dynamic motions is not possible. In addition, this characteristic results in the interaction between the user and the display being somewhat slower than with a refresh display.

A refresh CRT graphics display is based on a television-like cathode ray tube and since the phosphor used on this type of graphics display fades very rapidly, it is necessary to reconstruct the entire picture many times each second. This is called the refresh rate. If the refresh rate is too low a phenomenon called flicker occurs. A minimum refresh rate of 30 is required to avoid flickering. An essential part of each refresh CRT is display buffer in which the drawing instruction is stored. Immediately, a limitation of the refresh display is obvious: the complexity of picture is limited by the size of display buffer. However, reconstruction of display makes the dynamic motion possible. Further, since each element of instruction necessary to draw the complete picture exist in the display buffer, any individual element can be changed, deleted or an additional element added. Although a hard copy of the picture in this display is difficult to obtain, undoubtedly for dynamic motion in real-time or very rapid interaction this is the best graphics display unit.

A rastar scan CRT graphics display uses a standard television monitor for the display console. In the rastar scan display the picture is composed of a series of dots. These dots are traced out using dual rastar scan technique. The basic electrical signal used to drive the display is an analogue signal whose modulation represents the intensity of the individual dots. In using rastar scan display it is first necessary to convert the lines and character information to a form compatible with the rastar representation. The rastar scan displays are generally somewhat slower, the selective erase feature is more difficult to implement but may be directly interfaced to closed circuit television systems.

### 3.2.2.2. Plotters

Digital incremental pen and ink plotters are of two general types; flat based and drum. Digital incremental plotters can provide high quality hard copy of graphical output. Compared to CRT they are quite slow. Consequently, they are not used for real time interactive graphics.

The electrostatic dot-matrix printer/plotter operates by depositing particles of a toner onto small electrostatically charged areas of a special paper. The particles are attracted to charged areas and graphics are displayed. The electrostatic do-matrix, plotter/printer is a rastar scan device, i.e. it presents information one line at a time. Therefore, a substantial amount of computer storage is required. A further disadvantage is relatively low accuracy and resolution. The principal advantages are the very high speed with which the drawings can be produced and excellent reliability.

### 3.2.3. Classification of Graphics Devices

There are a number of ways of classifying graphical devices. Each method is based on a different point of view and objectives.

Consider first the difference between a passive and an active device. A passive device simply draws pictures under computer control; i.e. it allows computer to communicate with the user graphically. Examples are teletype, line printers, storage tube CRT's and plotters.

An active graphics device on the other hand allows the user to communicate with the computer as well. Usually an active graphics device has the ability to reposition the cursor and read its new position. Typical active graphics devices include analogue tablets, light pens, joy sticks, etc. These devices usually require some type of passive graphics device for support.

Another method of classification is whether the device is point plotting or line (vector) drawing. All the storage tube CRT's and most refresh CRT graphics devices and all pen plotters are line-drawing devices. The rastar scan CRT, high speed line printers and electrostatic plotters are point drawing devices.

Still another method of classifying graphics devices requires determining whether a device can accept three-dimensional data or whether three-dimensional data should be converted to two-dimensional data by application of some projective transformation and presented to sytem. In essence this method requires determining whether graphics device has two or three registers to hold co-ordinate data (39).

## 3.3. Computer Aided Design Software

### 3.3.1. Introduction to Numerical Geometry

Like in many other areas of science and technology, numerical geometry first assumed importance during the second world war. The pressure of production, particularly in aircraft and shipbuilding industries stimulated the development of new techniques in design. Unlike older design methods which were mainly graphical, the new methods were based on analytical curves, conic in particular.

The advent of electronic computer opened the way for the development of more ambitious techniques. One of the first computer aided numerical geometry systems to be developed was that of Ferguson (40). Ferguson used parametric rather than cartesian co-ordinates in its curve and surface definitions. This has since become standard usage, for a number of reasons.

Progress in the field of numerical geometry was of course helped by the advent of computer graphics devices, draughting machines and interactive computer peripherals. Numerically controlled machine tools were also becoming available which complemented production process to the manufacturing stage.

A number of important mathematical developments followed Ferguson's system. The properties of spline curves have been widely studied, as a result a very efficient method of spline curve construction has been developed, namely B-splines or "fundamental splines" as Curry and Shoenberg (41) called it, which have the property of enabling local modifications to be made to a shape during the design process without the need to recompute the whole process. The other major development was the introduction of Coons' theory in surface design by blending curves into a single smooth patch with inter-patch continuity(42).

In 1971 Bezier (43) introduced the system UNISURF and this was certainly an important step in the field of computer aided design. UNISURF is a good example of a successful interactive system. The designer starts with defining a polygon, which is then approximated by a smooth curve, the modification then takes place until the desired curve is produced. Gordon and Riesenfeld (44) in 1974 approached the design of curves and surfaces using defining polygons from a different angle, i.e. to use B-spline rather than simple polynomial which Bezier had done.

Polynomial based functions although providing continuity, are mostly oscillatory. To avoid this draw-back some new ideas have been put forward such as spline under tension (45) and non-linear spline used by the AUTOKON system (46).

A variety of systems for computer aided design has been developed in recent years. For the design of sculptured or free form surfaces, apart from the above mentioned systems, NMG (Numerical Master Geometry) at British Aircraft Corporation (47), POLYSURF (48) and DUCT (7) at Cambridge, BSURF at the University of Leeds (49), OKISURF in Japan (50) are better known.

Products that can be designed by these programs include propellers, car body panels, boilers, air ducts, pipes and exhaust systems, shoe lasts, ship hulls, air craft bodies, domestic appliances, etc.

### 3.3.2. Parametric Description of Curves and Surfaces

The path of a moving point in the three dimensional space may be described by the value of a position vector r at successive instants of time. Therefore, r is a function of time and the relationship can be shown as $r = r(t)$. This is equivalent to :

$$x = x(t), \quad y = y(t), \quad z = z(t) \quad \text{in terms of coordinates.}$$

The time in the above relationship only refers to a point on the path, hence can be replaced by any parameter; say u. It is of course, desirable that any given value for u only represents one point on the curve.

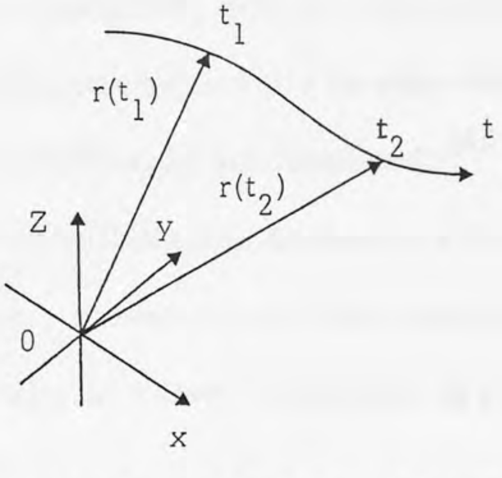


Fig. (3.2) Parametric Method in Curve Description

Now if the curve $r = r(u)$ moves in space, the successive position of this curve generates a surface where each point is distinguished by the time t and parameter u. Here again one can substitute t by any other parameter; say v. Then vector function $r = r(u, v)$ of two variable will represent a surface.

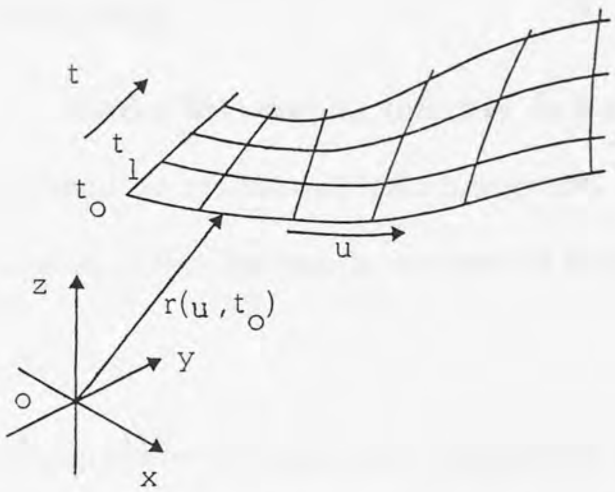


Fig. (3.3) Parametric Method in Surface Description

There are a number of reasons in using parametric curves and surfaces for computer aided design. In the non-parametric method alteration of one point specifying a fitted surface means that the entire surface must be recomputed, and all parts of the surface except at data points will be changed positionally to some degree. In other words the effect of modification is not localised (40). In addition this method enables twisted curves in three dimensions to be represented much simpler. Further, closed curves with vertical tangents can be represented without difficulty in a fixed coordinate system.

Another significant advantage is that translation or rotation of the axes or the object can usually be carried out by translating or rotating the vectors defining the curve without modifying the functions, as the mathematical representation is independent of coordinate system. Moreover, the computation of cutter offsets and similar related curves for numerical control purposes can also be much simpler when parametric method is used (51).

### 3.3.3. Curve Design

Most objects designed in industry do not have simple shapes that can be defined by single analytic functions, thus it has been the normal practice to define curves in piecewise manner and surfaces in patches.

Although there are many non-parametric methods used in curve and surface design, the following Sections only deal with parametric piecewise description of curves and parametric patch design of surfaces.

#### 3.3.3.1. Ferguson Cubic Curve

The use of parametric representation of curves and surfaces first was introduced by Ferguson (40) at the Boeing Company for the design of aircraft body. Considering a cubic, the segments of these curves are described in the form of

$$r = r(u) = a_o + ua_1 + u^2a_2 + u^3a_3 \qquad (3.1)$$

One of the ways to determine the values of $a_o$, $a_1$, $a_2$, and $a_3$ is to find the value of r and dr/du at both ends of the segments. Taking the parameter u between 0 and 1, u = 0 and u = 1 refer to both ends of the segment.

Denoting ${}^{dr}/du$ by $\dot{r}(u)$

$$\begin{aligned} a_0 &= r(0), \\ a_1 &= \dot{r}(0), \\ a_0 + a_1 + a_2 + a_3 &= r(1), \\ a_1 + 2a_2 + a_3 &= \dot{r}(1) \end{aligned} \qquad (3.2)$$

solving for $a_0$, $a_1$, $a_2$, $a_3$,

$$\begin{aligned} a_0 &= r(0), \\ a_1 &= \dot{r}(0), \\ a_2 &= 3\left[r(1) - r(0)\right] - 2\dot{r}(0) - \dot{r}(1), \\ a_3 &= 2\left[r(0) - r(1)\right] + \dot{r}(0) + \dot{r}(1) \end{aligned} \qquad (3.3)$$

By substituting these values in (3.1), r can be described in terms of $r(0)$, $r(1)$, $\dot{r}(0)$, $\dot{r}(1)$, i.e.

$$\begin{aligned} r = r(u) &= r(0)(1-3u^2 + 2u^3) + r(1)(3u^2-2u^3) \\ &+ \dot{r}(0)(u - 2u^2 + u^3) + \dot{r}(1)(-u^2 + u^3) \end{aligned} \qquad (3.4)$$

or in matrix form

$$r(u) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} r(0) \\ r(1) \\ \dot{r}(0) \\ \dot{r}(1) \end{bmatrix} \qquad (3.5)$$

It must be noted that r (0), r (1), $\dot{r}$ (1), $\dot{r}$ (1) are all vectors. The derivatives $\dot{r}$ (0) and $\dot{r}$ (1) are therefore proportional to the unit tangent vectors T (0) and T (1) at the end points. Thus, we may write

$$\dot{r}(0) = \alpha_o T(0), \quad \dot{r}(1) = \alpha_1 T(1)$$

Care must be taken to ensure that the magnitude of the end tangent vectors do not become too large, as this will lead to unwanted kind of curve, Fig. (3.4). A safe rule is that the magnitudes should not be much greater than the chord length of the segment.

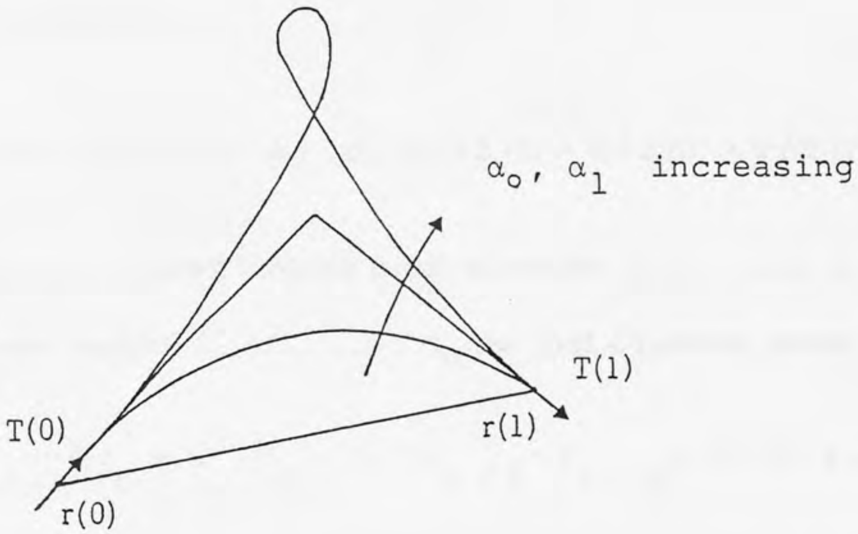


Fig. (3.4) The Effect of End Tangents on Ferguson's Curve

Suppose that we want to construct a composite curve by joining segment r1 (u1), $0 \leqslant u1 \leqslant 1$ to segment r 2 (u2), $0 \leqslant u2 \leqslant 1$. Normally the constraint is to have continuity at the joint and continuity of slope there. Therefore,

$$r1\,(1) = r2\,(0) \qquad (3.6)$$

$$\dot{r}1\,(1) = \alpha_1 T$$

$$\dot{r}2\,(0) = \alpha_2 T \qquad (3.7)$$

Ferguson in obtaining curvature continuity matches $r$, $\dot{r}$, $\ddot{r}$ across the joints, so that

$$\alpha_1 = \alpha_2 \qquad (3.8)$$

$$\ddot{r}\,1\,(1) = \ddot{r}\,2\,(0)$$

If $\ddot{r}\,1\,(1)$ and $\ddot{r}\,2\,(0)$ are evaluated from (3.4), conditions (3.6) and (3.8) yield that

$$6\,r1\,(0) + 2\,\dot{r}\,1\,(0) + 4\,\dot{r}\,1\,(1) = 6\,r2\,(1) - 4\,\dot{r}\,2\,(0) - 2\,\dot{r}\,2\,(1)$$

Now to fit a Ferguson curve through a set of points $r_o$, $r_1$, ... $r_n$ with tangents at those points $t_o$, $t_1$, ..... $t_n$, the last equation gives

$$t_{i-1} + 4\,t_i + t_{i+1} = 3\,(r_{i+1} - r_{i-1}),\ i = 1, 2, \ldots n-1$$

From this recursive relation it is only necessary to specify $t_o$ and $t_n$ to obtain the remaining tangents in terms of positional data using the following matrix.

$$\begin{bmatrix} 4 & 1 & 0 & \cdots & \cdots & 0 \\ 1 & 4 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 1 & 4 & 1 \\ 0 & \cdots & \cdots & 0 & 1 & 4 \end{bmatrix} * \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ \vdots \\ \vdots \\ t_{n-1} \end{bmatrix} = \begin{bmatrix} 3(r_2-r_o)-t_o \\ 3(r_3-r_1) \\ 3(r_4-r_2) \\ \vdots \\ \vdots \\ 3(r_n-r_{n-2})-t_n \end{bmatrix}$$

### 3.3.3.2. Spline Functions

A spline is a simple mechanical device; a long narrow strip of wood, metal, plastic or other elastic material used by loftsman to fair in a curve between specified points. Heavy objects called "ducks" are placed at the specified points and the spline is passed through so that a smooth or "fair" curve is produced.

Physical spline can be regarded to be a thin elastic beam then Euler's equation (52) yields

$$M(x) = \frac{EI}{R(x)}$$

where $M(x)$ : bending moment, $E$ : Young's module,

$R(x)$ : radius of curvature, $I$ : moment of inertia

If we denote by $y = y(x)$, the equation of the curvature produced by the spline

$$1/R = \ddot{y} / (1 + \dot{y}^2)^{3/2}$$

for small deflections small slopes, $\dot{y}$ can be neglected (53).

then $R \simeq {}^{1}/_{\ddot{y}} \quad \therefore \quad \ddot{y} = \frac{M(x)}{EI}$

Assuming that the ducks act as a simple support, then M(X) is a linear function between the supports. Taking M(X) = A X + B and integrating the resulting equation twice, shows that the physical spline is described by cubic polynomials between supports.

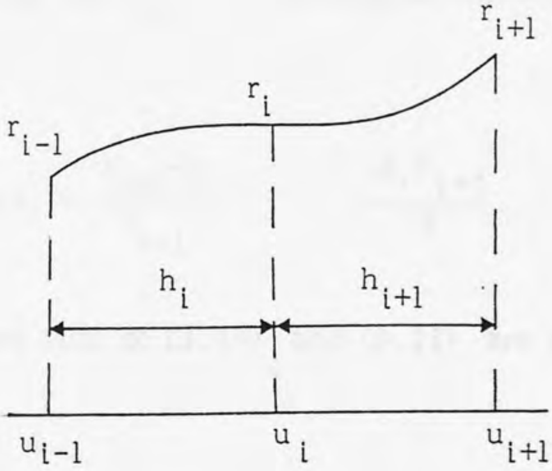
In general the mathematical spline is a piecewise polynomial of degree n with continuity of derivatives of order n - 1 at the joints.

There are many methods of constructing cubic splines. One of the more usual methods is as follows (54).

Taking $S(u)$ = cubic spline function $u_{i-1} < u < u_i$

since $S(u)$ is cubic $\ddot{S}(u)$ is linear in u over this span. If $\ddot{S}(u_{i-1}) = a_{i-1}$ and $\ddot{S}(u_i) = a_i$ then any point on $\ddot{S}(u)$ is calculated by

$$\ddot{S}(u) = \frac{a_{i-1}(u_i - u) + a_i(u - u_{i-1})}{h_i}$$



$h = u_i - u_{i-1}$

Fig. (3.5) Construction of Spline Curves

Integrating above equation twice

$$S(u) = \frac{a_{i-1}(u_i-u)^3 + a_i(u-u_{i-1})^3}{6h_i} + C_1 u + C_2$$

To determine the values of $C_1$ and $C_2$ the end conditions may be used which results in

$$S(u) = \frac{a_{i-1}(u_i-u)^3 + a_i(u-u_{i-1})^3}{6h_i}$$

$$+ \left[\frac{r_{i-1}}{h_i} - \frac{a_{i-1}h_i}{6}\right](u_i - u) + \left[\frac{r_i}{h_i} - \frac{a_i h_i}{6}\right](u-u_{i-1})$$

$$u_{i-1} \leqslant u \leqslant u_i \qquad (3.9)$$

To evaluate $a_{i-1}$ and $a_i$ the property of first order continuity at the joint is utilized. Differentiating (3.9) and taking $u=u_i$

$$\dot{S}(u_i) = \frac{r_i - r_{i-1}}{h_i} + \frac{a_i h_i}{3} + \frac{a_{i-1}h_i}{6} \qquad (3.10)$$

If in (3.9) $i$ is replaced by $i+1$ the function represents the next span $u_i \leqslant u \leqslant u_{i+1}$. Now differentiating this function and setting $u = u_i$.

$$\dot{S}(u_i) = \frac{r_{i+1}-r_i}{h_{i+1}} - \frac{a_i h_{i+1}}{3} - \frac{a_{i+1}h_{i+1}}{6} \qquad (3.11)$$

The right-hand side of (3.10) and (3.11) are equal then

$$h_i a_{i-1} + 2(h_i + h_{i+1}) a_i + h_{i+1} a_{i+1} =$$

$$\frac{6(r_{i+1} - r_i)}{h_{i+1}} - \frac{6(r_i - r_{i-1})}{h_i} \qquad (3.12)$$

Considering (3.12) over n spans of a spline curve built on $r_o, r_1, \ldots., r_n$, there will be n-1 functions and n + 1 unknowns, namely $a_o, a_1, \ldots.., a_n$, therefore,we need to know two additional relations for a. These additional relations depends upon the specific application (55, 56). Some possible relations are relaxed end (natural spline) ; i.e. $\ddot{S}(u_o) = \ddot{S}(u_n) = 0$, or clamped end, in which case the end tangents are specified.

### 3.3.3.3. B-Spline Curve

This is one of the most interesting mathematical methods introduced for computer aided design and was implemented in the system developed in this work.

The theory and the advantages of B-spline over other methods will be discussed later in this dissertation.

### 3.3.3.4. Bezier Curves

Curves such as Ferguson's that pass through all the specified data points are not very effective in interactive "ab-initio" design. This is because the control of the curve shape by numerical specification of both direction and magnitude of tangent derivatives does not provide the intuitive "feel" required for curve design; i.e. there is not always an obvious relation between the numbers and the shape of the curve.

The method of curve description devised by Bezier (43,57) allows the user much greater feel for the relation between input and output. This method enables the user to act as an artist, stylist or designer, varying curve shape until the satisfactory shape is achieved.

In this method the curve is defined by an open polygon. Only the first and last vertices of this polygon called "characteristic polygon" lie on the curve, however, the other points define the derivatives, order, and the shape of the curve. Local modification is easily achieved with Bezier curve. Any change in the vertices of a span will only affect the curve within that span.

The mathematical basis of Bezier curve is a polynomial blending function which interpolates between the first and the last vertices. Bezier curve is in fact defined by recombining of the terms in Ferguson parameterisation.

A cubic Bezier curve in simple form is defined as follows :

$$r = r\,(u) = (1-u)^3\, r_o + 3u\,(1-u)^2\, r_1 + 3u^2(1-u)r_2 + u^3 r_3$$

where $0 \leqslant u \leqslant 1$ for any given segment.

Compared to Ferguson's form we see that

$$a_o = r_o,$$

$$a_1 = 3\,(r_1 - r_o),$$

$$a_2 = 3\,(r_2 - 2r_1 + r_o),$$

and

$$a_3 = r_3 - 3r_2 + 3\,r_1 - r_o$$

Therefore,

$$r(0) = r_o\,,$$

$$r(1) = r_3, \qquad (3.13)$$

$$\dot{r}(0) = 3\,(r_1 - r_o),$$

and

$$\dot{r}(1) = 3(r_3 - r_2)$$

From (3.13) it can be seen that the curve passes through the points $r_o$, $r_3$ and has tangents at $r_o$ in the direction from $r_o$ to $r_1$ and at $r_3$ in the direction from $r_2$ to $r_3$.

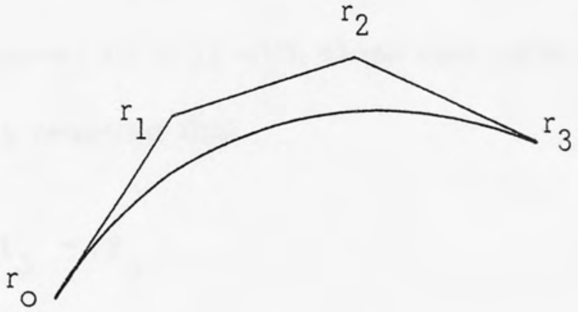


Fig. (3.6) A Bezier Curve Segment

The Bezier polynomial is related to the Bernstein polynomial. Thus, the Bezier curve is said to have Bernstein basis, and this basis function is given by :

$$B_{n,i}(u) = \frac{n!}{i!(n-i)!} u^{i} (1-u)^{n-i}$$

n is the degree of polynomial and i refers to the particular vertex from 0 to n. In general an n degree polynomial is defined by n + 1 vertices.

The general form of the Bezier curve is then defined by

$$r = r(u) = \sum_{i=o}^{n} \frac{n!}{i!\,(n-i)!} u^{i} (1-u)^{n-i} r_{i}$$

With the above definition curves of any degree can be constructed. Curves with higher order allow several orders of continuity, but as the order goes higher the characteristic polygon is less indicative of the shape of the curve.

If one wants to join a second curve, $r2(u2)$ to an already existing curve, $r1(u1)$ with slope and curavture continuity at the joint, it is firstly required that

$$r1_3 = r2_o \qquad (3.14)$$

since $r(0) = r_o$ and $r(1) = r_3$. It is also required that $\dot{r}1(1) = \dot{r}2(0)$.

From (3.13) $\dot{r}(0) = 3(r_1 - r_o)$ and $\dot{r}(1) = 3(r_3 - r_2)$

Therefore, $r1_3 - r1_2 = r2_1 - r2_o$ (3.15)

Remembering that all the parameters in (3.15) are vectors, (3.14) and (3.15) imply that $r1_2$, $r1_3 = r2_o$, and $r2_1$ must be collinear.

To achieve curvature continuity it is proved $^{(54)}$ that for a cubic $r1_1$, $r1_2$, $r1_3 = r2_o$, $r2_1$, and $r2_2$ must be coplanar.

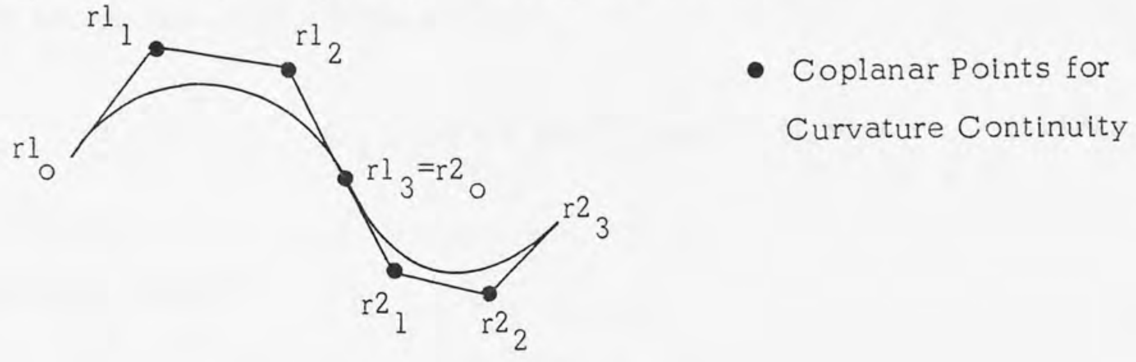


Fig. (3.7) Continuity of Bezier Curve Segments

### 3.3.4. Surface Design

### 3.3.4.1. Coons Patches

To understand Coons' $^{(42)}$ method of patch description we start by describing a lofted or ruled surface. Assume that two boundary curves associated with the opposite sides of the unit square in the u.v plane are known; say, $r(u,0)$, $r(u,1)$. A ruled surface is then obtained by linearly interpolating between these two curves. The linear interpolation scheme is

$$Q_1\ (u,v) = r\ (u,0)\ (1-v) + r\ (u,1)\ v \tag{3.16}$$

From the above method it is obvious that the edges of the interpolated surface coincide with the given data curves, i.e.,

$$Q_1 (u,0) = r (u,0)$$

$$Q_1 (u,1) = r (u,1)$$

Alternatively, if curves in the other two sides of the patches are known say, $r (0,v)$ and $r (1,v)$

$$Q_2 (u,v) = r (0,v) (1-u) + r (1,v) u \qquad (3.17)$$

again we obtain

$$Q_2 (0,v) = r (0,v)$$

$$Q_2 (1,v) = r (1,v)$$

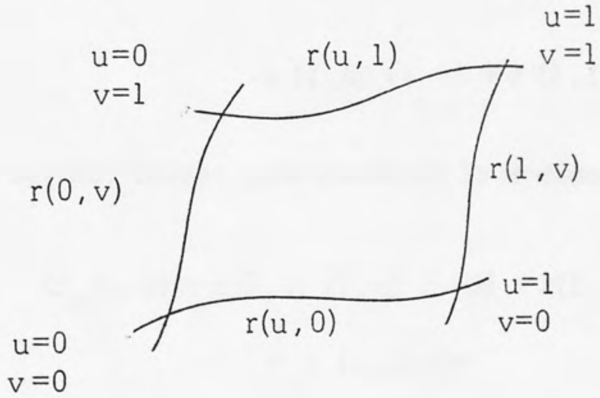


Fig. (3.8) A Parametric Surface Patch

A Coons patch is defined as simple sum of the lofted surfaces in the two directions.

$$Q = Q_1 + Q_2$$

$$Q (u,v) = r (u,0)(1-v) + r(u,1) v + r (0,v) (1-u) + r (1,v) u \quad (3.18)$$

However,(3.18) shows that at the corners

$$Q\ (0,0) = r\ (0,0) + r\ (0,0) \quad \text{etc.}$$

and at the edges

$$Q\ (0,v) = r\ (0,0)(1-v) + r\ (0,1)\, v + r\ (0,v) \qquad \text{etc.} \qquad (3.19)$$

Therefore, if one can find a surface patch $Q_3$ (u,v) whose boundaries are the unwanted linear interpolants the patch can be defined by forming $Q_1 + Q_2 - Q_3$. From (3.19) it is obvious that the unwanted boundary curve for u = 0 is

$$r\ (0,0)\ (1-v) + r\ (0,1)\ v$$

and for u = 1

$$r\ (1,0)\ (1-v) + r\ (1,1)\ v$$

and a further linear interpolation in u direction then gives

$$Q_3(u,v) = r(0,0)\ (1-u)\ (1-v) + r(1,0)\ u\ (1-v) + r\ (0,1)(1-u)v + r\ (1,1)\ uv \qquad (3.20)$$

The surface Q is then obtained by $Q_1 + Q_2 - Q_3$ which in matrix form is expressed as below :

$$Q(u,v) = \begin{bmatrix} 1-u & u & 1 \end{bmatrix} \begin{bmatrix} -r\,(0,0) & -r\,(0,1) & r\,(0,v) \\ -r\,(1,0) & -r\,(1,1) & r\,(1,v) \\ r\,(u,0) & r\,(u,1) & 0 \end{bmatrix} \begin{bmatrix} 1-v \\ v \\ 1 \end{bmatrix}$$

(3.21)

Linear Coons patch as in (3.21) is the most elementary of a class of surfaces introduced by Coons.

The auxiliary functions u, 1-u, v, and 1-v are called blending functions because they blend the boundary curves to produce the internal shape of the surface. These linear blending functions can be replaced by other functions. Suppose we replace 1-u by $A_o(u)$ and u by $A_1(u)$. These blending functions then should satisfy the conditions

$$1 - A_1 = A_o$$

$$A_o(0) = 1, \qquad A_o(1) = 0, \qquad A_1(0) = 0,$$

and $\quad A_1(1) = 1$

Usually blending functions are chosen to be continuous, therefore, polynomials are widely used.

Now if it is intended to construct a surface made up of the type of patches described, only positional continuity can be achieved on the boundaries. However, in most practical cases slope continuity is essential and one of the methods to achieve this is to impose additional conditions on the blending functions such as

$$\dot{A}_o(0) = \dot{A}_o(1) = \dot{A}_1(0) = \dot{A}_1(1) = 0$$

By doing so the slope across the boundary, say r (u,0) takes the form

$$Q(u,0)_v = r_v(0,0)\ A_o(u) + r_v(1,0)\ A_1(u)$$

Here the subscript refers to slope, i.e. $Q(u, 0)_u = \dfrac{\partial (Q(u, 0))}{\partial u}$

Therefore, the slope across the boundaries depends on the end tangent vectors across the boundary, and the blending function. Two patches with the same blending functions and common boundary curve, i.e. $r1(u, 1) = r2(u,0)$ are continuous in slope if $r1_u(0,1) = k.\ r2_u(0,0)$ and $r1_v(1,1) = k.\ r2_v(1,0)$.

In general, a patch is the sum of one or more types of Coons patches (58).

$$P(u, v) = Q(u,v) + R(u,v) + S(u,v) + \ldots.$$

$Q(u,v)$ described earlier according to Forrest is called the first cononical form, and other types can be regarded as correction surfaces which alter the internal shape and boundary conditions of the basic patch $Q(u,v)$.

Using this approach, first and second derivatives continuity across the boundaries can effectively be implemented. For slope continuity therefore we have $P(u,v) = Q(u,v) + R(u,v)$. The function $R(u,v)$ will change the slope and higher order boundary conditions without changing the shape of the boundary curves.

Similar to (3.21), $R(u,v)$ is defined as below :

$$R(u,v) = \begin{bmatrix} B_o(u) & B_1(u) & 1 \end{bmatrix} \begin{bmatrix} -r_{uv}(0,0) & -r_{uv}(0,1) & r_u(0,v) \\ -r_{uv}(1,0) & -r_{uv}(1,1) & r_u(1,v) \\ r_v(u,0) & r_v(u,1) & 0 \end{bmatrix} \begin{bmatrix} B_o(v) \\ B_1(v) \\ 1 \end{bmatrix} \tag{3.22}$$

$B_o$ and $B_1$ are the slope blending functions and to ensure that the correction surface has zero positional value on the boundaries, but has required cross boundary slope, they should satisfy the following conditions :

$$B_o(0) = B_o(1) = B_1(0) = B_1(1) = 0$$

and $$\dot{B}_o(0) = 1 \;,\; \dot{B}_o(1) = 0 \,,\; \dot{B}_1(0) = 0 \,,\; \dot{B}_1(1) = 1$$

Finally, (3.21) and (3.22) can be combined in a single matrix form for a more compact representation.

$$P(u,v) = Q(u,v) + R(u,v) = - \begin{bmatrix} -1 & A_o(u) & A_1(u) & B_o(u) & B_1(u) \end{bmatrix} M \begin{bmatrix} -1 \\ A_o(v) \\ A_1(v) \\ B_o(v) \\ B_1(v) \end{bmatrix}$$

$$\text{Where } M = \begin{bmatrix} 0 & r(u,0) & r(u,1) & r_v(u,0) & r_v(u,1) \\ r(0,v) & r(0,0) & r(0,1) & r_v(0,0) & r_v(0,1) \\ r(1,v) & r(1,0) & r(1,1) & r_v(1,0) & r_v(1,1) \\ r_u(0,v) & r_u(0,0) & r_u(0,1) & r_{uv}(0,0) & r_{uv}(0,1) \\ r_u(1,v) & r_u(1,0) & r_u(1,1) & r_{uv}(1,0) & r_{uv}(1,1) \end{bmatrix} \tag{3.23}$$

The most important feature of the Coons method is its extreme generality. In addition to surface design it can also be used in schemes for surface fitting. Two families of curves can be fitted to the data points which can then be used as patch boundaries.

There are several disadvantages when using this general form of surface definition in computer aided design systems. The major disadvantage exist because three different quantities, position, tangent and twist vectors must be specified. Corner twist vectors are normally difficult to comprehend when using this method. Therefore, sometimes they are assumed to be zero. This, however, causes flatness (59) near to the patch corners of the composite surface. In order to overcome this difficulty Forrest (58) suggests that twist vectors should be computed in terms of other data values.

Finally, it must be noted that the user lacks intuition when creating a surface by changing elements in the defining matrix, and cannot predict the effect of the change.

### 3.3.4.2. Ferguson Surface

A Ferguson curve as has been discussed can be shown in the form :

$$r = r(u) = r(0)\, A_0(u) + r(1) A_1(u) + \dot{r}(0)\, B_0(u) + \dot{r}(1)\, B_1(u) \qquad (3.24)$$

where
$$\begin{aligned} A_o(u) &= 1 - 3u^2 + 2u^3 \\ A_1(u) &= 3u^2 - 2u^3 \\ B_o(u) &= u - 2u^2 + u^3 \\ B_1(u) &= -u^2 + u^3 \end{aligned} \tag{3.25}$$

or one may write functions in (3.25) in matrix form as :

$$F(u) = \begin{bmatrix} A_o(u) & A_1(u) & B_o(u) & B_1(u) \end{bmatrix} = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix}$$

$$= UN \tag{3.26}$$

In Ferguson surface the blending functions are taken to be the same as the elements used in the boundary curves. If the same procedure used for constructing Coons patches, that is, the sum of two lofted surfaces subtracted by extra elements to satisfy the boundary conditions is used, the final result will be ;

$$P(u,v) = \begin{bmatrix} A_o(u) & A_1(u) & B_o(u) & B_1(u) \end{bmatrix} \begin{bmatrix} r(0,0) & r(0,1) & r_v(0,0) & r_v(0,1) \\ r(1,0) & r(1,1) & r_v(1,0) & r_v(1,1) \\ r_u(0,0) & r_u(0,1) & r_{uv}(0,0) & r_{uv}(0,1) \\ r_u(1,0) & r_u(1,1) & r_{uv}(1,0) & r_{uv}(1,1) \end{bmatrix} \begin{bmatrix} A_o(v) \\ A_1(v) \\ B_o(v) \\ B_1(v) \end{bmatrix} \tag{3.27}$$

taking

$$\begin{bmatrix} A_o(v) \\ A_1(v) \\ B_o(v) \\ B_1(v) \end{bmatrix} = N^T \left[ 1 \quad v \quad v^2 \quad v^3 \right]^T \quad *$$

(3.27) can more compactly be presented as

$$P(u,v) = \left[1 \ u \ u^2 \ u^3\right] N \ R1 \ N^T \left[1 \ v \ v^2 \ v^3\right]^T = U \ N \ R1 \ N^T V^T$$

where N is the square matrix in (3.26) and R1 the square matrix in (3.27).

Ferguson surface is in fact a particular case of Coons method in that boundary curves are restricted to polynomials that contain the same elements as the blending functions. Recalling previous section, in order to eliminate some of the problems, a simplified version originally due to Ferguson (40) called an F-patch is sometimes used in which twist vectors are assumed to be zero, therefore, only first order continuity is possible.

### 3.3.4.2.1. APT Surface Fitting Routine

One practical application of F-patch is in F-MILL (60) which is used in APT for surface fitting. In F-patch (54) twist vectors are taken to be zero and gradients $r_u$ and $r_v$ at mesh intersections are

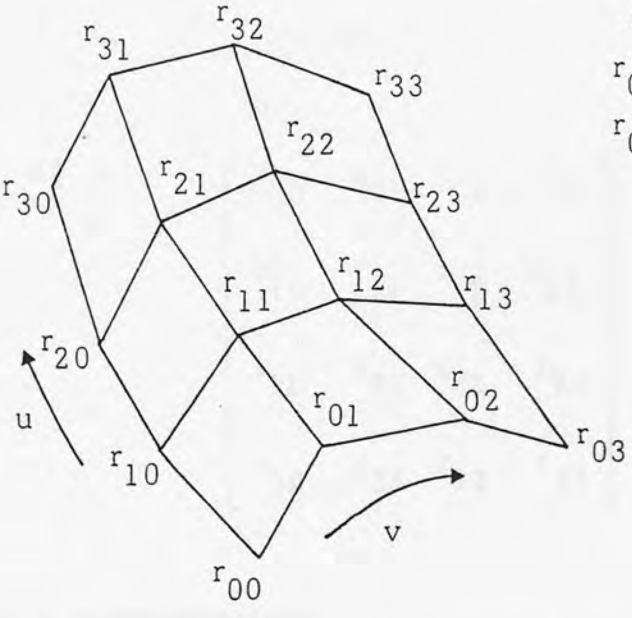
---

* $N^T$ means Transpose of matrix N

calculated using positional data. Then directions are taken as parallel to the chord line joining the preceding and the following data points and the magnitude is taken to be the length of the chord joining the point concerned to either the preceding or the following point. To avoid the problem of loops (refer to 3.3.3.1) in the curve in case of a high magnitude for the tangent, the smaller value is chosen.

### 3.3.4.3. Bezier Surfaces

The Bezier surface like its corresponding curve design is based on approximation rather than fitting of surface to a set of given data points. A Bezier patch is designed in terms of a "characteristic polyhedron". A cubic patch is defined by its 16 vertices. The patch is in fact an approximation to the polyhedron and only eight points of the polyhedron lie on the surface.



Vertices lying on the surface are :

$r_{00}$ $r_{10}$ $r_{20}$ $r_{30}$

$r_{03}$ $r_{13}$ $r_{23}$ $r_{33}$

Fig. (3.9) The Characteristic Polyhedron for a Cubic Bezier Patch

In most cases of surface description it is assumed that information such as position, tangent and twist vectors are available. However, providing this information may be extremely difficult. In addition, these schemes do not provide the user with sufficient "feel" in indicating the effect of changes in the defining input on the shape of the curve, the Bezier surface description is one of the most successful methods devised to overcome these difficulties. No gradient or twist vectors need to be specified, and the configuration of the polyhedron gives the user a good indication of the general shape of the patch.

Bezier patch using the binomial representation given in Section 3.3.3.2. for Bezier curve can be represented in the form (57);

$$P\,(u,v) = \left[(1-u)^3 \quad 3u\cdot(1-u)^2 \quad 3u^2\cdot(1-u) \quad u^3\right] R2 \begin{bmatrix} (1-v)^3 \\ 3v\,(1-v)^2 \\ 3v^2(1-v) \\ v^3 \end{bmatrix}$$

$$\text{Where } R2 = \begin{bmatrix} r_{00} & r_{01} & r_{02} & r_{03} \\ r_{10} & r_{11} & r_{12} & r_{13} \\ r_{20} & r_{21} & r_{22} & r_{23} \\ r_{30} & r_{31} & r_{32} & r_{33} \end{bmatrix} \qquad (2.28)$$

In matrix representation

$$\begin{bmatrix}(1-u)^3 & 3u(1-u)^2 & 3u^2(1-u) & u^3\end{bmatrix} = \begin{bmatrix}1 & u & u^2 & u^3\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ -3 & 3 & 0 & 0\\ 3 & -6 & 3 & 0\\ -1 & -3 & 3 & 1\end{bmatrix}$$

$$= U\ M \qquad (3.29)$$

Therefore,

$$P(u,v) = U\ M\ R2\ M^T \begin{bmatrix}1 & v & v^2 & v^3\end{bmatrix}^T = U\ M\ R2\ M^T\ V^T$$

Recalling 3.3.4.2. a Ferguson surface can be shown as

$$P(u,v) = U\ N\ R1\ N^T\ V^T$$

Comparing this with Bezier's

$$U\ N\ R1\ N^T\ V^T = U\ M\ R2\ M^T\ V^T$$

Hence $\quad R1 = (N^{-1}\ M)\ R2\ (N^{-1}\ M)^T \quad *$

or $\quad R1 = \begin{bmatrix} r(0,0) & r(0,1) & r_v(0,0) & r_v(0,1)\\ r(1,0) & r(1,1) & r_v(1,0) & r_v(1,1)\\ r_u(0,0) & r_u(0,1) & r_{uv}(0,0) & r_{uv}(0,1)\\ r_u(1,0) & r_u(1,1) & r_{uv}(1,0) & r_{uv}(1,1)\end{bmatrix}$

$$= \begin{bmatrix} r_{00} & r_{03} & 3(r_{01}-r_{00}) & 3(r_{03}-r_{02})\\ r_{30} & r_{33} & 3(r_{31}-r_{30}) & 3(r_{33}-r_{32})\\ 3(r_{10}-r_{00}) & 3(r_{13}-r_{03}) & 9(r_{00}-r_{10}-r_{01}+r_{11}) & 9(r_{02}-r_{12}-r_{03}+r_{13})\\ 3(r_{30}-r_{20}) & 3(r_{33}-r_{23}) & 9(r_{20}-r_{30}-r_{21}+r_{31}) & 9(r_{22}-r_{32}-r_{23}+r_{33})\end{bmatrix}$$

---

* $N^{-1}$ means Inverse of matrix N

It can be seen that gradient and twist vectors are expressed in terms of the vertices of the characteristic polyhedron. Therefore, unsatisfactory assumptions such as $r_{uv} = 0$ are avoided.

To achieve continuity on the border of two adjacent patches of a composite surface, let's take the patches as $r1(u,v)$ and $r2(u,v)$.

Positional continuity across the boundary will result if $r1(1,v) = r2(0,v)$ for all values of $v$ such that $0 \leqslant v \leqslant 1$. This is achieved when the two patches have common boundary polygon between the two characteristic polyhedrons, i.e.

$$r1_{3i} = r2_{0i} \qquad i = 0, 1, 2, 3$$

Slope continuity can be achieved in two different ways. First by collinearity of four paris of polyhedron edges that meet at the boundary, Fig. (3.10.a), and secondly, by coplanarity of three edges as shown in Fig. (3.10.b).

If higher degrees of continuity is required, cartesian product representation $^{(54, 61)}$ of Bezier surface should be used. In particular, the surface is given by;

$$P(u,v) = \sum_{i=o}^{n} \sum_{j=o}^{m} B_{n,i}(u)\, B_{m,i}(v)\, r_{ij}$$

Where $B_{n,i}(u)$ and $B_{m,j}(v)$ are Bernstein basis functions of degree n and m.

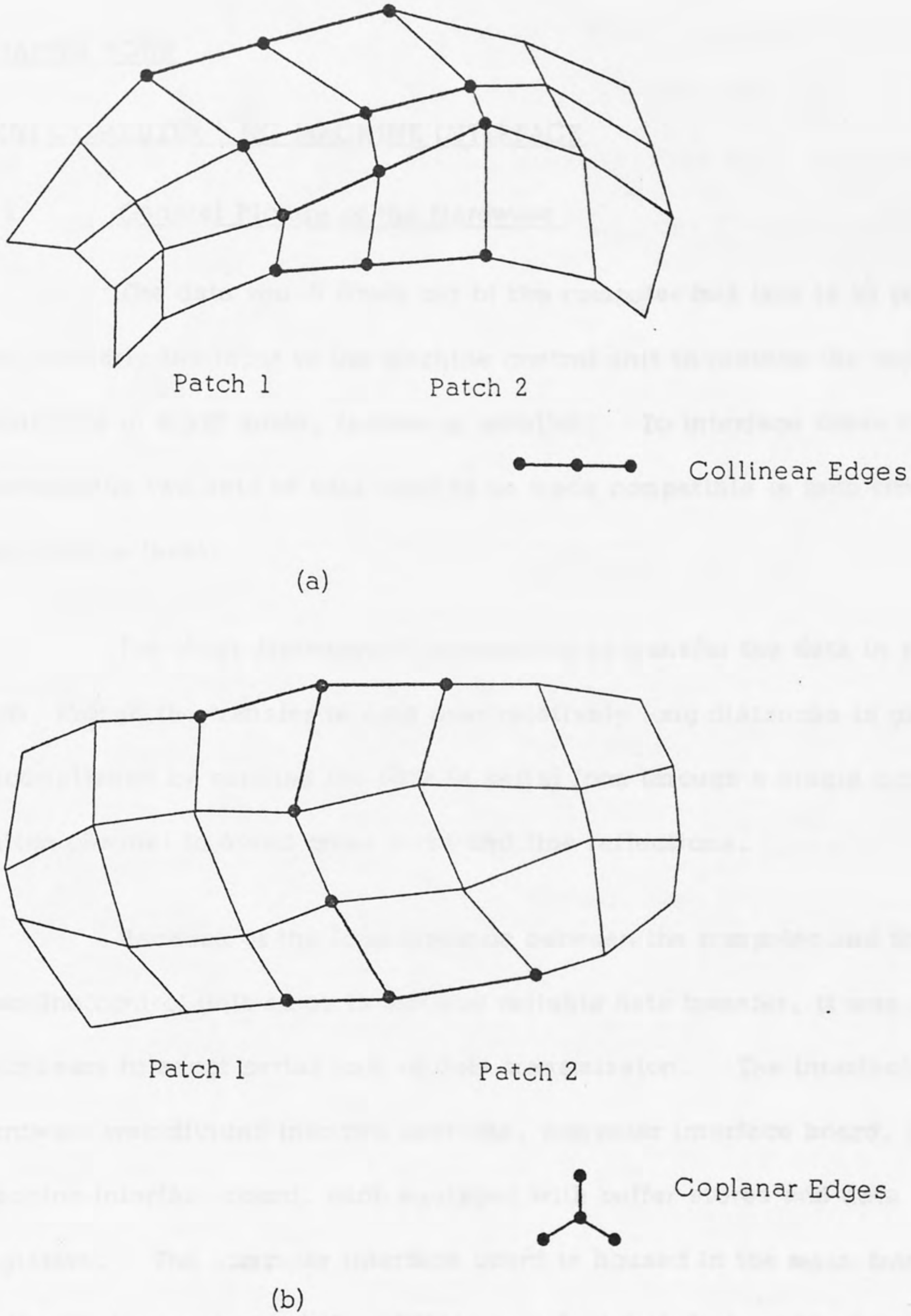


Fig. (3.10) Characteristic Polyhedra for Construction of Composite Surfaces with Positional and Slope Continuity

# CHAPTER FOUR

# MINI COMPUTER - NC MACHINE INTERFACE

## 4.1 General Picture of the Hardware

The data which flows out of the computer bus line is in parallel and similarly the input to the machine control unit to replace the tape input data in a BTR mode, is also in parallel. To interface these two systems the two sets of data need to be made compatible in both timing and voltage level.

For short distances it is possible to transfer the data in parallel form, though the transfer of data over relatively long distances is generally accomplished by sending the data in serial form through a single communication channel to avoid cross talks and line reflections.

Because of the long distance between the computer and the machine control unit so as to achieve reliable data transfer, it was necessary to adopt serial form of data transmission. The interfacing hardware was divided into two sections, computer interface board, and machine interface board, each equipped with buffer stores and data registers. The computer interface board is housed in the main frame of the mini computer and the machine interface board placed in the machine control console. Communication between the two parts is through serial asynchronous data transmission lines. The block diagram of the interface is shown in Fig. (4.1).

Parallel data out of the computer output registers is passed to the buffer store and then shifted to the transmitter when requested. The request is signalled when the last memory in the buffer store on the machine interface board is half full. The received data on the machine interface board is passed to the buffer store and then transferred to the machine control unit on receiving a pulse which would normally operate the stepping motor of the tape reader for feeding the tape under the reader.

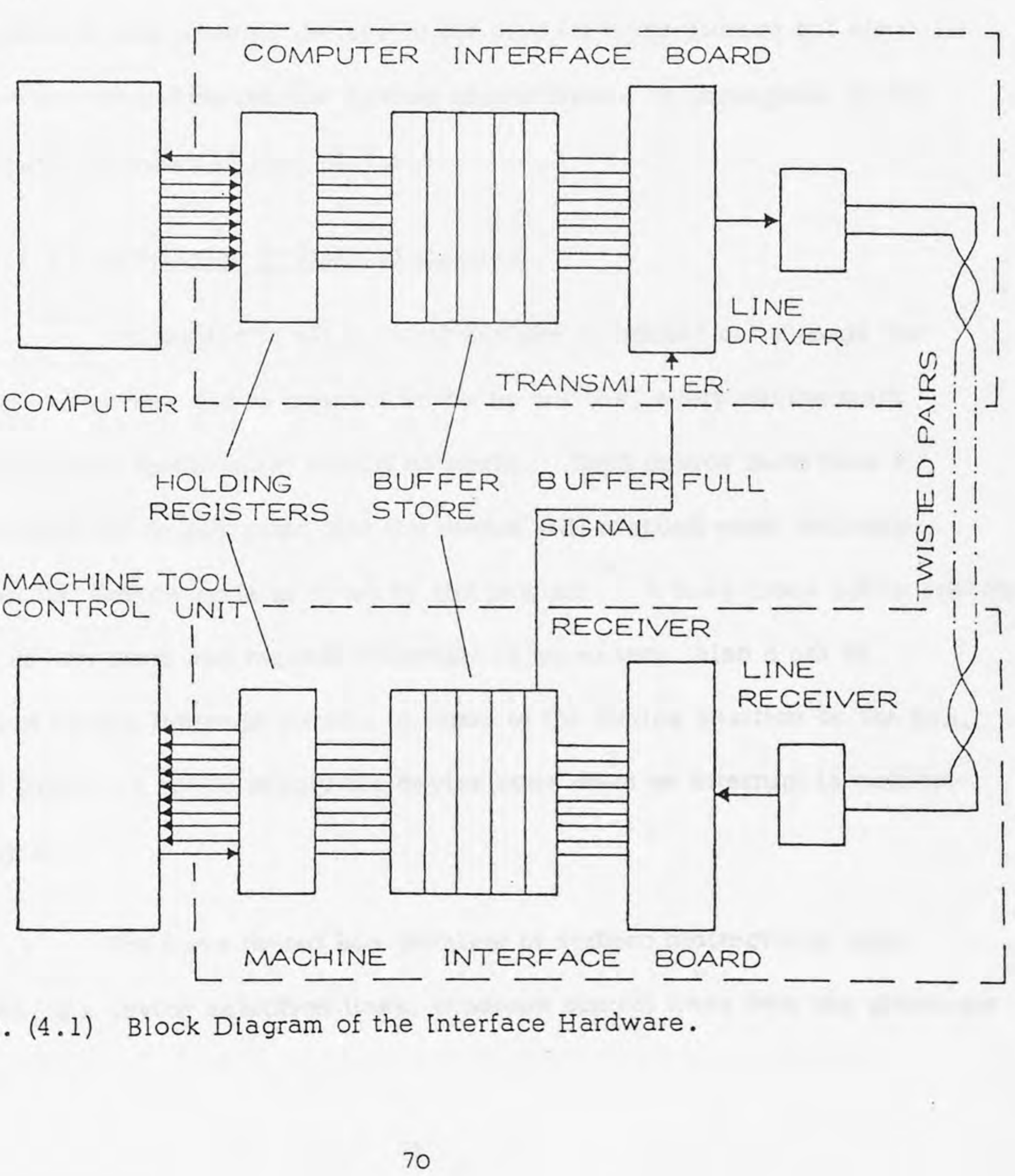


Fig. (4.1) Block Diagram of the Interface Hardware.

### 4.1.1. The Data General 'NOVA 1220' Mini Computer

The computer used is a general purpose mini computer made by Data General Corporation. The word length is 16-bit and the present memory capacity, 16,384 words.

The input-output hardware allows the software to address up to sixty-two devices. The unit as it stands can run under Stand-alone Operating System and Disc Operating System. The Disc Operating System not only enables the use of the disc for mass storage but also provides comprehensive file system capabilities. A photograph of the computer is shown in Fig. (4.3.a.).

#### 4.1.1.1. Interfacing to External Devices

Interfacing to all external devices is carried out through the input-output bus and to connect to the in-out bus, every device must have certain fundamental circuit networks. Each device must have a selection net to guarantee that the device will respond when and only when its device code is given by the program. A Busy-Done net to specify the device state and request interrupts is necessary, also a net to determine the interrupt priority in terms of the device position on the bus, and finally, a net to supply the device code when an interrupt is acknowledged.

The Nova in-out bus consists of sixteen bidirectional data lines, six device selection lines, nineteen control lines from the processor to device and six control lines from device to the processor. Signals on the control lines from the processor synchronise all the transfers on the data lines, start and stop device operations, and control program interrupts and data channels. Over the control lines to the processor a device can indicate the states of its Busy-Done flag and request program interrupt.

### 4.1.1.1.1. Programmed Data Transfer

Three classes of operations take place over the in-out bus(62) namely, programmed transfer, events associated with requesting and acknowledging program interrupts, and data channel transfers. Here only the programmed transfer is discussed.

Once a device is launched into performing some operation, either by a signal from the processor or by external stimulus, the device number is placed on device selection lines DS (0-5). The 6 bits makes it possible to select one out of 64 different devices. The data to be sent to the device is placed on the data lines (0-15), or if a device is supplying input to the processor this will appear on the data lines. To actuate the transfer a signal, "Data Out" (A, B or C) is placed on the appropriate line, or in the case of input, "Data In" (A, B,C) is supplied. These signals gate the appropriate registers of the selected device to or from data lines.

Following the transfer the processor generates control pulses to set the condition of Busy-Done flip flops. The timing diagram for programmed input-output transfer is shown in Fig. (4.2).

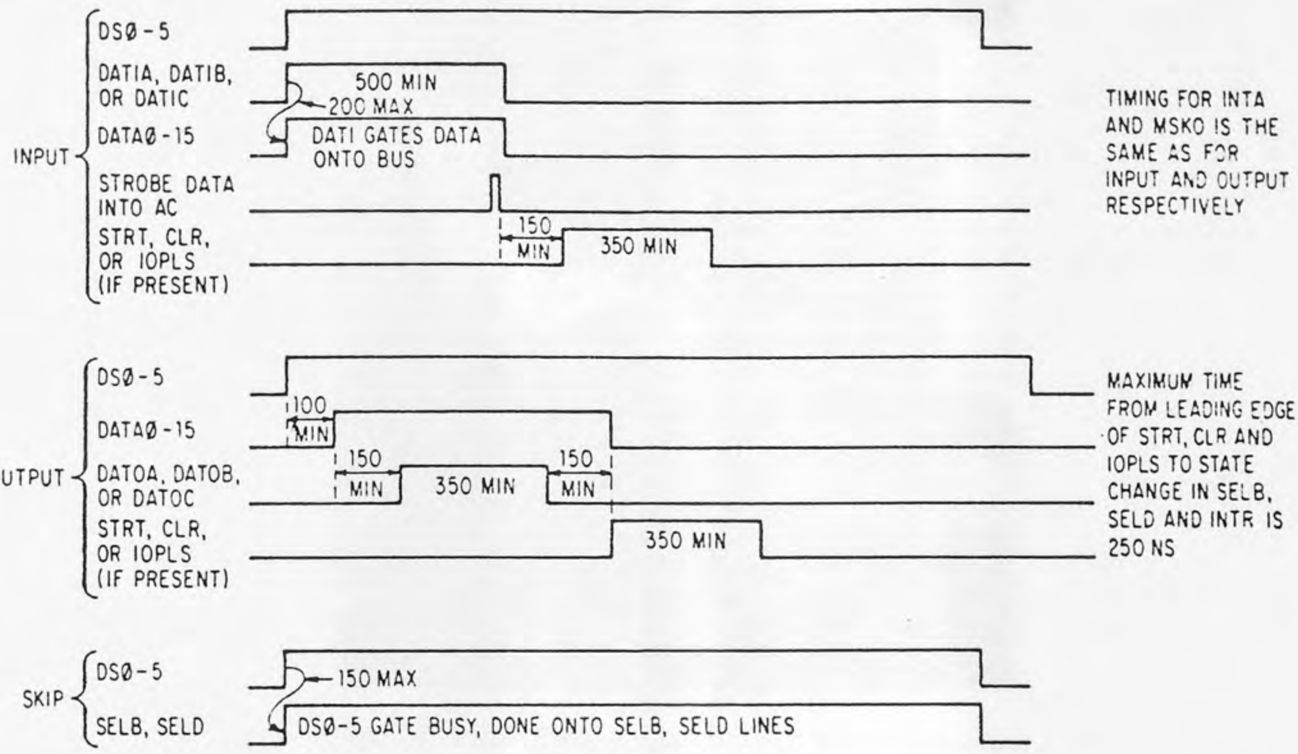


Fig. (4.2) Timing Diagram for Programmed Data Transfer (62)

### 4.1.2. Cincinnati Cintimatic 3VT-1000 Machining Centre - Acramatic Series 5 Control Unit

The machine tool in this work is a Cincinnati 3VT-1000 machining centre made by Cincinnati-Milarcon. The machine is fitted with a vertically moving turret that carries eight tools any one of which may be selected at random. A photograph of the machine is given in Fig. (4.3.b ).

Fig. (4.3.a) Data General 1220 Mini Computer

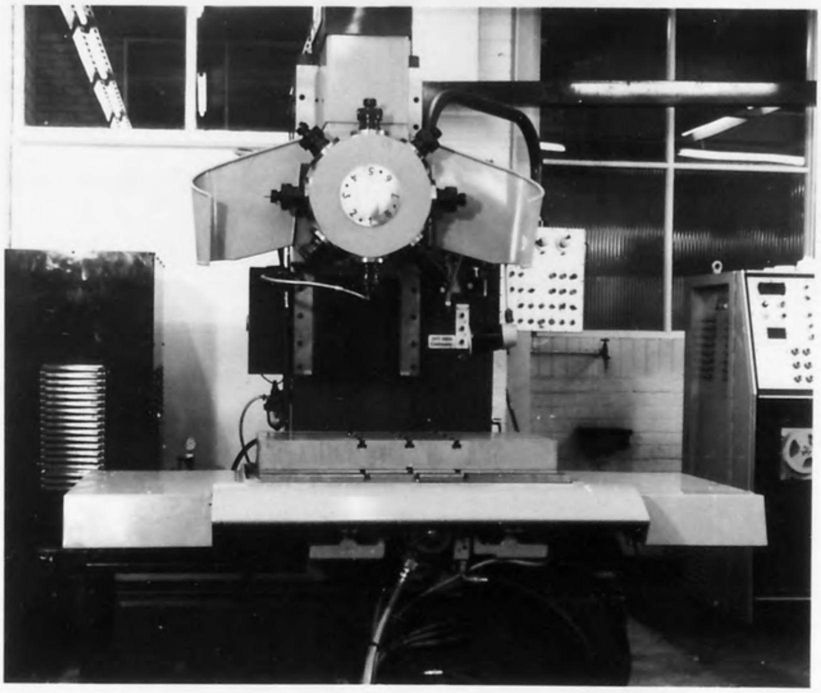


Fig. (4.3.b) Cincinnati 3VT-1000 Machining Centre

The machine is equipped with Acramatic series 5 control unit. It is a three axis positioning, two axis contouring control system ($2\frac{1}{2}$D). Logic construction is based on TTL integrated circuit components. The operating features of the system include:

a. Linear and circular interpolation

A straight line can be machined between two positions at a constant feed rate in X and Y simultaneously, thus providing any desired angle for machining. Circular interpolation in X-Y plane is also possible for cutting a variety of circular arcs.

b. Canned cycles

The system offers ten standard machining cycles, each of which can be called up from the tape. They include, milling, dwell, tap, cancel, drill, and various boring cycles.

c. Tool length compensation

With this facility the need for pre-setting of tool is eliminated by utilizing the panel on the side of the control. The tool tip can be positioned at a set point and panel switches altered to store the datum height.

Other features include, manual data input, switchable inch/metric operation, switchable EIA/ISO tape character code, and tape search facility.

### 4.1.2.1. Acraread Tape Reader

Data input to the control system is via an Acraread photo-electronic tape reader. The function of the tape reader is to drive the perforated tape under the reading head while converting the holes sensed into electrical signals.

### 4.1.2.1.1. Tape Transport

The drive system consists of two sprocket wheeles driven by a three - phase star-wound stepping motor via a toothed belt. One step of this motor will feed the tape in either directions a distance of one sprocket pitch.

The stepping motor is controlled by a reversible three bit ring counter which is clocked by the "MOD 32 K" signal from the Acramatic 5 control unit. This produces 156.25 motor steps per second.

### 4.1.2.1.2. Tape Perforation Sensing

A nine cell photo-transistor array in the reading head is used to sense the perforation in the tape. A remotely mounted light source with a fibre-optic array system provides continuous beam of light for the photo-transistors.

The output signals from the photo-transistors are then passed to the "Trimmer Card". Each input to the Trimmer Card is amplified by individual emitter follow circuits. The output signals from the Trimmer Card pass to the "Photo-cell Interface" board Fig. (4.4), where

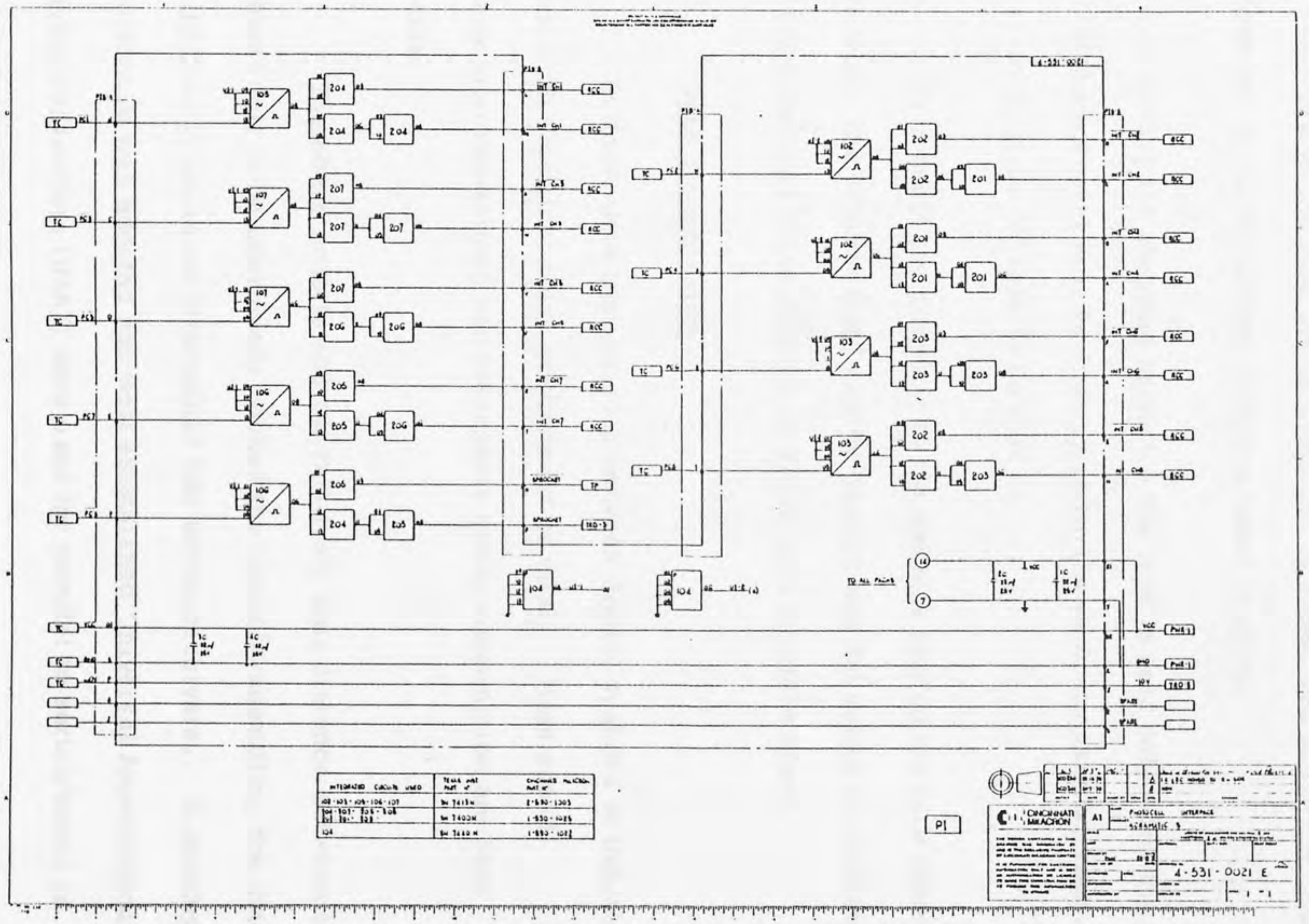


Fig. (4.4 ) Circuit Diagram of the Photo-cell Interface Board.

they are "squared up" and inverted by Schmidt trigger circuits to make them compatible with the TTL circuits used in the control. In Fig. (4.5.a) the timing diagram for tape reader operation is shown and in Fig. (4.5.b) waveform out of the Photo-cell Interface board is given.

During the stepping operation the tape is read only when the centre of the sprocket hole is directly under the photo-transistor, then permission is given for tape to be read in.

To connect the computer to the machine behind the tape reader in this work, the Trimmer Card was by-passed and the output of machine interface board was taken directly to Photo-cell Interface board.

### 4.1.3. Data Transmission

Reliable data transmission between digital systems in industrial and high electrical noise environment is vital. High speed, minimum interconnections and reasonable power consumption are also desirable.

The above criteria and the relatively long distance between the peripheral unit to be interfaced, backed the idea of transmitting the data serially through balanced differential line driver/receivers. A synchronous technique was adopted and well established Universal Asynchronous Receiver/Transmitters (UART) were used for parallel to serial/serial to parallel conversions.

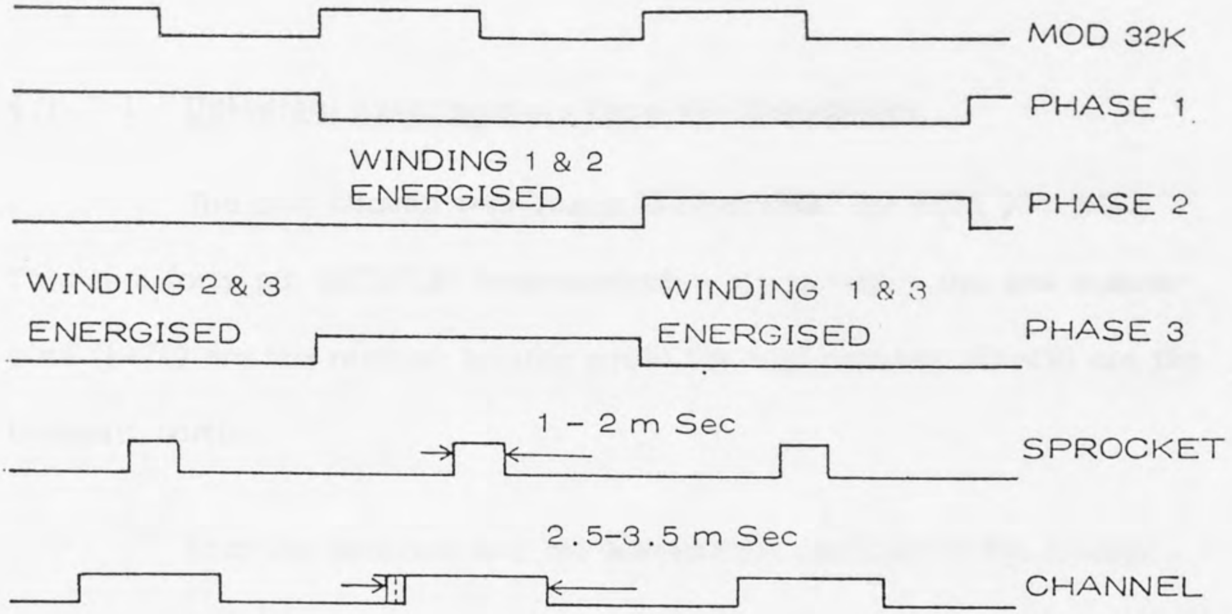


Fig. (4.5.a ) Timing Diagram for Tape Reader Operation

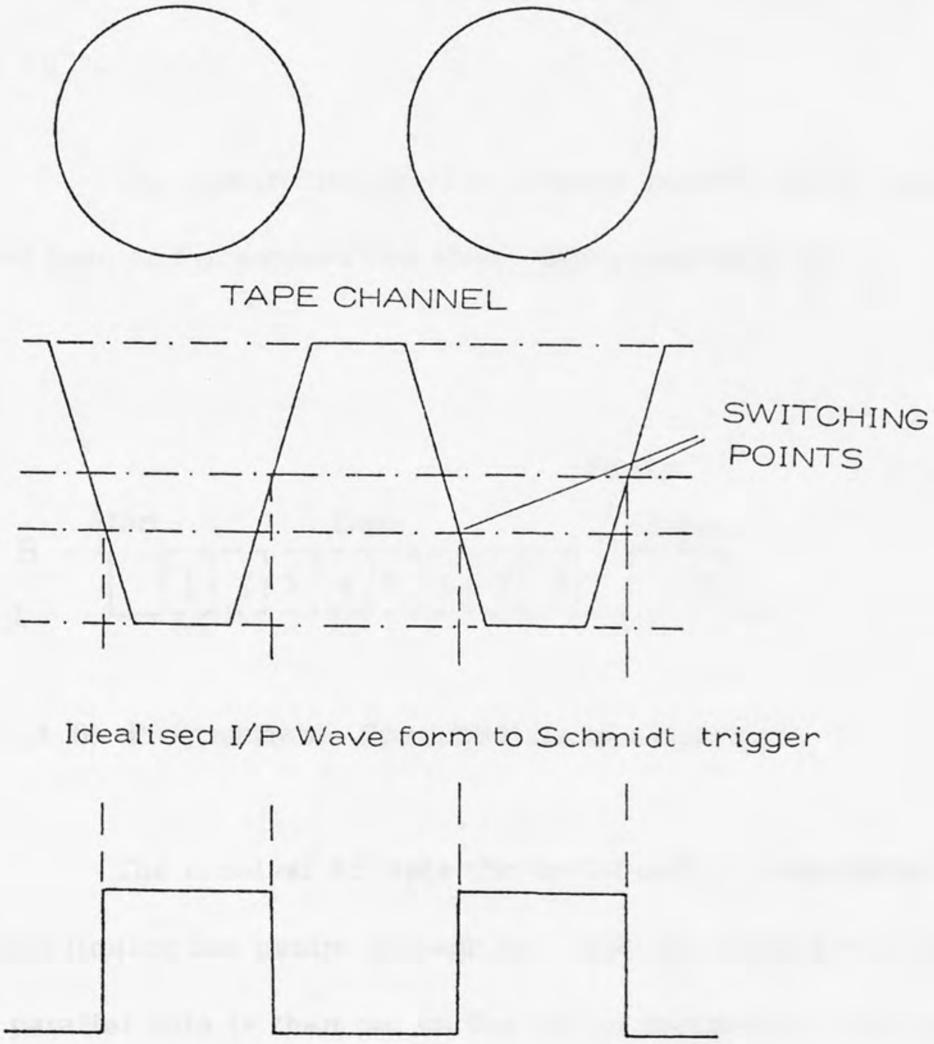


Fig. (4.5.b ) Waveform Out of Photo-cell Interface Board

### 4.1.3.1. Universal Asynchronous Receiver Transmitter

The unit chosen was Texas Instruments' TM 6011 JC, NC. This is a forty pin MOS/LSI programmable chip in which the low number pins (1-20) are the receive portion while the high number (21-40) are the transmit portion.

Both the receiver and the transmitter portions of the circuit need a clock having a frequency equal to 16 times the baud rate. This high frequency enables UART to do such things as sample the centre of each data interval and recheck for valid start signals. The crystal clocks to run the system were set at 200 KHZ, giving a baud rate of $200/16 * 10^3 \simeq 12500$.

The transmitter section accepts parallel data, converts them to serial form and generates the start, parity and stop bit .

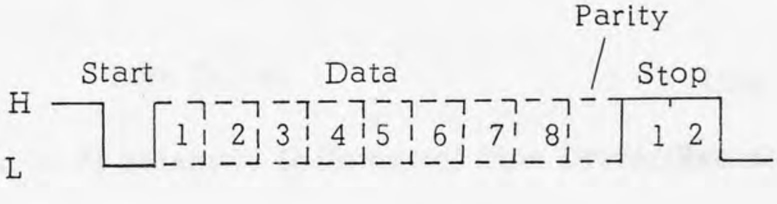


Fig. (4.6) Programmable Specification of Characters

The receiver accepts the serial data by searching for the start bit and finding the centre of each bit, then converting it to parallel form. The parallel data is then put on the output registers to flow through the rest of the circuit.

### 4.1.3.2. Line Drivers and Line Receivers

Advanced Micro Devices' Am 9714 and Am 9615 were selected as line driver and line receiver respectively. These work on a differential basis, therefore, in industrial environment electrical noise will affect both lines but not the difference.

The response time of each receiver and thereby immunity to AC noise can be controlled by an external capacitor. Line matching can be achieved in a number of ways, such as using a terminating resistor to connect the input terminals of the receiver, or by including serial resistors to the transmission line at the drive end.

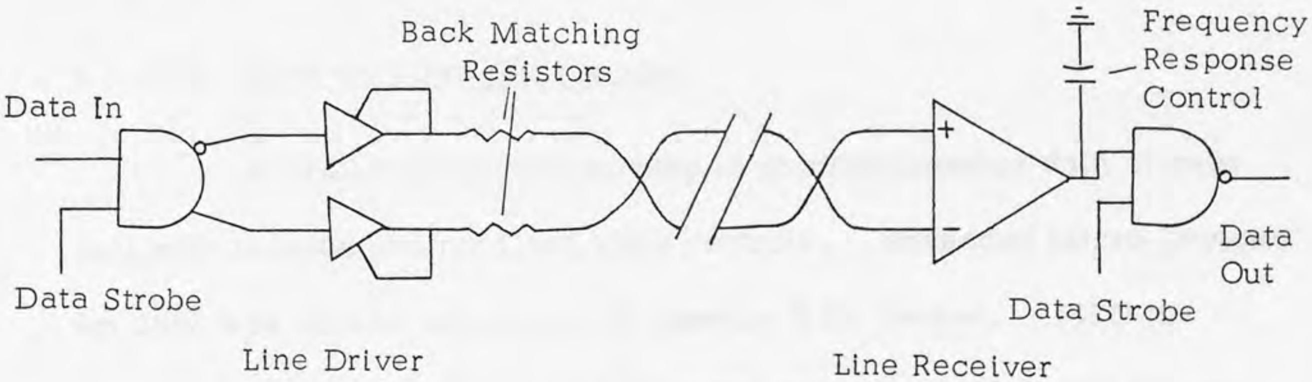


Fig. (4.7) Balanced Differential Line Driver/Receiver.

### 4.1.4. Buffer Store

A dual buffer storage technique was used in the interface. The buffer store on the machine interface board ensures that there is always data available for the machine and any delay which could cause a broken cutter or dwell is avoided. This is more important if the older generation of machine tools are incorporated into the system as most of them do not have any internal memory buffer. Using such machines, one of the factors affecting the surface finish is the speed at which the data is input.

Another advantage associated with this technique is that the buffer area on the computer interface board can be filled with a block of information consequtively instead of by a character at a time as the characters are transmitted. This will provide the central processor of the computer with longer intervals of time in servicing other peripherals and processing the control routines and functions. Consequently, larger number of machines could be added into a DNC cell controlled by a single computer.

### 4.1.4.1. First-In-First-Out Memory

AFirst-In-First-Out memory is an asynchronous data storage unit with independent read and write controls. Advanced Micro Devices' Am 2812 was chosen which is a 32 word by 8 bit device. FIFO is normally used as a buffer memory between two pieces of digital equipment operating at different speeds.

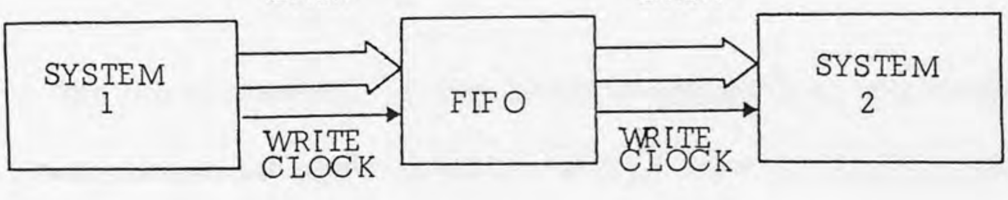


Fig. (4.8) Asynchronous Interface Using FIFO

There is a Parallel Load (PL) input, used to load an 8-bit word onto the first stage of the FIFO, and an InputReady output (IR) which indicates that the FIFO is ready to receive new data. At the output, there is a dump command (PD) used to bring the next data word to the output lines. Data entered into the FIFO falls through the registers until it reaches either output or another data word. The input and output,shift and ready signals can be directly connected end to end to make a longer buffer memory.

### 4.1.5. Elements of the Interface

The function of the individual elements in the interface is now explained to help understanding of the logic circuit developed.

### 4.1.5.1. General Purpose Interface Board

Data General Corporation has available standard general purpose interface that includes all of the ordinary circuitry needed to connect a device to the input-output bus. This board can be accommodated in the main frame of the mini computer with direct connections to the computer bus system. Part of this board is reserved for customer logic and mounting standard IC components, thus it was used in construction of the computer end of the interface.

The basic elements on this board as provided by the manufacturer are, two sixteen bit shift registers for input-output, device select, busy done, and interrupt circuitry.

### 4.1.5.2. Device Select

The input-output bus of the mini computer has six lines for device selection which allows up to $2^6$ combinations or 64 devices (77 in octal) to be addressed from which devices number 0 and 77 are reserved for special functions.

Certain device codes have already been allocated to various peripherals, device code 63 was therefore selected for the interface. As explained earlier, in a programmed data transfer the processor places the code of the selected device by software on the select lines (DS 0-5). In order to clock the data out of the output registers, the output from the eight input "NAND" gate should be '0', Fig. (4.9). Therefore, the jumper leads W1-W12 must be chosen in a way to enable having logic 1's at the input of the NAND gate.

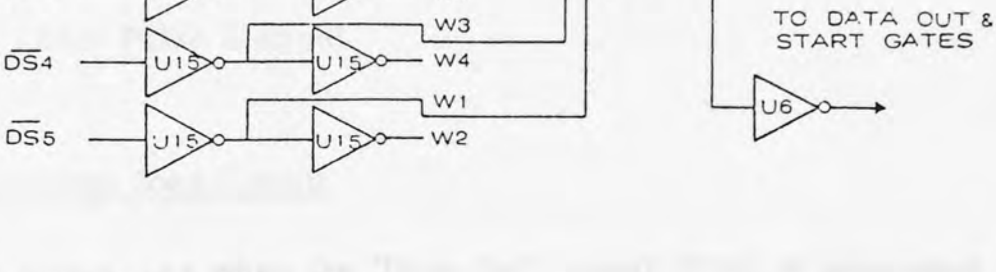


Fig. (4.9) Device Select Circuit

### 4.1.5.3. System Reset

The data holding components on the interface such as buffer store memories, registers, and receiver/transmitters are all provided with a Master Reset (MR). At the beginning of each data transfer session it is necessary to clear the interface from the unwanted information.

To achieve this, the first character to be output should place 8 logic 1's on the data lines, which after being "NAND gated" initiates the multivibrator (C 3) and on receiving the "Start" pulse, (refer to 4.2.1.1) produces a pulse of sufficient duration to clear the interface, Fig. (4.10).

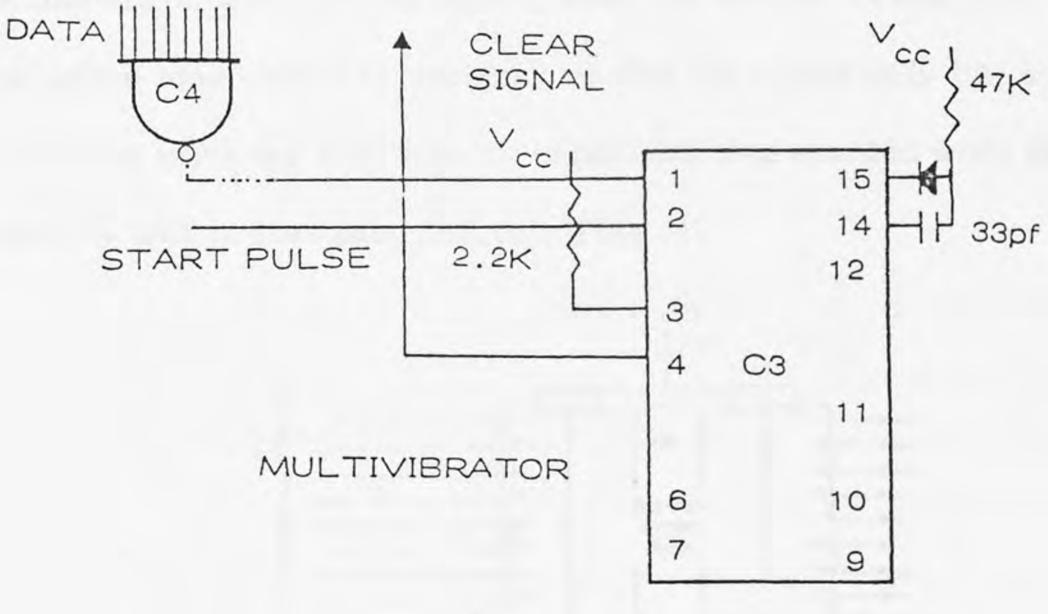


Fig. (4.10) Reset Pulse Element

### 4.1.5.4. Interface Load Circuit

In normal use when the "Data Out" signal (DOA) is generated the data is loaded onto the output lines of the computer bus regiers. The "Start" pulse then loads the data onto the first FIFO memory. When the data has been loaded the InputReady (IR) signal on the FIFO is generated to reset the computer "Done" circuit so that the next set of data can be transmitted, until the buffer is full.

If this procedure is implemented, data enters the memory in trickles whenever one data character is shifted out. In order to make efficient use of the buffer store, once the computer buffer memory is full so as to relieve the computer processor until all the stored data has been transmitted, the interface load circuit has been built.

As shown in Fig. (4.11) FIFO memories are incorporated with a flag signal that goes to logic 1 when the memory is half full. The interface load circuit is designed so that the InputReady (IR) signal is inhibited when the first memory is half full and enabled when the last memory unit is less than half full (Flag 0).

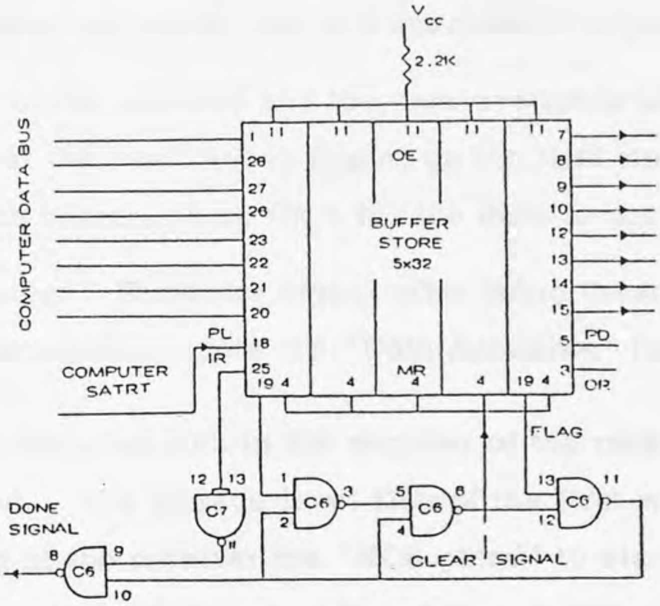


Fig. (4.11). Interface Load Circuit.

### 4.1.5.5. Transmitter Receiver

The data transfer between the two parts of the interface is carried out in a hand shaking mode through the transmitter and the receiver which are in contact with their respective buffer stores.

Two units are used one at each end, and both have their own crystal clock oscillating independently, but at the same frequency. If the following conditions are satisfied the transmission of a set of data can take place; namely, valid data is available on the output lines of the buffer store (Output Ready Signal generated), transmitter buffer is empty, and the receiver buffer is not full. In this case multivibrator (C3) is triggered and generates a 180 n sec. wide low going pulse. The leading edge of the pulse strobes the parallel data onto the transmitter and the trailing edge causes the parallel data to be put on the output lines of the memory.

The received serial data is assembled into parallel form in the holding register of the receiver and the data available signal is generated. Now provided that the Input Ready Signal on the first memory of the machine interface board is high (IR = 1), the data is put on the output lines of the receiver. The same signal after being inverted loads the data onto the memory and resets the "Data Available" flag.

While data remains in the register of the receiver, the overrun flag is raised. The storage level flag of the first memory unit and the over run flag of the receiver are "NOR gated" to signal when more data is required, refer to Fig. (4.14) and Fig. (4.15).

Both the receiver and the transmitter are hardwired to give even parity, eight data bits and two stop bits.

### 4.1.5.6. Interface Unload Circuit

If valid data is available in the FIFO memory the Output Ready signal (OR) goes to logic 1. Provided this is true, when the data is requested by the controller, (M5) produces a pulse that ongoing low dumps a new set of data onto the output lines ofthe FIFO memory. The circuit is designed so that the data flow is carried out at a certain sequence as shown in Fig. (4.13) to avoid any loss of data.

The machine interface board also includes a circuit that generates "code delete" when no valid data is available. This code is ignored by the machine control. To produce this code, the 'OR' signal is converted and then fed through eight NOR gates, the output of which if'OR' is low is passed to the controller.

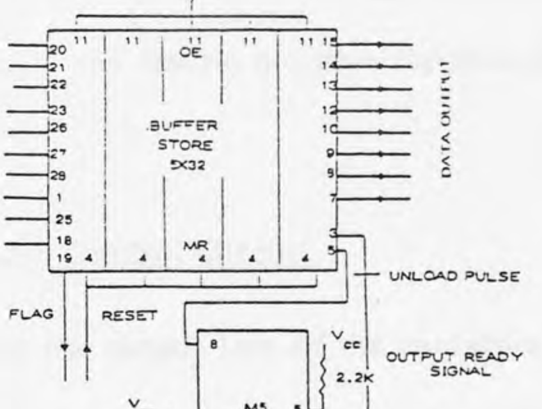


Fig. (4.12) Interface Unload Circuit

### 4.1.5.7. Tape Reader Time Matching Circuit

The "MOD 32K" signal from the Acramatic 5 control acts as a clock in inputting data through the tape reader. When data is requested the MOD 32K signal starts pulsing which controls a ring counter that in turn generates the phases for driving the stepping motor of the tape reader. When the tape perforations are sensed, the generated signals are amplified and squared up to produce pulses 2.5 – 3 m sec. wide.

In order to replace the tape reader data in the BTR system with the data from the interface, the same timing diagram should be obtained and thus the time matching circuit as shown in Fig. (4.15) was constructed. Two monostable multivibrators are used for time matching. The TTL compatible MOD 32K signal is the input to (M3) which on going high causes (M3) to produce a time delay of 1.8 m sec. This time delay signal is in turn input to the second unit(M4). The trailing edge of this signal produces a pulse 90 n sec. wide to clock the information onto the output lines of the registers. The timing diagram for this circuit is depicted in Fig. (4.13).

### 4.1.5.7.1. <u>Data Pulse Width Control Circuit</u>

A set of data on the output line of the registers remains unchanged until the next set is output. In order to limit the width of data pulses to 2.5-3 m sec. which is required by the NC control, this element was incorporated into the circuit. MOD 32K signal is used as an input to multivibrator unit (M9) by which is generated a pulse with sufficient duration that on going low clears the output registers.

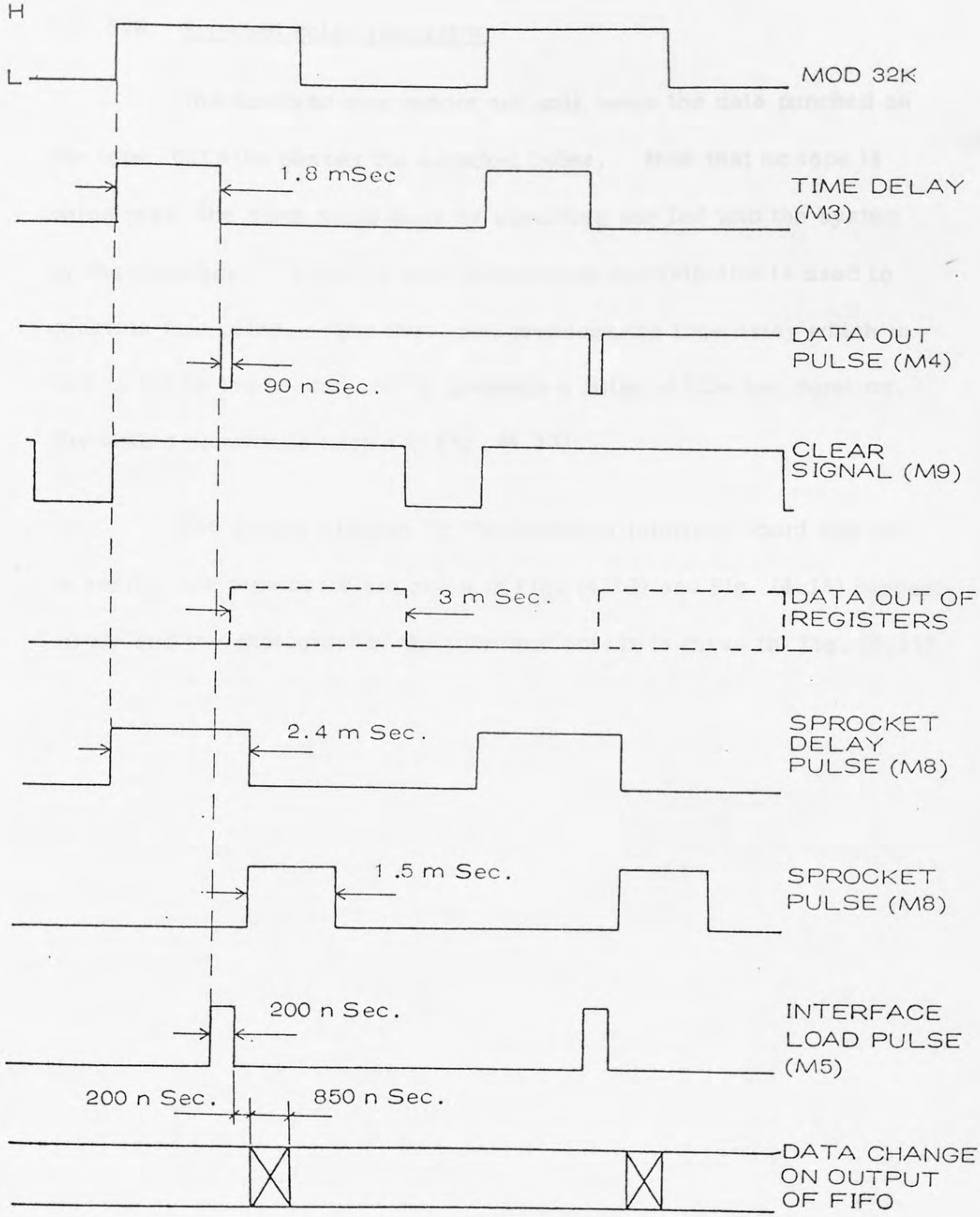


Fig. (4.13) Timing Diagram for Time Matching and Interface Unload Circuit.

### 4.1.5.8. Sprocket Pulse Generator

The Acraread tape reader not only reads the data punched on the tape, but also senses the sprocket holes. Now that no tape is being read, the same pulse must be simulated and fed into the system by the interface. A unit of dual monostable multivibrator is used to generate this pulse. The first part produces the time delay which in turn is fed to the second part to generate a pulse of 1.5m Sec. duration. The timing diagram is shown in Fig. (4.13).

The circuit diagram for the computer interface board and the machine interface board are given in Fig. (4.14) and Fig. (4.15) respectively and the photograph of the interface boards is given in Fig. (4.16).

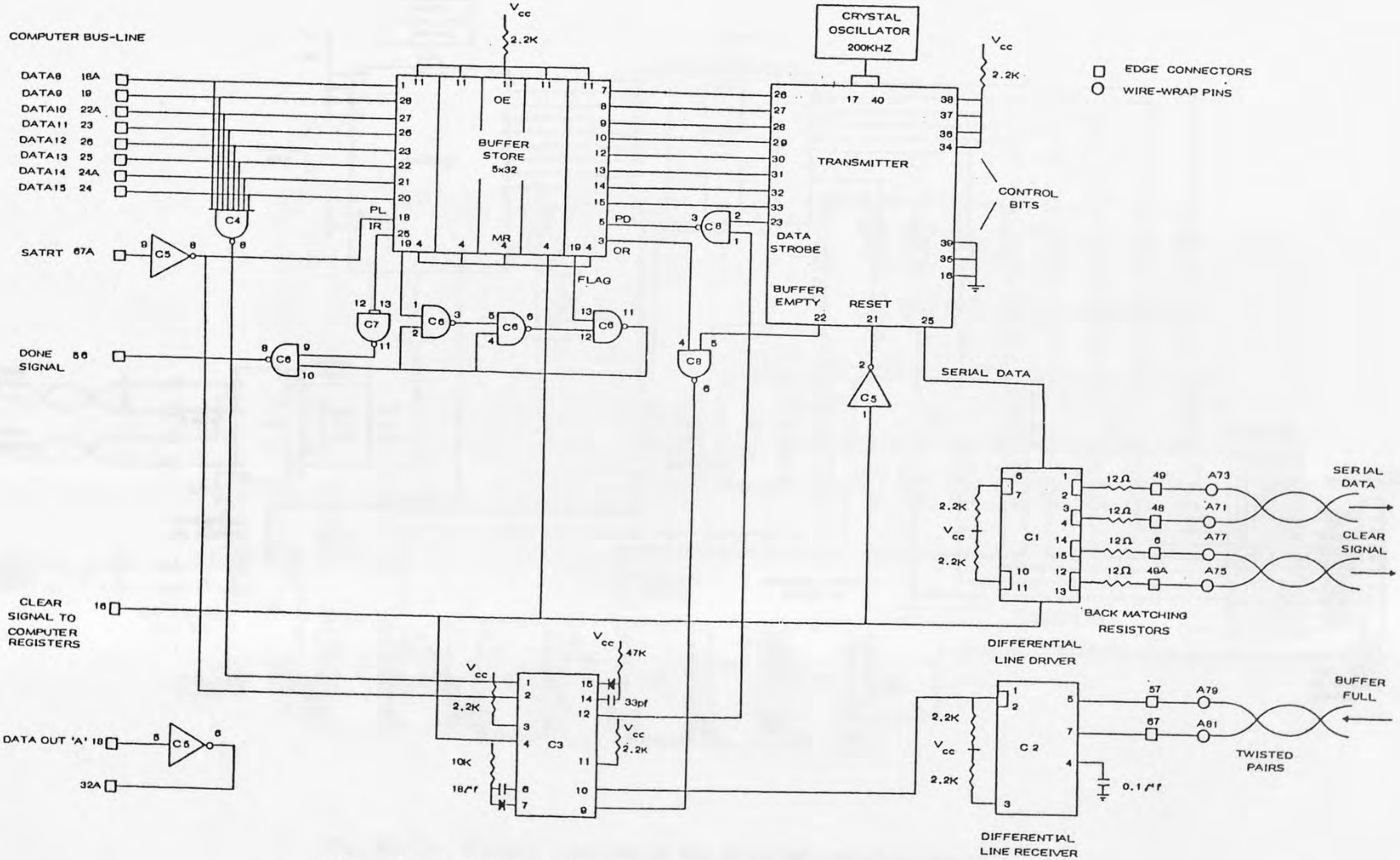


Fig. (4.14) Circuit Diagram of The Computer Interface Board

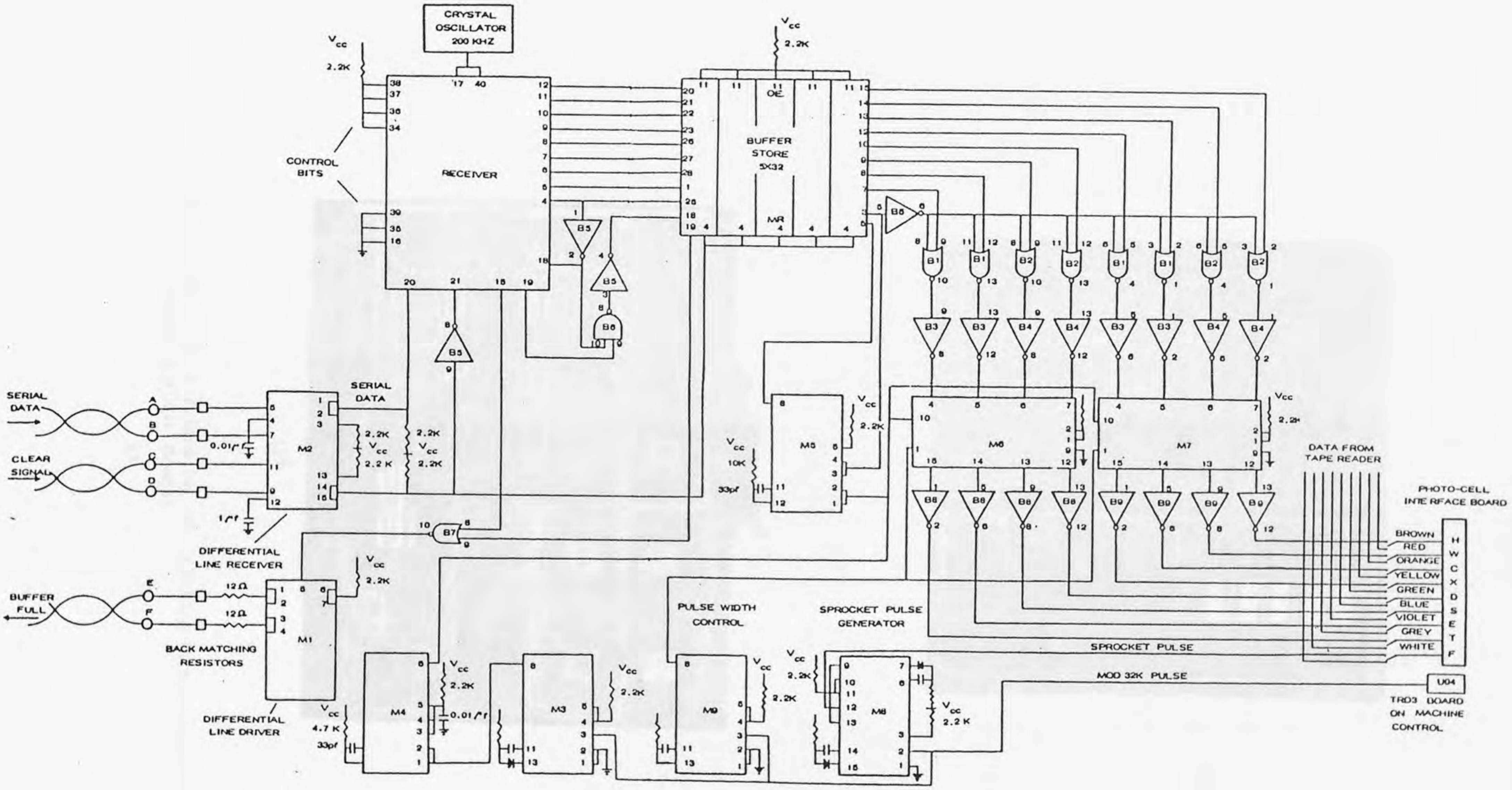
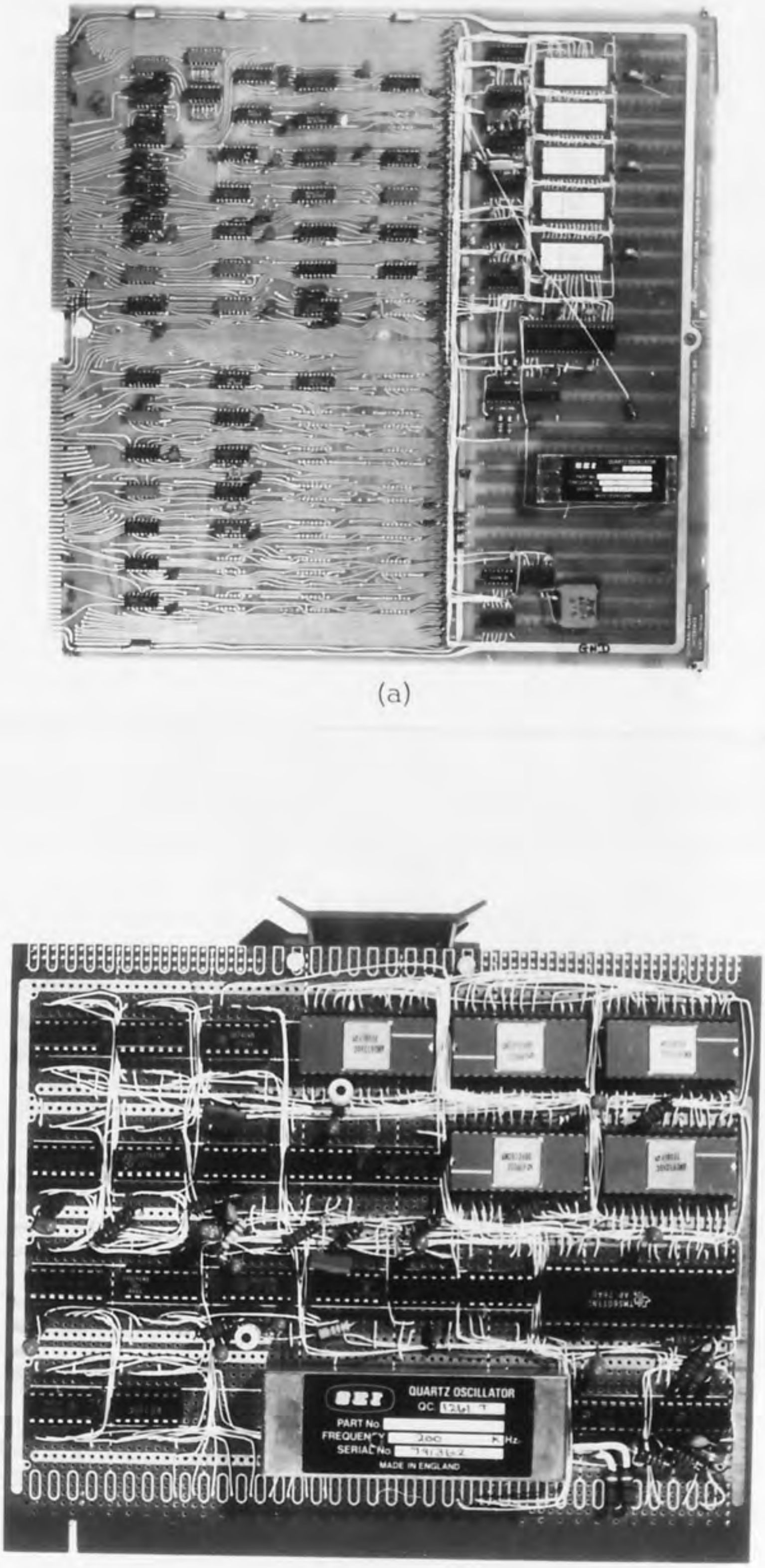


Fig. (4.15) Circuit Diagram of the Machine Interface Board



(a)

(b)

Fig. (4.16) Computer Interface (a) & Machine Interface (b) Printed Circuit Boards

## 4.2. Software Implementation of the Interface

### 4.2.1. Data Code in Numerical Control

Digital binary system has for years proved to be a very practical and logical way of communicating with many control systems, and punched paper tape has widely been used as a medium for this purpose. There are a number of codes for representing characters in binary and the Electronic Industries Association (EIA) code is the oldest and the most widely used in Numerical Control systems. The EIA contain only six information bits, and in theory provides only 63 distinct characters. The 5th track carries the parity check. The 8th track is occupied by only a single character, the carriage return, to enable the operator to visually find the end of one line of information. Now the use of only six tracks in EIA code restricts this form of coding. American Standard for Information Interchange (ASCII) code on the other hand, uses 7 bits with the 8th track for parity check. ASCII satisfies requirements for telecommunication, machine tools as well as for computers. In consequence of its more universal application, this code has been adopted by International Standard Organisation (ISO) for Numerical Control.

The control system on the machine used could accept both the EIA and the ISO codes. However, because some of the older equipment in the laboratory could only handle the EIA, it was decided to limit the coding of the part programs to this form. To communicate with the computer in this code,therefore, requires that code conversion be carried out.

### 4.2.2. Data Format in Numerical Control

Characters fed into the control systems are related to machine functions and this data, depending on the type of the control system, should be organised to a certain format. Several different formats are in common use, such as fixed sequential format, word address format, and tab sequential format.

In fixed sequential format the instructions in one block are always recorded in the same way. That is, all instructions, including those which remain unchanged, are given in every block.

In word address format, each word is preceded and identified by a letter called an 'address' and here the unchanged instructions are omitted. In tab sequential format each word is preceded by a tab character and the same sequence is retained.

The Acramatic 5 control accepts word address format with thirteen address words, and ignores the tab.

### 4.2.3. Control Programs of the DNC Cell

A set of software routines have been developed for the control of the DNC system, the main functions of which can be summarised as below:

I) Data input-output control

The control programs receive data from one peripheral and transfer it to another where the computer/machine interface is naturally regarded as one of the peripheral devices.

II) Code conversion

The transferred characters code should be changed so that it is compatible with the receiving device.

III) Format manipulation

The word format and the structure of the data block in the part program that is going to be transferred to the machine control must be manipulated according to the requirements of the NC controller.

IV) Valid data check

Control programs incorporate certain debugging routines to ensure that only valid data are sent behind the tape reader. Error messages are produced on the teletype if this is not the case.

These programs in addition to providing the necessary executive control over the part program transfer to the machine tool, cater for a few specific functions that greatly enhance the role of the computer in the system. For example, it is now made possible to load the conventionally prepared EIA coded tapes onto the disc. The edit facility of the operating system can therefore, be employed for any subsequent modification. NC machine coded and formatted tapes can also be produced using the high speed paper tape punch as a permanent file fromthe library of files on the disc. If there is any break down in the interface, part programs produced with computer assistance would directly be input through the tape reader.

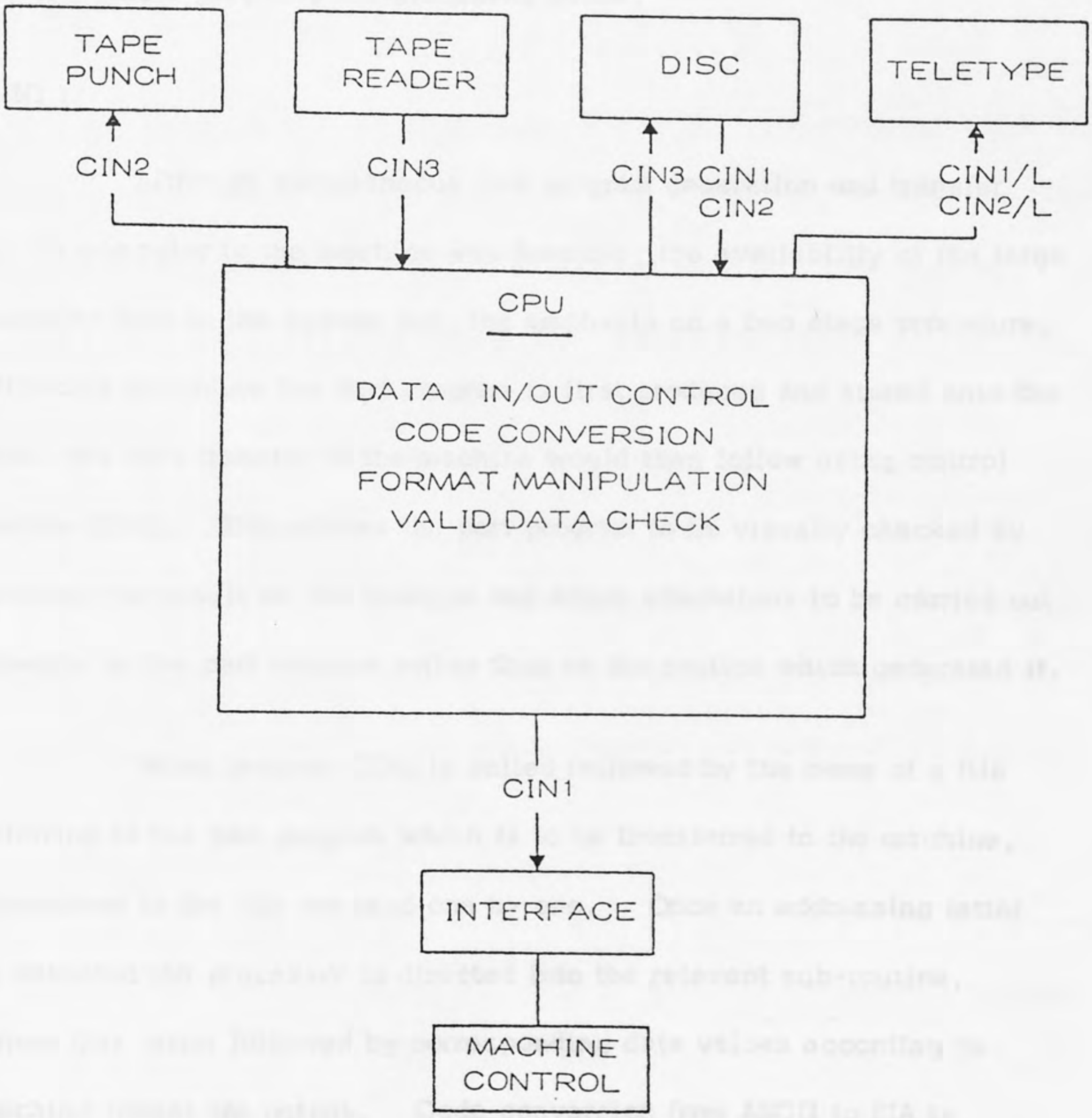


Fig. (4.17) Data Flow Under Control Programs

To perform the bit and byte manipulating operations required in the control routines, it was necessary to use a low level computer language. An Assembler was available,therefore, all control programs were written in Assembly language which would give very fast data transfer rates and required low computer storage area. The function of the individual programs are discussed below.

CINI :

Although simultaneous part program generation and transfer by the computer to the machine was feasible, the availability of the large capacity disc in the system put the emphasis on a two stage procedure. With this procedure the part program is first produced and stored onto the disc, the data transfer to the machine would then follow using control routine CIN1. This allows the part program to be visually checked by printing the result on the teletype and minor alterations to be carried out directly on the part program rather than on the routine which generated it.

When program CIN1 is called followed by the name of a file referring to the part program which is to be transferred to the machine, characters in the file are read one by one. Once an addressing letter is detected the processor is directed into the relevant sub-routine, where this letter followed by corresponding data values according to machine format are output. Code conversion from ASCII to EIA is carried out and such details as suppression of decimal points, addition of the leading zeros and tab characters are accomplished.

CIN2:

In function it is similar to CIN1, the only difference is in the output device being controlled. Here the result is produced in the form of punched paper tape rather than the direct transfer to the NC machine. With this routine the advantages of the computer aided part programming can be gained regardless of the interface.

CIN3:

To make full use of the computer in part programming, CIN3 was developed which enables EIA coded tapes to be read into the computer. It reads each character and converts the code from EIA to ASCII and then loads it onto the disc. Listing of these programs and a guide to their use are given in Appendix I.

## 4.3. Addition of a Rotational Axis

This project was conceived out of the need for a device which would enable surfaces to be generated (on milling machines) which consist of continuous curves, variable lead helices and non-cylindrical helicoids.

The most commonly adopted solution involves the superimposition of an additional rotation upon a basic system with axial translation of cutter relative to the workpiece. The conventional method is to use a universal or vertical milling machine with an inclinable workhead in conjunction with a dividing head driven by the table lead screw.

All conventional methods employed to provide rotational movements suffer from severe limitations which can be summarised as follows :

i. Only one relationship can be accommodated, a varying lead hedix with the same mathematical form from start to finish as rotational and axial translations are limited to one speed.

ii. Dwells can not automatically be accommodated.

iii. Axial translation is restricted to the longtitudinal or lead screw axis only.

Any change therefore requires manual intervention such as changing the gear ratio, disengaging the table feed for a dwell, etc., and hence, an operator in attendance is needed.

By the development of a fourth axis drive suitable for retrofitting to existing NC machine tools and linked to the computer in the DNC system, many of the problems mentioned can be resolved. Translational movements will be possible in any direction with clock-wise and anti-clock-wise rotation as well as complete cessation as and when required.

Helical grooves with more complex mathematical relationships could be cut and in addition, grooves with varying helix angle can be produced without the need to modify the drive gear ratio. The range of machining operations possible will therefore be greatly extended, and works which previously required a dedicated drive may be, by software manipulation, readily accommodated.

### 4.3.1. Design Considerations

The movements of the independently driven axis must be correlated to the movements of the other axes. Ideally a closed loop control is required, preferably under total computer control with some means of control over rotational speed, and encoding devices to monitor the actual position.

Alternatively at a cost of some loss of accuracy an open loop system in the following forms could be devised. a) A variable speed drive, consisting of a stepping motor, where control over the rate of drive pulses determines the rotational speed. b) A constant speed drive and a cyclic encoder to feed back the angular displacement information to the computer which is then used to modify the machine linear feed rate.

The determining factor in adopting one of the above methods in this case was the availability of the equipment. A stepping motor with a constant speed drive was readily available; and hence, only a suitable cyclic encoder was needed to complete the system.

### 4.3.2. Elements of the Fourth Axis

A dividing head was used as work holding device because it could be geared down through its internal and external gear trains. In addition, it could be indexed manually for setting up and would allow the angle of work-piece to be changed as and when required.

The stepping motor available would generate a torque of 300 oz-inch, and with the existing drive could run at a speed of 240 rpm. The power therefore generated would only be sufficient for light machining. The cyclic disc encoder used was constructed in the department and was based on Excess 3 (XS 3) code where the reading head consisted of an array of ten transmission diodes and ten infra-red sensitive diodes. XS3 code was chosen because of its favourable properties in design of encoders (63).

### 4.3.3. Control of the Fourth Axis

The output from the electronic circuitry of the reading head was directly connected to the input-output bus of the mini computer. A line from the computer for start/stop of the rotation, and a link to carry the data request signal from the machine tool controller to the computer were also provided.

In order to treat the rotational axis attachments as an integral part of the DNC system, the control program CIN1 (refer to Section 4.2.3.) was modified to cater for the control of this axis as well as transfer of data behind the tape reader. This modification also enabled a part program generating subroutine to be called and as a result, an interactive link to be established. While running, the angular position of the component via the encoder and the state of data request signal is constantly sampled and fed into the part program generating subroutine where accordingly, the feed rate and position coordinates, and the instructions regarding the rotation of the fourth axis are produced and then output. Block diagram of the added rotational axis is shown in Fig. (4.18).

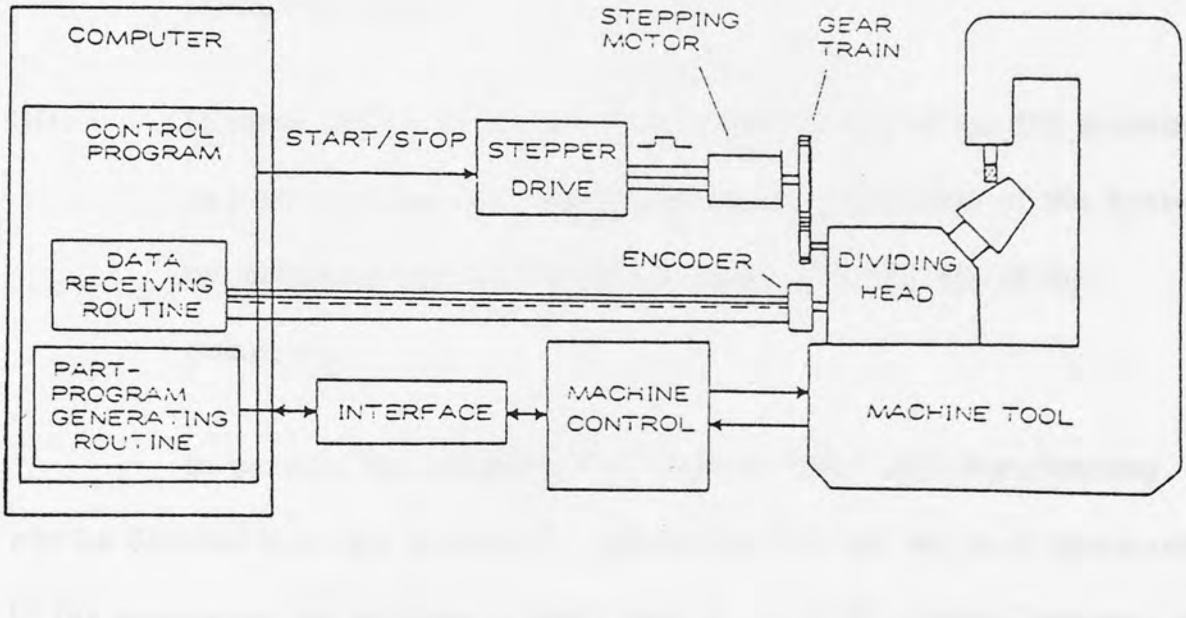


Fig. (4.18) Block Diagram of the Rotational Axis

## 4.4. Computer Aided Part Programming

If a component needs a great deal of machining, involving very many sequence elements and requiring much calculation, the preparation of tape by manual methods would be very time consuming. To facilitate the programming process a computerised method can be used.

The reasons for adopting computer aided part programming can be outlined as below :

I) Part programming procedure is largely simplified and speeded up, and considerable time saving can be achieved.

II) The task of calculation is transferred from programmer to the computer, therefore, the possibility of making an error can be reduced and programmer is relieved of an extremely tedious and labourious job.

III) In some cases the scope of the capabilities of the NC machine can be extended beyond the hardware limitations of the system by manipulations in the part program with the aid of the computer.

In general the software for computer aided part programming can be divided into two sections; the processor and the past-processor. In the processor the problem is dealt with in the most general manner, a fictional tool path for a given tool is determined and its value accompanied by some other relevant information are stored in an external store.

This type of output is known as the CL tape (Cutter Location). Programs such as APT, 2CL, ADAP, EXAPT, PROFILE DATA, etc. are some of the most widely used processors.

The output of processors must be put into a form suitable for the control system of the particular machine tool on which the workpiece is to be machined, and checks must be made to ensure that it does not make demands over the capacity of the machine tool. This task is performed using a program called post-processor.

### 4.4.1. Part Programming in the DNC System

For the NC machine in the DNC cell there was no post-processor available in the department so that computer aided part programming as described could be exercised. In order to overcome this shortcoming the following simple procedures were adopted.

a) Combined processing and post-processing.

In this primitive method of data preparation, high level computer languages such as FORTRAN and BASIC were used, allowing complicated calculations to be made and tool compensations computed. Both the calculation of the cutter location, and the structuring of data in machine compatible format are carried out in a single purpose written program, The part program stored onto the disc can then be modified when necessary or re-used.

b) Post-processing of cutter location data.

If the co-ordinates of cutter centre-line path are already produced using routines such as those for computer aided surface design, a simple post-processor using FORTRAN was written to read this data from the tape or disc and supplementing it by preparatory and miscellaneous functions, feed rate, etc.

# CHAPTER FIVE

# DEVELOPMENT OF THE COMPUTER AIDED SURFACE DESIGN PACKAGE "BCSURF"

## 5.1. Introduction to BCSURF

Computer Aided Design of curved or free form surfaces is based on mathematical description of the shape from which instructions for a numerically controlled machine tool, or drawings can be produced. One approach to categorise the CAD systems may be in terms of mathematical basis. Three categories of surface representation can be distinguished(64). The first category is the cartesian product surface, which interpolates through point data only. Bezier surfaces and B-spline surfaces are usually implemented as cartesian product surfaces with Bernstein and B-spline polynomials as bivariate interpolants, respectively. The second category lofting, utilizes one family of parametric curves and the surface is interpolated over this set of curves. Finally the transfinite surface representation, which is the result of interpolation through two families of curves. The interpolants in this last case are bi-orthonormals such as cardinal spline functions. Coons surfaces can be viewed as a special form of the transfinite approach which employds a Hermitian patch-like interplant.

BCSURF, an experimental surface description method has been developed in the present work and is explained in the subsequent Sections. Vector valued parametric representation is adopted and the basic scheme employed is based on lofting technique. The cross-sectional curves are represented by uniform B-splines where the order of the curves can be chosen by the user and is used as one of the "handles" in achieving the desired shape for the sectional curves. The parametric method used enables planar / non-planar, closed or open curves in three dimensional space to be designed, and there is no need for the sectional curves to be parallel. The surface between the cross-sectional curves is interpolated by cubic cardinal splines. Again, using closed parametric cardinal splines, ring-like objects having periodic surfaces can be designed very simply.

All programs in this package are written in FORTRAN IV and run on the ICL 1904 computer under GEORGE 3 operating system. Interactive facilities were not available so that the routines could be written for implementation in an interactive environment, therefore, batch processing has been employed and Gino-F (65) graphical routines were used to produce graphical outputs. The programmed mathematical formulation of the method used will,of course, facilitate the development of a future interactive CAD system that can be ultimately established.

## 5.2. Mathematical Basis of BCSURF

### 5.2.1. Definition of Spline Functions

Given an increasing set of real numbers $x_o, x_1, --- x_L$, a function S is called a (polynomial) spline function of degree n-1 (order n) if the following two conditions are satisfied.

i) In each interval $(x_i, x_{i+1})$ S is a polynomial of degree n-1.

ii) S and its derivatives of order 1,2,..... n-2 are continuous everywhere.

The points $x_i$ are called the knots, and the ith span of the spline lies in the interval $(x_i, x_{i+1})$.

### 5.2.2. B-Spline Basis

Several basis functions have been introduced in representing spline functions, among which the truncated power functions, cardinal splines, and B-splines are best known and given most serious attention.(66)

In Fig. (5.1) B-spline basis functions are given for progressively increasing degree and having knots at integers. B-spline is said to be spline of minimal support, as Fig. (5.1) shows for degree n-1, the basis functions have finite local support of width n, that being the number of spans over which a spline is non-zero. Thus, such bases are local in the sense that at every point, only a fixed number (equal to the order) of the B-splines is non-zero.

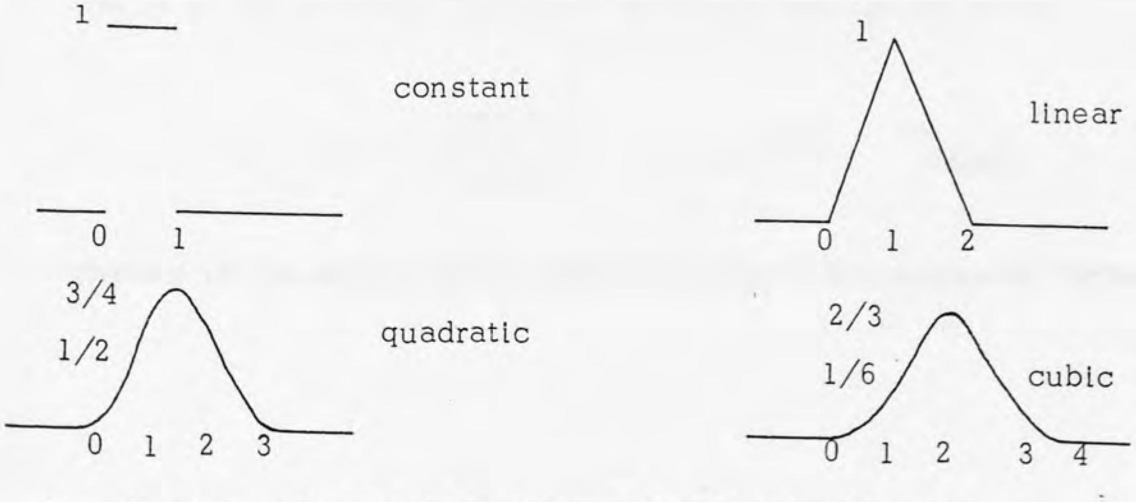


Fig. (5.1) B-spline Basis Functions for Different Degrees

### 5.2.2.1. B-Spline Versus Bernstein Basis Function

B-Spline was first introduced by Schoenberg (53) in the paper that dealt with statistical data smoothing and approximation. From a mathematical point of view basis functions or in another term weighting functions relate a curve to a defining polgon.

Recalling Section 3.3.3.5., Bernstein basis function produces Bezier curve and the B-spline basis is proved (11) to contain Bernstein basis as a special case.

Although Bernstein basis function shows several useful properties, its characteristics, however, limit the flexibility of the resulting curve. Firstly the order of the curve is determined by the number of defining polygon vertices. For example, a cubic curve must be defined by a polygon with four vertices and a polygon with six vertices will always produce a fifth-degree curve. Therefore, to increase or reduce the order of the curve the number of vertices must be increased or reduced respectively.

Secondly, Bernstein basis function (5.1) is global in nature, that is to say it is non-zero over an entire span of the curve.

$$B_{n,i}(u) = \frac{n!}{i!\,(n-i)!}\; u^{i}\,(1-u)^{n-i} \qquad 0 \leq u \leq 1 \qquad (5.1)$$

where n is the degree of the polynomial and i the particular vertex.

Any point on a Bezier curve is a result of weighting the values of all defining vertices, a change in one vertex is felt throughout the entire span. This therefore, eliminates the ability to produce a local change within one span.

The Bezier curve is tanget to the polyhedron sides at the end points and it is possible to change the shape of a five-point polygon without changing the direction of the end slopes. However, any change will affect the shape of the curve everywhere.

The B-spline function on the other hand is non-global. This is due to the fact that although each vertex is associated with a unique B-spline basis function, this function is only non-zero over a limited number of spans. Thus, any change on a vertex will result in a local change, namely only over the range of parameter values within which the function is non-zero.

In addition the order of the curve is not fixed by the number of polygon sides. The B-spline basis allows the order of the resulting curve to be changed without changing the number of polygon vertices.

### 5.2.2.2. Evaluation of B-spline Basis Functions

Although B-spline basis can be expressed in various ways, the recursive method of Cox (67) and de Boor (66), provides a stable and efficient way of evaluating the B-spline basis functions.

For the $i^{th}$ normalised B-spline basis curve of order n, the weighting function $N_{i,n}(u)$ are defined by the recursive formulae.

$$N_{i,1}(u) = \begin{bmatrix} 1 & x_i \leqslant u \leqslant x_{i+1} \\ 0 & \text{otherwise} \end{bmatrix}$$

$$N_{i,n}(u) = \frac{(u-x_i)\, N_{i,n-1}(u)}{x_{i+n-1} - x_i} + \frac{(x_{i+n} - u)\, N_{i+1,n-1}(u)}{x_{i+n} - x_{i+1}} \qquad \text{for } n > 1 \qquad (5.2)$$

The convention $0/0 = 0$ is assumed here.

The values of $x_i$ are elements of a knot vector. A knot vector $X = (x_o, x_1, \ldots\ldots x_L)$ may contain identical knots up to multiplicity n. Multiple interior knots reduce the degree of differentiability of basis functions. With M multiple knots at $x_i$, $N_{i,n}$ will have differentiability up to $C^{(n-M-1)}$ at this point.

The other effect of having multiple knots is the induction of spans with zero length and consequently a reduction in the width of support of the basis function. Multipticity M of the interior knots generate M-1 zero length spans.

A knot vector is simply a series of real-integers $x_i$ such that $x_i \leqslant x_{i+1}$ for all $x_i$. For non-periodic basis multiple knots with multipticity equal to the order of the curve must be defined at each end of the knot set. For operiodic basis, knot vector consists of a series of equally spanned numbers such as X = (0,1.... L) or a cyclic shift of the above. In Fig. (5.2) basis splines for a non-periodic and aperiodic curve are given.

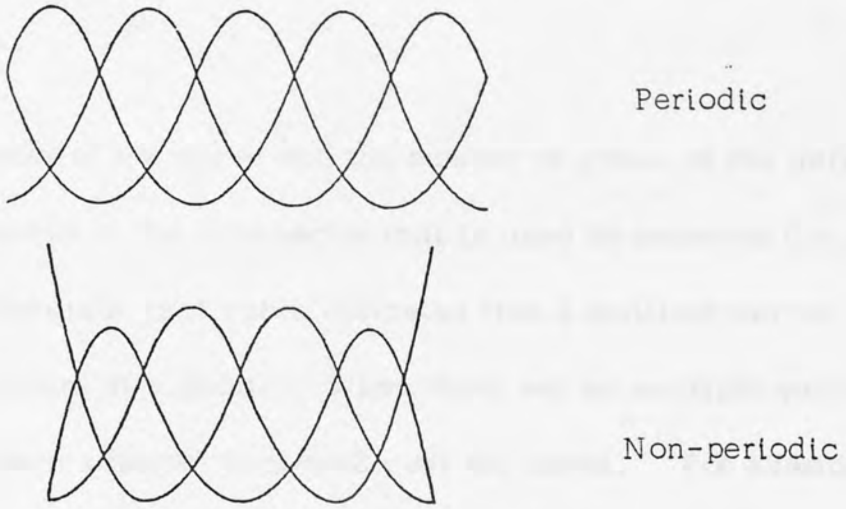


Fig. (5.2) B-Splines for Periodic and Non-periodic Curves

In a parametric curve, the values of $x_i$ are considered to be parametric knots. They can be used to indicate the range of parameter u used to generate a B-spline curve. For example, the vector (0 1 2 3) indicate that the parameter u varies from 0 to 3.

### 5.2.3. B-spline Curve

A B-spline curve is an approximation to a defining polygon. Taking r (u) to be position vector along the curve as a function of parameter u, a curve generated by the use of the B-spline basis is given by

$$r(u) = \sum_{i=0}^{m} P_i N_{i,n}(u) \tag{5.3}$$

where $P_i$ are m+1 vertices of the defining polygon, and n the order of the curve.

The first step in generating a B-splinecurve is to define the knot vectors. In this work uniform parametric B-spline basis as defined by Gordon and Riesenfeld $^{(44)}$ was adopted in which knots are restricted to integer values.

The order of the curve and the number of spans of the defining polygon are reflected in the knot vector that is used to generate the curve. A duplicate intermediate knot value indicates that a multiple vertex (span of zero length) occurs at a point. When there are no multiple vertices, the parameter u varies from 0 to m-n+2 over the curve. For example, for a fourth order curve defined by vertices ($P_o$, $P_1$, $P_2$, $P_3$, $P_4$), this is given by $0 \leqslant u \leqslant 4 - 4 + 2 = 2$, and the complete set of knots with multiplicity of 4 at each end is given by

$$\underbrace{0000}_{n} \; 1 \; \underbrace{2222}_{n}$$

For a closed (periodic) curve a cyclic set of knots must be defined (that is, $x_L = x_o$, $x_{L+1} = x_1$, ..... and soon), no multiple knots in this case are necessary at the end points.

### 5.2.4. Interpolating with B-splines

In an interpolating question with B-splines, it is natural to construct splines of order n on a set of points $x_o$, $x_1$, .... $x_L$ that is expressed as a sum of multiples of B-splines defined on the same set of points now used as knots and extended by (n-1) additional knots at each end.

Expression (5.3) can be rewritten in the form

$$S(x) = \sum_{i=0} C_i \; N_{i,n}(x) \tag{5.4}$$

where $S(x)$ is any spline of degree n-1 on the given set of points and $C_i$ are numerical coefficients.

If the values to be interpolated at $x_o$, $x_1$, .... $x_L$ are $z_o$, $z_1$, .... $z_L$, then

$$z_J = S(x_J) = \sum_{i=0} C_i \; N_{i,n}(x_J)$$

For simplicity if one considers a cubic case and calculates the values of $N_{i,n}(x_J)$ using de Boor algorithm (66), for each $X_J$; ($J = 0$, ..... L) only three values $N_{i,4}(x_J)$, $N_{i+1,4}(x_J)$, $N_{i+2,4}(x_J)$ are non-zero. Therefore, from (5.4) one has

$$Z_J = S(x_J) = C_i N_{i,4}(x_J) + C_{i+1} N_{i+1,4}(x_J) + C_{i+2} N_{i+2,4}(x_J)$$

in which the C's are unknown. There will be one such linear equation for each of the (L + 1) knots ($x_o$, $x_1$, .... $x_L$). But there are (L + 3) B-splines ($N_{o,4}(x_o)$, $N_{1,4}(x_1)$, ..... $N_{L+2,4}(x_L)$), and hence (L + 3) coefficients $C_o$, $C_1$, .... $C_{L+2}$ to be determined. One therefore, needs two extra items of information to determine S (x) completely. These will usually be derivative values. For instance, if $\ddot{Z}_J = \ddot{S}(X_J) = 0$, then for J = 0

$$\ddot{Z}_o = \ddot{S}(x_o) = C_i \ddot{N}_{i,4}(x_o) + C_{i+1} \ddot{N}_{i+1,4}(x_o) + C_{i+2} \ddot{N}_{i+2,4}(x_o)$$

and given a similar condition at $x_L$, there is enough information to calculate the values of the coefficients. The values of the B-spline derivatives can be evaluated as suggested by de Boor.

Once all the coefficients have been determined Expression (5.4) can again be used to find the function values for any desired x.

#### 5.2.4.1. Inversion Technique

The problem of interpolation can be solved in a more elegant and efficient way using an inversion method as introduced by Gordon and Riesenfeld (44). This is the procedure employed in this work, which allows any degree curve to be interpolated on a given set of points without the need to put extra conditions at end points.

Given a function $f : [o, k] \rightarrow R$, the spline approximation of order n to f is defined as :

$$S_n(s) = \sum_i f(\xi_i) N_{i,n}(s) \tag{5.5}$$

where $$\xi_i = \frac{1}{n-1}(x_{i+1} + x_{i+2} + \dots x_{i+n-1}) \tag{5.6}$$

The $\xi_i$'s are called the nodes.

If the polygon P is considered as a piecewise linear function F, so that the B-spline curve is the parametric B-spline approximation to F, then

$$F(\xi_i) = P_i \qquad i = 0, 1, \ldots m \qquad (5.7)$$

Suppose a set of points lie on a spline S, the problem is to find the unique polygon $P_o, P_1, \ldots\ldots P_m$, the B-spline approximation to which corresponds to S. By determining the knot set $x_o, x_1, \ldots\ldots x_L$ as required by S, the nodes can now easily be found using (5.6). Now the knots and the nodes available the matrix N as defined below can be calculated.

$$N = N_{J,n}(\xi_i) \qquad 0 \leqslant i \quad , \quad J \leqslant m \qquad (5.8)$$

From (5.5), (5.7), and (5.8) this leads to finding a polygon to satisfy

$$N \cdot (P_o\ P_1\ \ldots\ldots\ P_m)^T = (S(\xi_o)\ \ S(\xi_1)\ \ \ldots\ldots\ \ S(\xi_m))^T$$

The problem of interpolation can now be solved by replacing the parametric values $S(\xi_o), S(\xi_1), \ldots\ldots S(\xi_m)$ by the actual cartesian point data to be interpolated.

### 5.2.5. Cubic Cardinal Splines

Here the attention is limited to cubic cardinal splines and in particular, the method introduced by Nilson, E.N. (68) to construct them. In this method, two cubic arcs as defined below are used ;

$$T1(\theta) = \theta - \theta^3 , \quad T2(\theta) = (\theta^2 - \theta^3)/2$$

$R(\theta)$ with parameter $\alpha$ is also introduced,

$$R(\theta) = 1 - (\alpha + 3)\theta^2 + (\alpha + 2)\theta^3 \qquad 0 \leqslant \theta \leqslant 1$$

The construction of any cubic cardinal spline on a uniform mesh begins with the arc $R(\theta)$ over $[0, 1]$ to which is attached arcs made up of suitable linear combinations of $T1(\theta)$ and $T2(\theta)$.

Considering a mesh consisting of integar points $0, \pm 1, \pm 2,$ ....., splines $A(\theta)$ with $A(0) = 1$, and $A(\pm n) = 0$; $n = 1, 2, \ldots$ are as follows:

on $[0,1]$ $\quad A(\theta) = R(\theta)$,

on $[1,2]$ $\quad A(\theta) = \dot{A}(1)\,T1(\theta - 1) + \ddot{A}(1) \cdot T2(\theta - 1)$,

on $[2,3]$ $\quad A(\theta) = \dot{A}(2)\,T1(\theta - 2) + \ddot{A}(2)\,T2(\theta - 2)$,

together with $A(\theta) = A(-\theta)$ it is proved that on $[n, n+1]$, $n \geqslant 1$, $A(\theta)$ is defined by

$$A(\theta) = (\alpha_n \alpha - 3\alpha_{n-1})\,T1(\theta - n) + \left[2(2\alpha_n + \alpha_{n-1})\alpha + 6(\alpha_n + 2\alpha_{n-1})\right] T2(\theta - n) \tag{5.9}$$

where $\alpha_n + 4\alpha_{n-1} + \alpha_{n-2} = 0$, $\alpha_o = 0$, and $\alpha_1 = 1$

The choice of the constant $\alpha$ is governed entirely by consideration of convenience.

A non-periodic spline on the mesh $\Delta$ : $x_o < x_1 \dots < x_N$ can be defined as

$$S(x) = \sum_{i=0}^{N} f_i A_i(x) + b_o B_o(x) + b_N B_N(x) \qquad (5.10)$$

where $f_i$ $(i = 0, \dots N)$ are the ordinates,

and $B_o(x)$ and $B_N(x)$ cardinal splines for implementing the end conditions.

For "free-b ending" ends, i.e. $\frac{d^2 S(x)}{dx^2} = 0$ at $x = x_o$ and $x = x_N$ (5.10) can be reduced to

$$S(x) = \sum_{i=0}^{N} f_i A_i(x) \qquad (5.11)$$

This can be applied to both periodic and nonperiodic, provided that for a periodical case one takes $f_N = f_o$.

In developing this algorithm Nilson (68) takes the value of $\alpha$ to be $\alpha = 3 \frac{\lambda - \lambda^{N-1}}{1 - \lambda^N}$ , $\lambda = -2 + \sqrt{3}$

Thus on $[n, n+1]$ , $1 \leqslant n \leqslant N-1$ using $\lambda^n = \alpha_n \lambda - \alpha_{n-1}$,

$$A(\theta) = \frac{3\lambda^{n}}{1-\lambda^{N}} T(\theta-n) + \frac{3\lambda^{N-n-1}}{1-\lambda^{N}} T(1-(\theta-n)) \tag{5.12}$$

and on $[0,1]$

$$A(\theta) = (1-\theta)^3 + 3\left[\frac{1}{1-\lambda^{N}} T(\theta) + \frac{\lambda^{N-1}}{1-\lambda^{N}} T(1-\theta)\right] \tag{5.13}$$

where $T(\theta) = \theta - (\lambda+2)\theta^2 + (\lambda+1)\theta^3$

Given the uniform mesh $\Delta$: $x_o, x_1, \ldots\ldots x_N$, $h = x_J - x_{J-1}$, one obtains for x on $[x_k, x_{k+1}]$, $(k = 0,1,\ldots\ldots N)$, and defining $\theta = (x-x_k)/h$, the Expression

$$S(x) = \frac{T(\theta)}{1-\lambda^{N}}\left[\sigma_{N-1}\lambda^{k+1} + \sigma_k(1-\lambda^{N})\right] + (1-\theta)^3 f_k + \frac{T(1-\theta)}{1-\lambda^{N}}\left[\tau_1\lambda^{N-k} + \tau_{k+1}(1-\lambda^{N})\right] + \theta^3 f_{k+1} \tag{5.14}$$

where $\sigma_k = 3\left[f_o\lambda^k + \ldots\ldots\ldots + f_{k-1}\lambda + f_k\right]$,

and $\tau_k = 3\left[f_k + f_{k+1}\lambda + \ldots\ldots\ldots\ldots\ldots + f_N\lambda^{N-k}\right]$

The parametric form of this expression was used to interpolate the cross-sectional curves designed by the B-splines in forming the surface.

## 5.3. Curve Design Procedure

In the routines "BSCURV" developed for design of B-spline curves, the most general form as shown by Expression (5.15) has been implemented and the basis functions are evaluated using the recursive formulae given by (5.2).

$$r(u) = \sum_{i=0}^{m} P_i N_{i,n}(u) \tag{5.15}$$

This method allows curves of different order, closed or open, planar or space curves to be designed.

BSCURV is written in the form of nested subroutines whereby the knot vectors for closed or open curves are defined in their respective subroutines using the information about the defining polygon. In both cases knots are given at integers. For an open curve the knot set

$x_o, x_1, \ldots\ldots x_{m+n}$ is defined in which

$x_o = x_1 = \ldots\ldots x_{n-1} = 0$ , and

$x_m = x_{m+1} = \ldots\ldots x_{m+n} = u_{max}$

Multiple vertices are checked and corresponding multiple knots are inserted. Now without any multiple vertex

$x_i = x_{i-1} + 1$ , $i = n, m$

otherwise, if $P_{i-n} = P_{i-n+1}$

$$x_i = x_{i-1}$$

In a closed curve a method slightly different from what described in Section 5.2.3. was adopted. Here for an $n^{th}$ order curve having the control polygon $(P_o, P_1, \dots . P_m)$, $P_{m+1} = P_o$, $P_{m+2} = P_1$, ....., and $P_{m+n-1} = P_{n-2}$ are defined, therefore, no multiple knots at the beginning and the end of the knot set are given, and the new defining polygon $(P_o, P_1, \dots . P_m, P_{m+1} = P_o, \dots , P_{m+n-1} = P_{n-2})$ is treated as an open one.

The first step in designing a B-spline curve using BSCURV is to define or sketch the defining polygon. The number and the co-ordinates of the polygon vertices accompanied by the order of the required curve are then input to BSCURV. The user can also specify the number of points generated along the curve, the range of parameterisation (0 to $u_{max}$) divided by this number will therefore be the parametric intervals at which the B-spline curve is evaluated.

## 5.4. Curve Fitting

In many design problems, functional, aesthetic, or scientific constraints governing the design result in some information regarding the shape of the object. The information may be in the form of some empirical data obtained statistically or experimentally about the geometrical shape, or in the form of a physical model, photographic information, drawings, sketches, etc. Computer aided design based on mathematical representation of shapes must be able to use the above information and generate a surface which closely represents the existing shape or approximates to the available data. For this reason curve and surface fitting are a necessary part of every CAD system. Taking the curve as the basic element in design, only curve fitting routines have been developed in this work and are explained here. It was decided to base the routines on mathematics of B-spline because; firstly the result could very easily be used in surface design process, secondly, the schemes were very efficient.

### 5.4.1. Least Squares B-spline Curve Fitting

The problem is to fit a B-spline curve to a set of data points $(x_J, y_J)$, $J = 1, 2, \ldots\ldots, L$. Assuming the spline $S(x)$ being constructed on knot set $\lambda_1, \lambda_2, \ldots\ldots, \lambda_m$ (where $\lambda_1 < \lambda_2 \ldots < \lambda_m$), we need to define $\lambda_{2-n} = \ldots\ldots = \lambda_o = \lambda_1$, and $\lambda_m = \lambda_{m+1} = \ldots \lambda_{m+n-1}$, where n is the order of the B-spline. With this augmented set of knots, we can define m + n basis splines $N_{i,n}(x)$, $i = 1, 2, \ldots m + n$. Then according to Curry & Schoenberg [41] $S(x)$ in the range $x_1 < x < x_L$ can be represented in the form

$$S(x) = \sum_{i=1}^{m+n} C_i N_{i,n}(x)$$

The curve fitting problem then becomes a matter of determining values for the coefficients $C_i$ as the least squares solution to the observation equations [(69)].

$$S(x_J) = \sum_{i=1}^{m+n} C_i N_{i,n}(x_J) = y_J \qquad J = 1,2,\ldots,L \tag{5.16}$$

That is to minimise

$$\sum_{J=1}^{L} (S(x_J) - y_J)^2$$

(5.16) may be written in matrix notation as

$$A C = Y \tag{5.17}$$

where A is the $J \times (m+n)$ matrix whose element in column i row J is $N_{i,n}(x_J)$, and C and Y are the column vectors with elements $C_i$ and $y_J$ respectively.

In program "BSLSQ" for least squares B-spline curve fitting, the knots are chosen arbitrarily within the range of the data ( $\lambda_1 = x_1$, $\lambda_m = x_L$). The selection of most suitable set of knots and the number of intervals is a matter of trial and error, experience and general knowledge of the shape of the curve as indicated by the data. BSLSQ is written in the form of a series of nested subroutines. x and y coordinates of the data points, knot set, and the order of curve to fit the data are input to subroutine "COEFF" which calculates coefficients $C_i$. Now to evaluate the B-spline curve at certain point, x, subroutine "FUNCAL" is called. The calculation of basis functions is carried out in a separate subroutine which is used by both COEFF and FUNCAL.

To solve simultaneous equation system of the least squares problem, Cholesky's elimination method was adopted; because it is the most economical in terms of the number of arithmetic operations and hence, fastest of the basic elimination methods (70).

It should be noted that only n elements in each row of the matrix A are non-zero and these are adjacent. Moreover, if the data points are arranged in order of increasing x, the first of these n elements in any particular row will never lie further to the left than that of previous row. The structure may be described as band structure and advantage is taken from this property in simplifying the computer program.

Defining $\lambda_i$'s at equal intervals, the coefficients $C_i$ give the y values for defining polygon vertices at points $\lambda_i$ in a two-dimensional cartesian system which could later be used as input to surface design program.

### 5.4.2. Implementation of Inversion Technique in Curve Fitting

The problem is to establish the location of polygon vertices for which a B-spline approximation will pass through the data points. If the data is a set of selected points on an existing curve, the closeness of the approximated curve to the original depends on the number and position of the points selected.

In implementation of this technique, to simplify the procedure, again the uniform B-spline method as discussed in Section 5.4.4.1. was adopted. The user can interpolate B-spline curves of different order to the data points. A parametric method is employed to allow defining polygons to non-planar curves as well as closed curves to be determined.

The number and the coordinates of the data points, and the order of the curve are input to subroutine "INVERT" within which coordinate values of the vertices are calculated. To evaluate B-spline bases, and the nodes ($\xi_i$) a set of uniform knots must be defined, the number of the data points and the order of the curve are here the determining factors. The number of the vertices evaluated with this method is equal to the number of selected input points whereas in a least squares fit one can determine the number of the vertices (coefficients $C_i$) independent of the number of input points.

## 5.5. Surface Design Procedure

Recalling Section 5.2.5. a cardinal spline curve can be defined by

$$S(x) = \sum_{i=0}^{N} f_i A_i(x)$$

where $f_i$, ($i=0, \ldots\ldots. N$) are the ordinates and the $A_i(x)$ the cardinal spline basis functions.

In a parametric form with parameter v

$$S(v) = \sum_{i=0}^{N} f_i A_i(v)$$

Now let $B_i(u)$ be the parametric representation of the $i^{th}$ sectional B-spline curve in three-dimensional space. At a parametric position; say $u_J$, the value of curve $B_i(u)$ is shown as $B_i(u_J)$. Replacing $f_i$ with $B_i(u_J)$ one will have

$$S_{u_J}(v) = \sum_{i=0}^{N} B_i(u_J) A_i(v)$$

Denoting $S_{u_J}(v)$ symbolically by $Q(u_J, v)$ and generalising the case for all values of u, one may write

$$Q(u,v) = \sum_{i=0}^{N} B_i(u) A_i(v) \qquad (5.18)$$

Here u has the same meaning as used in the definition of the B-spline curves; that is, each sectional curve is a function of parameter u at a constant value of v.

$$Q(u, v_i) = B_i(u)$$

It is obvious that (5.18) respresents a two-dimensional parametric space or a surface which is the result of interpolating the sectional curves using cardinal spline basis functions. Because the cardinal spline curves have the characteristic property that the spline contains the controlling points, the surface will pass through all the sectional curves.

The continuity of surface derivatives depends on the degree of the curves used in defining sectional curves and the degree of splines that interpolated them.

It should be noted that there is no requirement for the sectional curves to be parallel or planar.

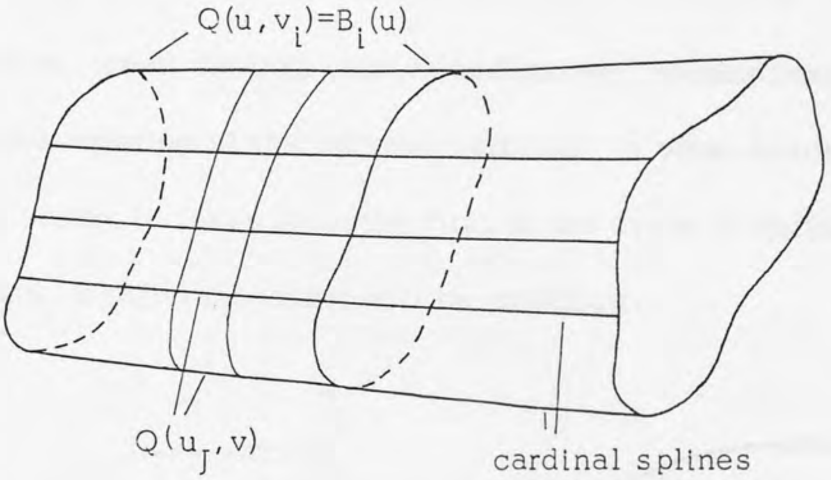


Fig. (5.3) Surface Description Method

If points $((u_J, v_i), J = 1, n \;, i = 1, m)$ are generated at equal parametric distances using Expression (5.18) and directly used in graphics displays, it may lead to a visually misleading representation of the surface. This is due to the fact that location and the number of defining polygon vertices are the major factor in parametrisation of sectional B-spline curves. As a result, equal parametric distances on a curve may produce completely uneven distribution of points at varying arc lengths, Fig. (5.4).

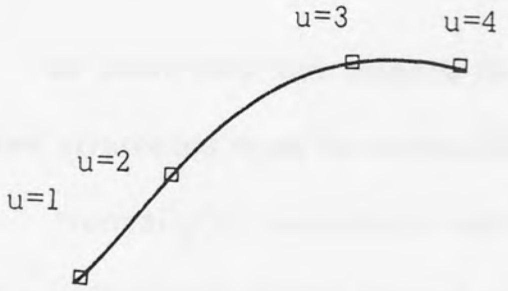


Fig. (5.4) Equal Parametric Distances with Uneven Distribution

Moreover, when dealing with closed curves, parametrisation depends upon the ordering of the polygon vertices, in other words, the fact that which vertex is taken to be the first in the cycle of defining polygon vertices, a different result will be produced.

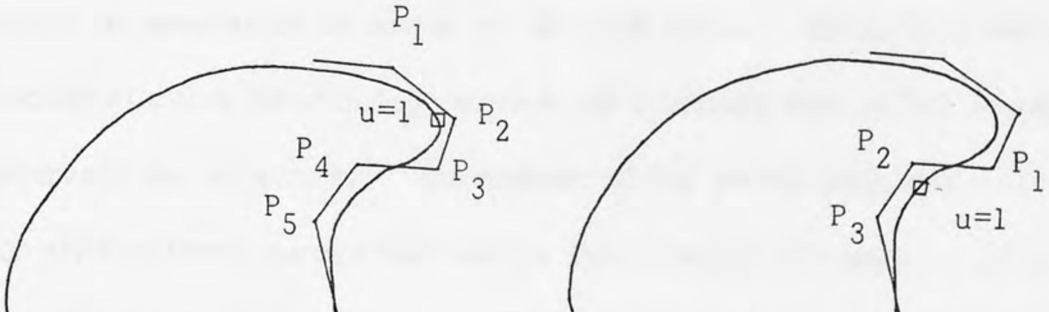


Fig. (5.5) The Effect of Vertex Numbering on Parametrisation

Therefore, over a set of sectional curves, points identified by certain parametric value $u_J$ could be non-planar, or in extreme cases lie at opposite sides of the surface. The cardinal splines joining these points, Q ($u_J$, v) may twist on the surface and hence, when used for generation of graphical displays produce a wrong impression of the shape.

To avoid this shortcoming of parametric representation, alternative strategies must be employed in graphical displays of the surface. Normally to represent a surface graphically, heights at the nodes of a sufficiently finely spaced regular rectangular grid of points in (x,y) plane is produced and then lines joining the adjourning points are isometrically projected. In an acceptable drawing of the surface, projected lines on the surface must be at least roughly equally spaced, but the isoparametric lines do not normally satisfy this requirement .

The scheme adopted in BCSURF aims not only at overcoming the above problems, but is also intended to simplify the use of the result in generation of cutter centre-line path. Using this method each sectional curve after being defined is digitised and points at equal arc intervals are selected. The number of the points selected is the same for all sectional curves and can be specified by the user. Cardinal splines are then generated in the other direction so that to interpolate the selected points on the contours. The cardinal spline curves are also divided into equal arc intervals and intermediate contours are produced. The final result therefore, consists of a mesh of points that can be used for generation of cutter centre line path, or joined together and projected to produce drawings of the shape. In the BCSURF package the routine BCSURF based on the method described is included for surface design.

## 5.6 Justification of Approach

The criteria on which the choice of a mathematical representation in Computer Aided Design is based, vary, depending upon the particular application. Most applications for curve and surface design, however, have certain features in common, such as, simplicity, flexibility and means of enforcing continuity of derivatives in the surface. Ideally a CAD system should require the designer very little or no knowledge of the mathematical formulation used, and yet should enable him to control the shape of curves or surfaces by control parameters that it offers. The modification of curves and surfaces must be easy and the effects of modification should be predictable.

The major obstacle in the problem of three-dimensional surface design is man-machine communication. At present most problems of 3-D interactions are handled by typing the command on the key board. Although inputting limited amounts of three-dimensional data points is possible, it is very difficult to input a curve in three-dimensions with the available hardware equipment. In some systems positional data is specified by a tablet stylus or a light pen, but these mainly deal with 2-D or orthographic and perspective projections of 3-D objects. Attempts have been made to use non-standard devices, such as head mounted displays (71, 72) and 3D-input devices or "wand" which enable the designer to "walk about" in 3-D environment containing 3-D objects, modifying them with the wand; but these are in early experimental stages.

As reviewed in Chapter three, among the more important surface design methods, Ferguson and Coons patch design, and Bezier and B-spline surface can be mentioned. Ferguson's and Coon's method do not provide the necessary intuition for the user and he is not able to predict readily the effect of changing the defining inputs on the final shape. More advanced Bezier and B-spline surface approaches allow modification to be carried out, but these also require interaction in three-dimensional space. Moreover, even if effective manipulative communication were available, it would be difficult to use in comparing the resulting approximation to the existing shape due to the lack of information regarding the original shape definition.

Lofting techniques by-pass many difficulties involved in communicating three-dimensional data, because curves are taken to be the basic data elements for surface design. Limiting ourselves to planar sections, the available hardware devices can efficiently be used to input 2-D data and to manipulate the resulting curves. In addition to achieving better manipulative control over the resulting shape, this method also offers greater potential for comparison of approximations to the original.

With lofting method of surface descritpion it is the manipulative power of the curve design procedure that lie at the centre of attention. The shape of real objects in general is non-analytic in nature and local conditions do not normally have an overall effect on the shape. For this reason most mathematical representations of curves and surfaces so far devised are based on piecewise methods. As design proceeds by a series of local changes, analytic functions do not have the desired properties, whereas piecewise functions appear to provide satisfactory solution. The classical piecewise methods of curve construction lack the required flexibility and are difficult to be implemented in computers. For example, using Hermitian method, curve segments can be interpolated through two end points. The composite curve is then a combination of all the segments joined with continuity of specified derivatives at the joints. The draw back of such schemes is that the effect of end erivatives is too obscure to be visualised and the process is very tedious.

Using Bezier method, though construction of piecewise curves is easier and local modification can be carried out by changing the position of polygon vertices, the order of the curve increases proportionally to the number of polygon vertices used.

The most desirable properties in computer aided curve design is perhaps offered by the B-spline representation of curves, because:

i) The shape of the curve can easily be manipulated by control points.

ii) A given control point only affects a limited portion of the entire curve (non-global characteristic).

iii) B-spline representation has built-in, or intrinsic derivative continuity across adjacent segment boundaries.

In addition, the order of the curve can be chosen according to the purpose of application and is not, as in Bezier curve, dictated by the complexity of the defining polygon shape. Finally, the computational aspect of the B-spline seems more efficient than all other schemes which is a major factor, particularly in interactive design situations.

The above reasons providéd sufficient grounds for adoption of lofting technique in surface representation, and the application of B-spline for cross-sectional curve design at the present work.

The choice of cardinal splines to interpolate cross-sections in forming the surface is also justifiable due to the fact that, firstly it provides a simple procedure for interpolation which is computationally efficient both in terms of memory requirements and speed, and secondly, the interpolated curve in most practical cases is free from unwanted oscillations.

## 5.7. Machining

To integrate the design and manufacture, the two processes must be linked so that the result from the design stage can easily be passed into the manufacturing system. In addition,the result on shape definition should be transformed into a format that is acceptable to the manufacturing system and that the manufactured product only slightly varies from the shape designed.

Machining in computer aided manufacturing systems is normally carried out using Numerical Control machines. Although some of these machines are capable of producing curved contours, in the majority of cases, particularly cutting surfaces, machining is accomplished by moving the cutter along straight lines. Now,whether cutting process is in point to point fashion or linear interpolation is used, point data is required. Therefore, to cut the surfaces defined by analytic equations, or developed from incomplete boundary specifications, or otherwise, the data introduced to the manufacturing unit should be in form of coordinates of a set of points on the surface.

In cutting free form surfaces one is faced with the problem of cutter offset, accuracy, and the so called "cutter interference". Customarily ball-nose cutters have been used in these circumstances and the calculation of cutter offset is achieved by finding the direction of normals on the surface. This can be accomplished either analytically or numerically. Having determined the direction of normals, the cutter is then offset by the radius of the cutter along this vector over the surface.

### 5.7.1. Analytical Method of Computing Normals

Given Q(u,v) to represent a surface, the tangent vectors of a point on the surface can be calculated by differentiating this equation with respect to the surface parameters u and v.

If $Tu_{iJ}$ and $Tv_{iJ}$ are taken to be the tangent vectors at point $(u_J, v_i)$

$$Tu_{iJ} = \left[ \frac{\partial Q}{\partial u} \right]_{\substack{v=v_i \\ u=u_J}}$$

and $$Tv_{iJ} = \left[ \frac{\partial Q}{\partial v} \right]_{\substack{v=v_i \\ u=u_J}}$$

The direction of normal at this point is obtained by taking the cross product of these two vectors.

$$TN_{iJ} = Tu_{iJ} * Tv_{iJ}$$

Assuming that the ball-nose cutter used has the radius R, cutter must be offset by $TN_{iJ} . R$ .

If the same process is repeated in a consistent sense for every point on the surface, in parametric representation the cutter centre positions can be given by

$$CQ\ (u,v) = Q\ (u,v) + TN.R \qquad (5.19)$$

where TN is the normal vector at point (u, v).

Partial differentiation of surface equation as given by Expression (5.18) with respect to parameter u and v provides

$$Tu = \frac{\partial Q}{\partial u} = \sum_{i=o} \dot{B}_i\ (u)\ A_i\ (v) \qquad (5.20)$$

$$Tv = \frac{\partial Q}{\partial v} = \sum_{i=0} B_i(u)\,\dot{A}_i(v) \qquad (5.21)$$

Both $\dot{B}_i(u) = \frac{d\,B\,(u)}{d\,u}$ and $\dot{A}_i(v) = \frac{d\,A\,(v)}{d\,v}$ are easy to find. If an n order B-spline curve is constructed over the knot set $x_o, x_1, x_2, \ldots$ and the defining polygon vertices are $P_o, P_1, P_2 \ldots, Pm$, de Boor (66) shows that the $J^{th}$ derivative can be found using the following Expression :

$$S^{(J)}(u) = (n-1) \ldots\ldots (n-J) \sum_i P_i^{(J)} N_{i,n-J}(u) \qquad (5.22)$$

where $P_i^{(J)}$ is defined as

$$\left[\begin{array}{l} P_i^{(0)} = P_i \\ \\ P_i^{(J)} = (P_i^{(J-1)} - P_{i-1}^{(J-1)}) \;/\; (x_{i+n-J} - x_i)\;, \qquad J > 0 \end{array}\right.$$

and $N_{i,n-J}(u)$ is the (n-J) order B-spline basis function at interval $x_i < u < x_{i+1}$.

The first derivative of an n order uniform B-spline curve with knots at integers can therefore be given by

$$\dot{B}\ (u) = S^{(1)}(U) = (n-1)\sum_{i=1}^{m} (P_i - P_{i-1})\, N_{i,n-1}(u) \qquad (5.23)$$

While computing the B-spline basis functions $N_{i,n}(u)$ for evaluating the B-spline curves using the recursive formula (5.2), $N_{i,\,n-1}(u)$ are automatically determined and it is only necessary to save them for future use.

Evaluation of the cardinal spline basis functions derivatives are equally simple, and one only needs to differentiate Expressions (5.12) and (5.13). However, it is also possible to differentiate Expression (5.14) which embodies both (5.12) and (5.13), and determine the first derivative of the cubic cardinal spline, which in conjunction with (5.23) enable the computation of tangent vectors. This is the method applied in the routine BCSURN in order to simplify the procedure. The tangent vectors Tu and Tv after being determined, are normalised and their cross products computed to obtain the direction of the normals on the surface.

### 5.7.2. Numerical Method of Calculating Tool Offset

Unlike analytical methods in which cutter offsets at the data points are calculated, here three of these points are joined to form a triangular plane, a group of which form a multi-faced polyhedron that approximate the desired shape. The approximate machining of the surface is obtained by directing a spherically ended milling cutter to touch in turn all the triangular facets on the surface. Points of contact are at the centroid of the triangles and slightly within the surface, but to a lesser extent as the vertices are more closely spaced.

Considering triangular plane facet ABC, such plane has the general equation

$$a x + b y + c z + d = 0$$

The coefficients a, b, c, d are found knowing the coordinates of points A, B, and C. The direction cosines of the normal to this plane are easily found by dividing a, b, c, and d by $(a^2 + b^2 + c^2)^{\frac{1}{2}}$ to yield, say, L, M, N, and P. Plane A B C represented in its perpendicular form would be

$$L x + M y + N z + P = 0$$

where P is the negative of the perpendicular distance of plane ABC from the origin.

Referring to Fig. (5.6) if c is the centroid of plane ABC and T a point R from c on the normal to the plane through c, then

$$x_T = x_c + L \,.\, R$$

$$y_T = y_c + M \,.\, R$$

$$z_T = z_c + N \,.\, R$$

which gives the coordinates of the centre of a spherical-ended tool of radius R with its centre at T and tangential to the plane ABC at its centroid. BCSURF includes a routine in which the numerical method of computing tool offset as discussed above is implemented. As most components require checks to be carried out on the cutter interference, the calculation of tool offsets and interference checking are combined in one routine which will be discussed in the following Sections.

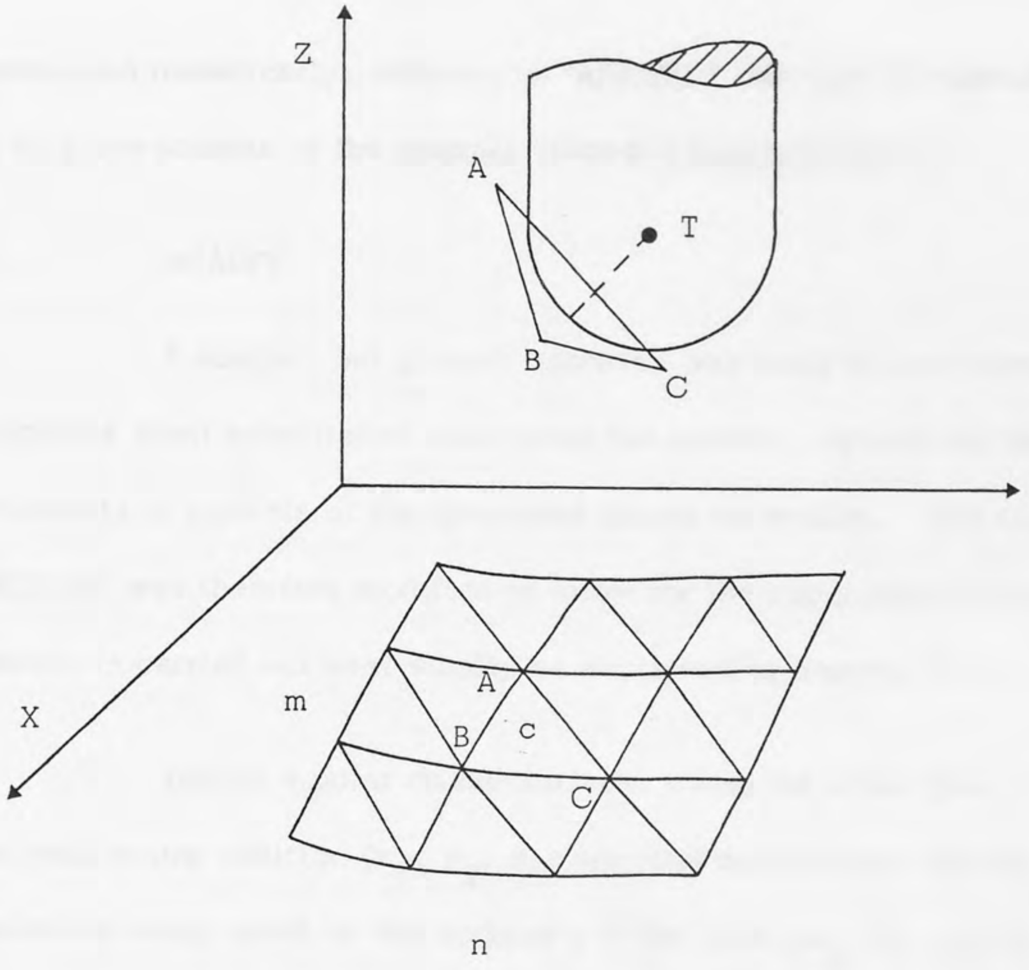


Fig. (5.6) Tool Offset on a Multi-faced Polyhedron

## 5.8. Interference Checking

The object in interference checking is to see if the spherical ended cutter used for machining while cutting to a specified position to produce the surface will remove any part of the adjourning area. For a smooth surface the largest feasible tool should be used, but if the surface is planar or concave no interference will occur. Otherwise the size of the tool must be reduced to avoid it.

Two programs as described below have been developed to accomplish interference checking. In "NUMOFF" cutter offset is computed numerically, whereas in "ANAOFF" this task is carried out using components of the normals produced independently.

### a. ANAOFF

A simple, but general approach was used in this routine. It requires point coordinates specifying the surface, as well as the elements of normals at the generated points as inputs. The routine BCSURF was therefore modified to allow for the computation of normals which is carried out analytically as explained in Section 5.7.1.

Taking a point on the surface, using the input data, coordinates of tool centre position ($x_T$, $y_T$, $z_T$) are first determined, and the distance between every point on the surface and the point ($x_T$, $y_T$, $z_T$) is then calculated. If this distance is smaller than (R - Tolerance), the tool is raised to the new $z_T$ position to avoid interference, and the test similarly continues for the rest of the points. Having finished the check with the largest available tool, the next largest tool is taken and the same procedure follows until either there is no interference or the smallest size tool set by the user has been considered.

This routine also incorporates segments that generate the NC machine formatted part program using the interference checked cutter centre position data. All the cutters employed, following the parametric lines joining the data points, travel over the component and one after another produce a better finish of the surface.

### b. NUMOFF

A more comprehensive procedure of interference checking was adopted in this routine. It uses coordinates of points generated on the surface by the surface design program BCSURF, though this data could of course have originated from other sources, such as being the result of digitisation of a physical model for the purpose of replicating an existing object. The mathematics employed is based on algebraic geometry and extensively discussed by Duncan and Mair (37, 73). The details of mathematical theory used is not explained here, but only very briefly the function of the program discussed.

Program NUMOFF reads in the point data which are organised in a matrix form, the radius of the cutters used (up to eight tools starting from the largest available size), and control parameters. From the data it constructs the triangular planes taking points from adjoining rows and columns. The program incorporates a control switch which by setting, diagonals are drawn in the opposite direction (see Fig. 5.7).

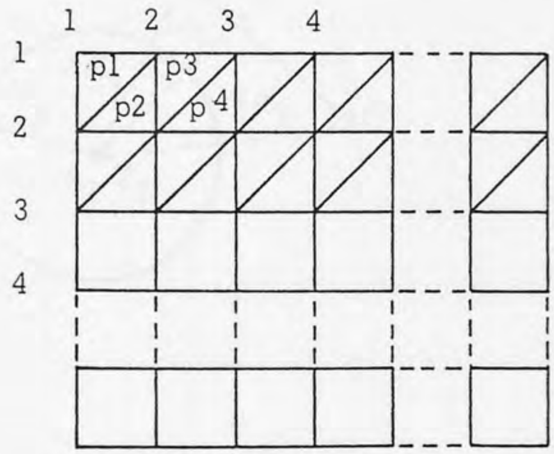


Fig. (5.7) Data Point Organisation for NUMOFF

From the three vertices of each triangle is derived the equation of the plane $a x + b y + c z + d = 0$. A tool centre position $(x_T, Y_T, Z_T)$ is taken at a distance R = tool radius along the normal at the centroid $(x_c, y_c, z_c)$ of the triangle, program then extracts the indices of all the planes that lie within the radial distance of the tool at position $(x_T, y_T, z_T)$. Let (i, J) be the row and column of plane under consideration. The distance of all three vertices of the neighbouring plane (i, J + 1) from the point $(x_T, y_T)$, i.e.

$$(\Delta v)^2 = (x_T - x_v)^2 + (y_T - y_v)^2$$ is computed and the minimum taken. If this distance is less than the tool radius, plane (i, J + 1) is within the radial distance of the tool. Planes (i, J + 2), (i, J + 3), .... are also tested, the process terminating as soon as one plane is farther than R from $(x_T, y_T)$. Similarly (i, J-1), (i, J-2), ......, and planes in upper and lower columns (i-1, J), (i-2, J), ......, (i + 1, J), (i + 2 J), ..... are tested and the indices for all the planes threatened with under cutting are saved.

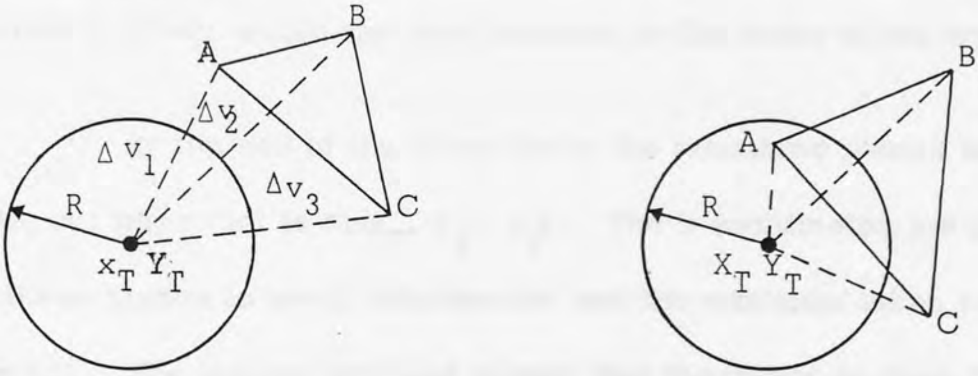


Fig. (5.8) Examination of Planes within the Radial Distance of the Tool

The next step is to eliminate planes that are coplanar or lie below the plane being examined. This is carried out in the following manner. If the plane under examination is represented in its perpendicular form by $L x + M y + N z + P = 0$,substituting x, y, and z by xl, yl, zl the right-hand side of the above expression produces the distance of the point (xl, yl, zl) from this plane. Now one can substitute $x_T$, $y_T$, $z_T$, and then the coordinates of the three vertices of the planes from previous tests suspected of being undercut in the above expression. If all the three latter results are zero or of opposite signs to that of the first, then that plane is coplanar or lies below the plane under examination. The indices of the planes not eliminated are saved for further verification.

More planes previously suspected are eliminated by testing the normal distance from the ($x_T$, $y_T$, $z_T$) to each of these planes in turn. If it is greater or equal to the radius less a tolerance specified by the user, there is no interference. Finally, it must be determined whether the tool cuts the planes which has so far not been eliminated and interference is likely within the area bounded by the sides of the triangle.

By the end of the above tests the remaining planes will be under cut when tool is at ($x_T$, $y_T$, $z_T$). The z coordinates are found for all these planes to avoid interference and the maximum taken to be the new $z_T$. The indices of those planes that the cutter on their facets will produce interference are saved for later processing by a smaller tool. This procedure now continues until either the number of saved indices are zero or the smallest tool provided has been examined.

# CHAPTER SIX

# RESULTS

## 6.1. Machining in the DNC System

### 6.1.1. Interpolation

The term "interpolation" in the context of numerical control means the ability of the machine control unit to obtain intermediate coordinate values lying between the start and the end of a line. In other words, it means producing contours using discrete slide movements (74). If the segments joining the points are straight lines, the process is called linear interpolation; if they are arcs of circles or parabolas, it is known as circular or parabolic interpolation.

Most contouring machines have only the capacity to cut straight line segments. Now depending upon the complexity of the machine,these straight line segments can vary from being along the machine axes only (1D machines), to lines in X-Y plane (2D), and finally in three-dimensional space (3D). This basic ability can be exploited and by programming the capability of the machine extended to interpolate linear or curve segments.

The interpolation can be accomplished either manually, or with the help of a computer. In practice, even the simple contours require many calculations to be made, making it impracticable to be carried out with manual programming, but can easily be done with the aid of a computer.

In any interpolation scheme as the contour is approximated by segments, the stress is put on reducing the error so that it lies within the required tolerance. Therefore, formulation used in interpolating points inevitably contains factors relating to the allowable error.

### 6.1.1.1. Linear Interpolation of Straight Lines

The Cincinnati 3VT-1000 is a $2\frac{1}{2}$ axis machine; that is, the movement of the cutter in relation to the work table can either be in X-Y plane or in Z direction independently. In order to produce a continuous path in three-dimensional space therefore requires that the cutter be moved stepwise. From a point at X1, Y1, Z1 to X2, Y2, Z2, the movement of a square end cutter can take one of the three forms depicted in Fig. (6.1).

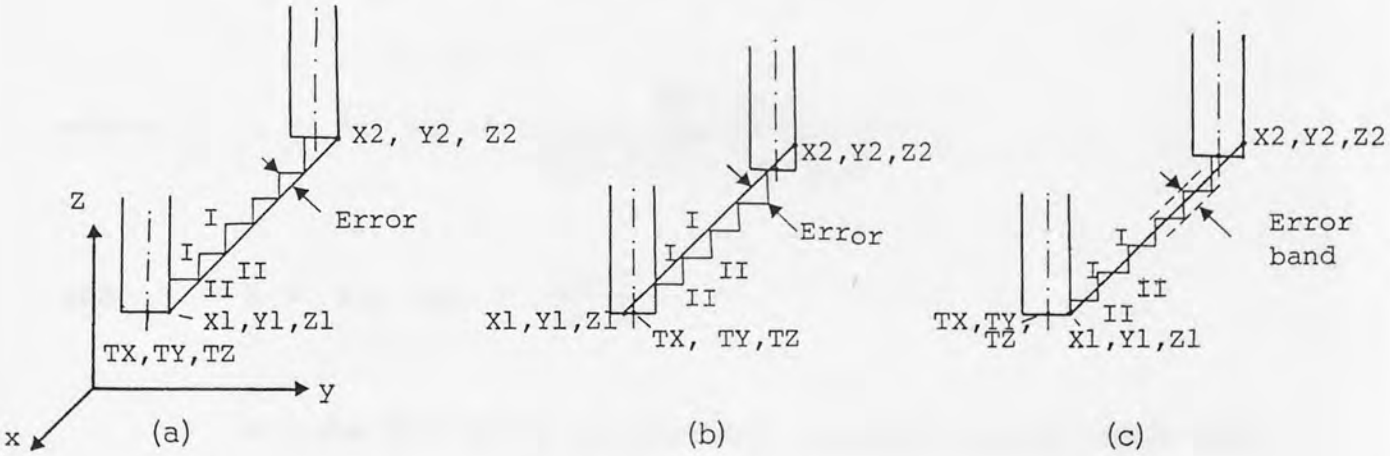


Fig.(6.1) Linear Interpolation of Straight Lines

With method (a) there will be a positive (+E) and with method (b) negative errors (-E). The more general form of interpolation is the method (c) where an error band is resulted. If the number of steps to produce a contour with method (c) is taken to be equal to the number of steps with method (a) or (b) on a similar path, the error of the surface finish will reduce to ± E/2.

If X1, Y1, Z1 and X2, Y2, Z2 are the cordinates at the beginning and at the end of a segment, the coordinates of any two successive points (II) and (I) on the interpolated path can be shown as

$$X_I = X_{II} \text{ (last)}$$

$$Y_I = Y_{II} \text{ (last)}$$

$$Z_I = Z_{II} \text{ (last)} + E/\mathrm{Cos}\,(\alpha)$$

and

$$X_{II} = X_{II} \text{ (last)} + E \cdot \mathrm{Cos}\,(\beta)\ /\ \mathrm{Sin}\,(\alpha)$$

$$Y_{II} = Y_{II} \text{ (last)} + E \cdot \mathrm{Sin}\,(\beta)\ /\ \mathrm{Sin}\,(\alpha)$$

$$Z_{II} = Z_{II} \text{ (last)} + E/\mathrm{Cos}\,(\alpha)$$

where $\alpha = \mathrm{Arc\ tan}\left( \dfrac{Z2 - Z1}{\sqrt{(X2-X1)^2 + (Y2-Y1)^2}} \right)$

and $\beta = \mathrm{Arc\ tan}\left( \dfrac{Y2-Y1}{X2-X1} \right)$

With the tool radius compensation applied, the tool centre path for a tool having the radius R can be shown as below:

$$TX_I = X_{II} \text{ (last)} - R \cdot \mathrm{Cos}\,(\beta)$$

$$TY_1 = Y_{II} \text{ (last)} - R \cdot \mathrm{Cos}\,(\beta)$$

$$TZ_I = Z_{II} \text{ (last)} + E/\mathrm{Cos}\,(\alpha)$$

and

$$TX_{II} = X_{II} \text{ (last)} + E \cdot \mathrm{Cos}(\beta)/\mathrm{Sin}\,(\alpha) - R \cdot \mathrm{Cos}\,(\beta)$$

$$TY_{II} = Y_{II} \text{ (last)} + E \cdot \mathrm{Sin}(\beta)/\mathrm{Sin}(\alpha) - R \cdot \mathrm{Sin}(\beta)$$

$$TZ_{II} = Z_{II} \text{ (last)} + E/\mathrm{Cos}(\alpha)$$

To achieve any of the three methods discussed, referring to Fig. (6.1) it is only necessary to calculate the corresponding coordinates of the first "II" point on the path ($X_{II}$, $Y_{II}$, $Z_{II}$) and initiate cutting from position $TX_{II}$, $TY_{II}$, $TZ_{II}$. Therefore, by taking advantage of the calculating power of the computer on line with the NC machine, software interpolation makes it possible to use the system as a three axis machine. For this purpose, sub-programs were developed that on being called generate part program for machining such paths.

### 6.1.1.2. Linear Interpolation of Curves

Generally there are three types of linear interpolation for curved contours:

a) Chord Method

b) Tangent Method

c) Secant Method

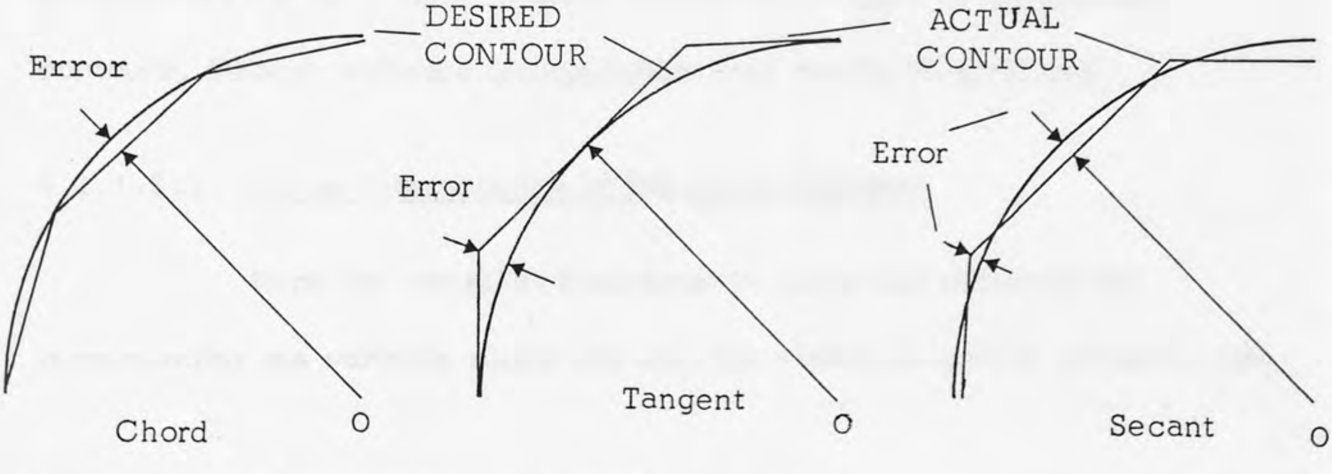


Fig. (6.2) Linear Interpolation of Curved Contours

In interpolating circular arcs, the accuracy of the arc produced depends on the length of the line segment and the radius of the arc. The method used in practice must depend on the contour being cut and the characteristics required from the component.

Although development of a general formula is possible, the mathematics of interpolation of curve segments largely depends on the type, mathematical representation of the curve and the method employed. Of the most encountered curves in engineering applications, conics and in particular, circular and elliptical arcs can be mentioned. To establish practicability of producing non-linear contours, using linear interpolation in the system, programs were developed and circular and elliptical grooves as shown in Fig. (6.5) were machined.

The control unit of the Cincinnati machining centre was already incorporated with hardwired interpolater for producing circular arcs. However, the limited software control achieved in this system gives a new dimension to the capabilities of the NC machine and clearly demonstrates the merits of software orientated controller. Taking the interpolation here as an example, not only circular arcs, but all other types of curves and contours, through software manipulation, can easily be produced.

### 6.1.1.2.1. Linear Interpolation of Elliptical Contours

Here the variation from point to point was achieved by incrementing one variable along one axis for a certain portion of the ellipse and incrementing the next variable along the next axis for the next defined position. The process then goes on until the finish point is reached.

As shown in Fig. (6.3), the complete ellipse is divided into four regions by a pair of conjugate diameters. Each region embraces a length of arc, namely AB, BC, CD and DA.

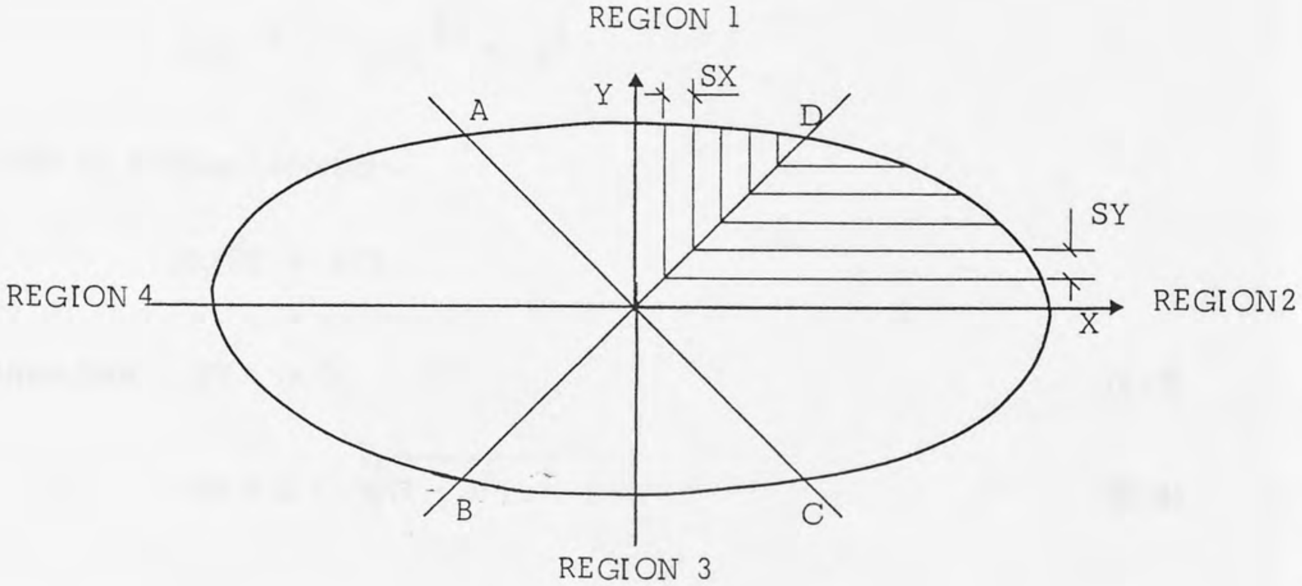


Fig. (6.3)

Expressing the ellipse in the form $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ ;

the equations used for different regions are as follows :

$$X' = X \pm SX \tag{6.1}$$

$$Y' = \pm b \sqrt{(1 - (X'^2/a^2))} \quad \text{for regions 1 and 3}$$

and

$$Y' = Y \pm SY \tag{6.2}$$

$$X' = \pm a \sqrt{(1 - (Y'^2/b^2))} \quad \text{for regions 2 and 4}$$

The sign of the increments depends on the region the calculated point lies.

### 6.1.1.2.1.1. Determination of Tolerance

In determination of error, the maximum error that can occur is calculated. Thus the error at points A, B, C and D, where the longest chords lie is considered.

From Fig. (6.4.a)

$$(SX)^2 + (SY)^2 = L^2$$

also by simple majority:

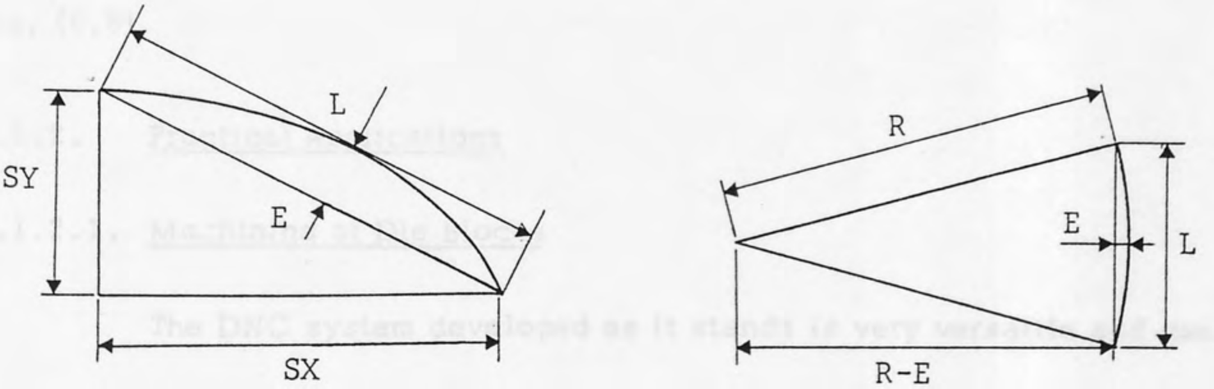
$$SX/SY = a/b$$

therefore

$$SX = (a/b) \cdot SY \tag{6.3}$$

$$SY = L / \sqrt{(1 + a^2/b^2)} \tag{6.4}$$

In order to find the relation for SX, SY and L, E and other known variables, it can be assumed that any small portion of an ellipse is very nearly a circular arc, (Fig. 6.4.b).



(a) (b)

Fig. (6.4)

From Fig. (6.4.b)

$$L/2 = \sqrt{R^2 - (R-E)^2} \qquad \therefore \; L = 2\sqrt{E\,(2R-E)}$$

where R is the radius of the ellipse at the position being studied.

Hence, SX and SY from (6.3) and (6.4) become

$$SX = (a/b)\,(2\sqrt{E\,(2R-E)})/\sqrt{(1 + a^2/b^2)} \tag{6.5}$$

$$SY = (2\sqrt{E\,(2R-E)})\,/\,\sqrt{(1 + a^2/b^2)} \tag{6.6}$$

### 6.1.1.2.1.2. Machining of the Test-Piece

By substituting (6.5) and (6.6) in (6.1) and (6.2) respectively, a general formula for calculation of intermediate points for chordal interpolation of ellipse is found. The program written for this job requires that the major and minor axes of the ellipse (a, b), maximum tolerance (E), coordinates of the centre of the ellipse, start and finish points of the contour and some control variables to be input. Taking the value of "a" equal to "b" a circular arc is produced. The result of this exercise is shown in Fig. (6.5).

### 6.1.2. Practical Applications

### 6.1.2.1. Machining of Die Blocks

The DNC system developed as it stands is very versatile and can be applied to machining various complicated shapes. One of the areas it

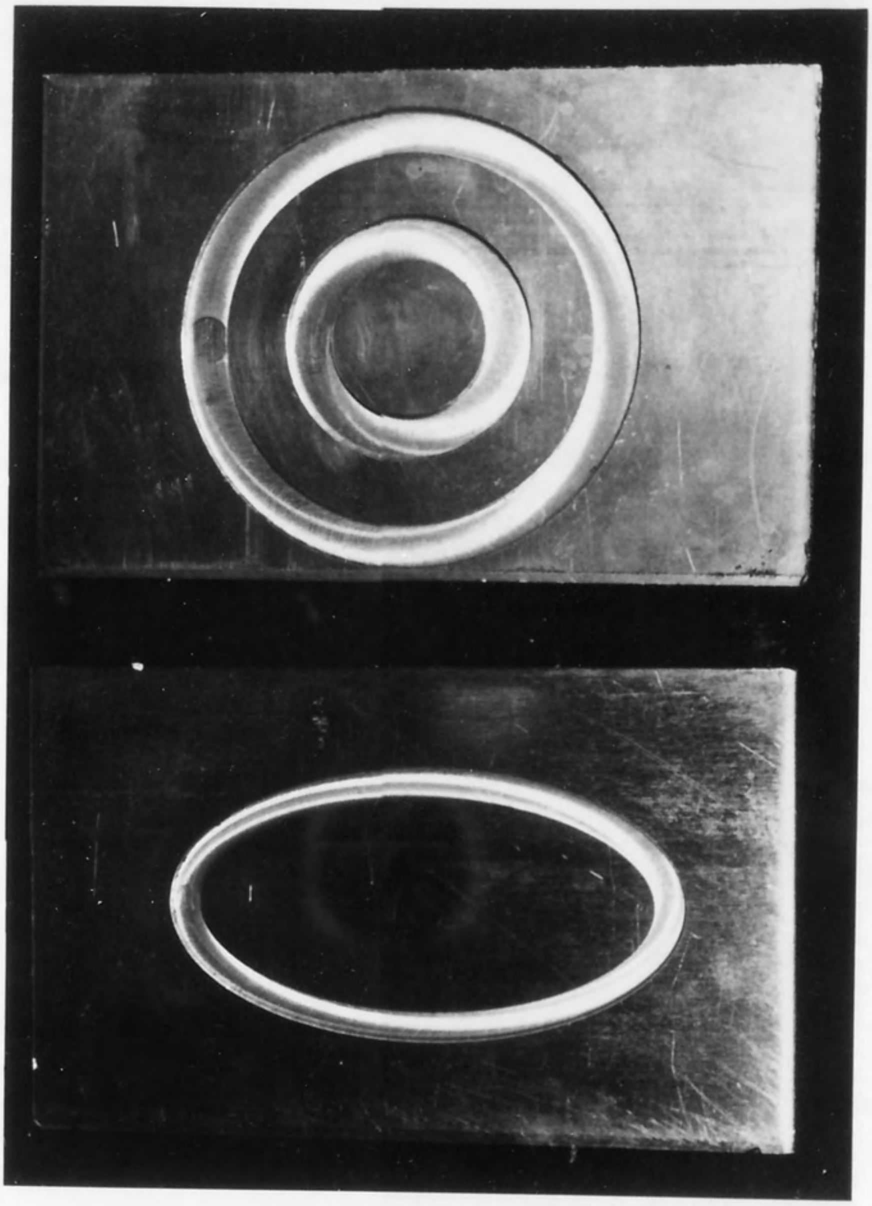


Fig. (6.5) Conic Curves Produced in the System best suited is the manufacture of injection moulds. Components of this type that have been cut in this system include a die block for injection moulding of a fork shaped component, die block for injection moulding of plastic handles for kitchen utensils and a block used to control the flow in extrusion of plastic profiles.

The drawing of the "Fork" and the die required is shown in Fig. (6.6) and as can be seen in Fig. (6.6) the die consists of two deep walls, and some circular shapes and fillers. Moreover, the deep wall changes in depth in some parts of the circular arc, and this is a complex shape to be produced.

The block machined for pre-extrusion flow path control also mainly consists of slant surfaces and inclined paths. Generally, the cutting of such parts on milling machines would be by the use of special fixtures so that the whole block can be tilted at various stages during the cutting operation, or by the use of NC machine tools with continuous path control having three or four axes . In this system, software manipulation, however, enables the inherent shortcomings of the NC machine to be overcome and complicated components manufactured.

As explained earlier, linear interpolation has widely been used in solving many problems for generation of different contours. In part programs for NC machines each step occupies one line of the program, and therefore, for high quality machining in which low tolerance is required, the number of the blocks of information needed to cut parts such as those

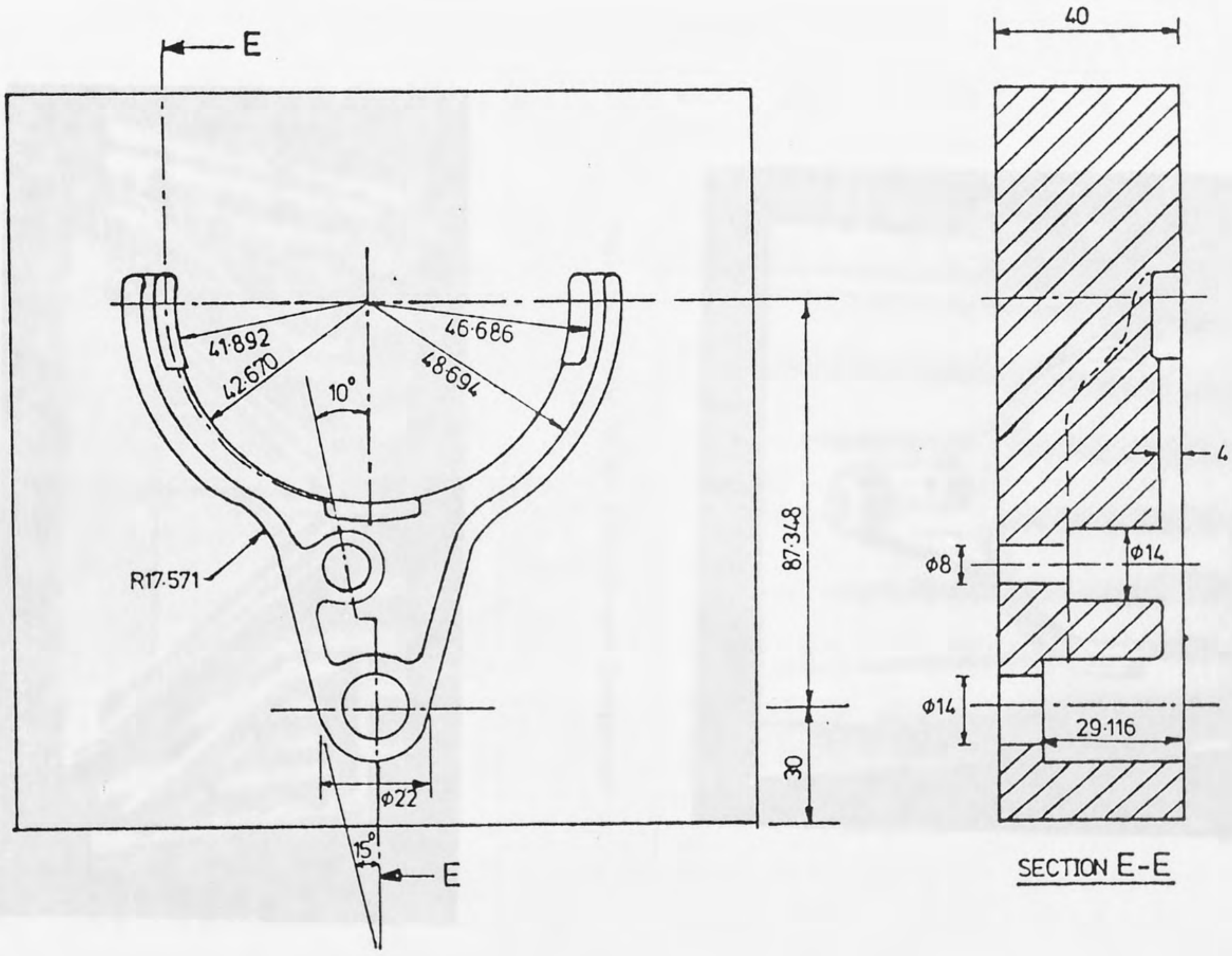


Fig. (6.6) Drawing of the Fork Shaped Die

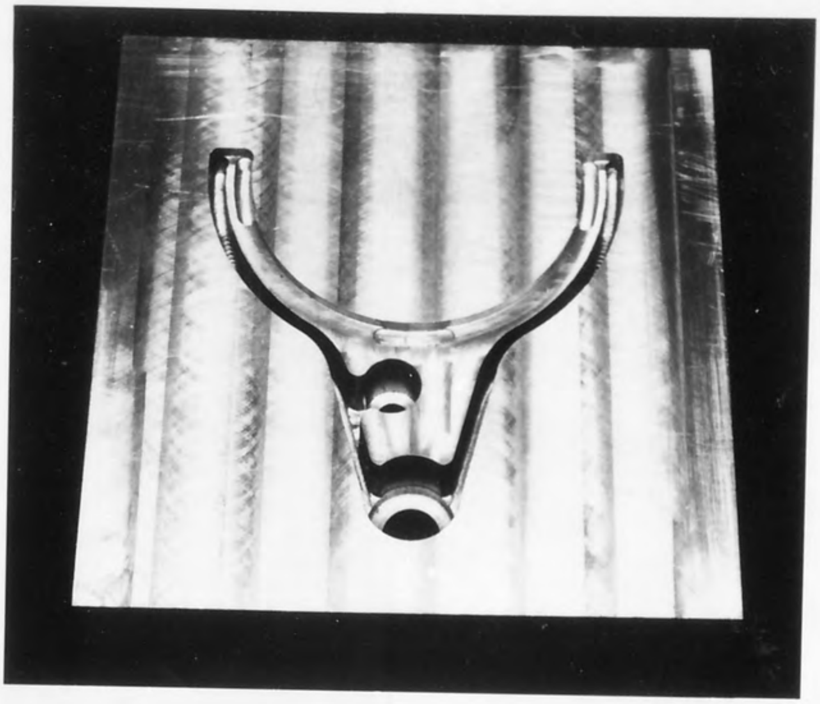


Fig. (6.7) Die for the Fork-Shaped Component

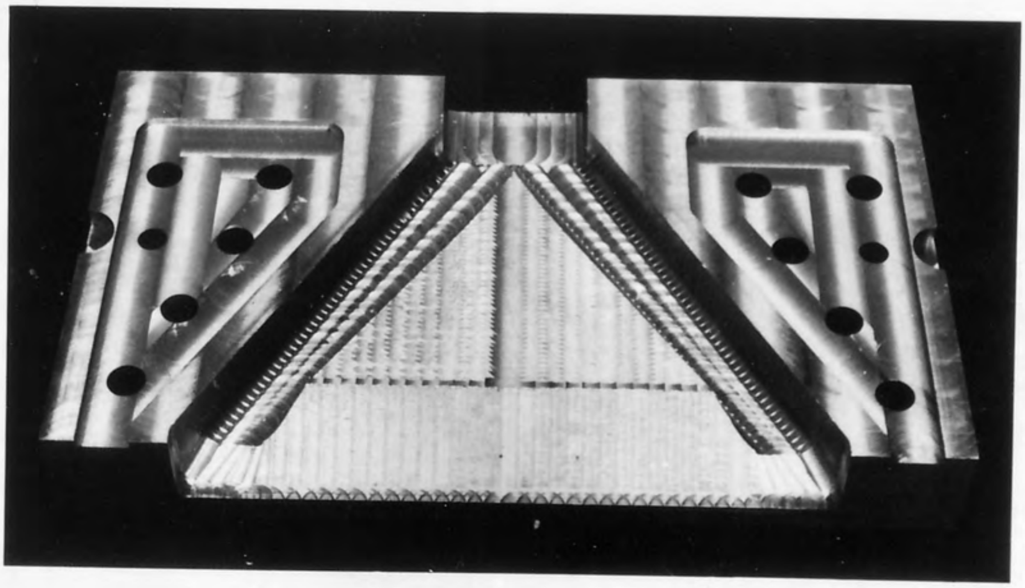


Fig. (6.8) Block for Extrusion Flow Control mentioned, would be very high. For instance, the part program to cut the extrusion block shown in Fig. (6.8) had a length of over 30,000 lines of information. The cost of producing punched paper tape of such programs is high; bearing in mind that generation of error free past programs at first run even for simple shapes is not usual. This points to the fact that on line transfer of data is essential and the existence of a high capacity storage device in the system is also indispensable.

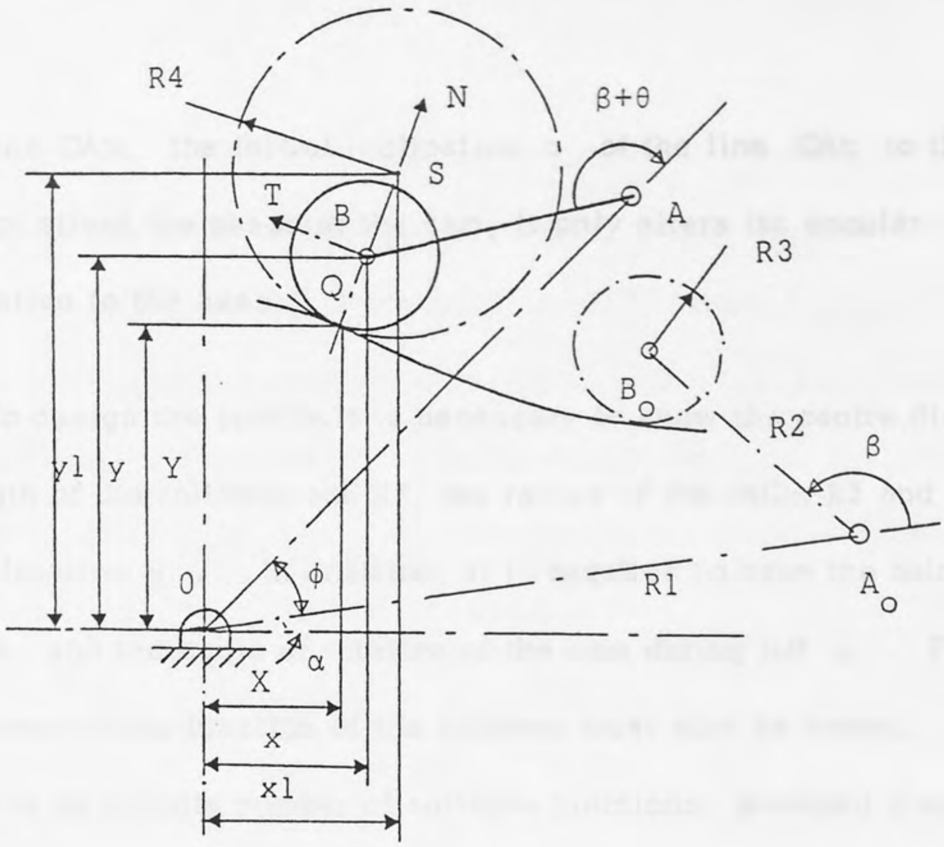
The primitive method of generating part program using a purpose written routine in FORTRAN was applied in cutting the die blocks. The function of the FORTRAN program was to calculate the data points, carry out the interpolations by calling subroutines applicable, compute cutter offsets and finally, store the part program produced on the disc.

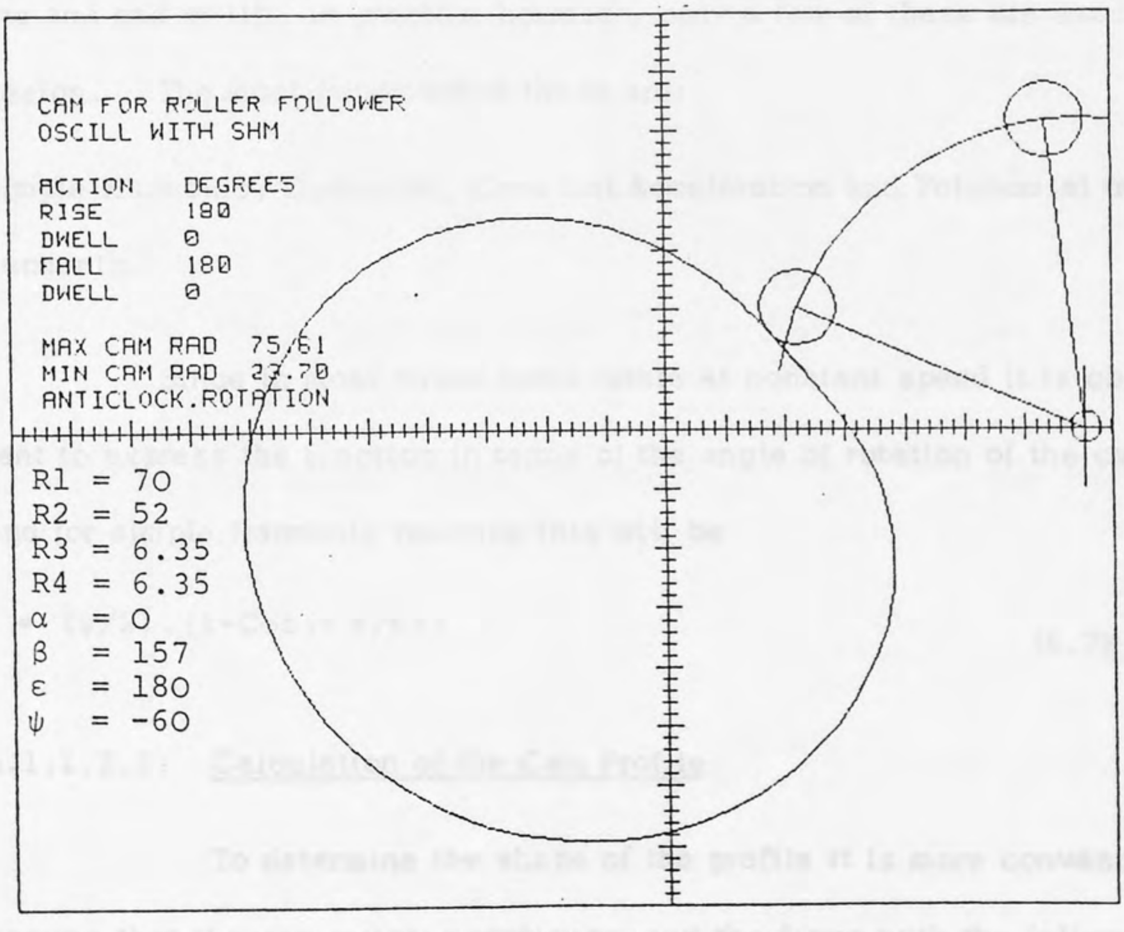
#### 6.1.2.2. Computer Aided Design and Machining of Cam Profiles

Cams operating at high speed must be designed very carefully to avoid excessive inertia forces and the profiles have to be machined to a very high standard of accuracy. Computer aided design and machining of cam profiles not only offers substantial time saving in calculations of the coordinates of the profile and the determination of milling cutter centre-line it also enables high accuracy to be achieved.

##### 6.1.2.2.1. General Principles

Fig. (6.9) shows the follower arm pivoted at A and carrying a roller at B. In the initial position the arm is inclined at an angle $\beta$ to





Computer Drawing of the Designed Cam

Fig. (6.9)

the centre line OAo; the initial inclination $\alpha$ of the line OAo to the X-axis does not affect the shape of the cam, it only alters its angular position relative to the axes.

To design the profile, it is necessary to know the centre distance R1, the length of the follower arm R2, the radius of the roller R3 and its angle of inclination $\beta$. In addition, it is required to have the total lift of the arm $\psi$ and the angle of rotation of the cam during lift $\varepsilon$. Finally the displacement-time function of the follower must also be known. In theory there is an infinite number of suitable functions, provided they satisfy the condition that the velocity of the follower is zero at the beginning and end of lift, in practice however, only a few of these are used in cam design. The most important of these are:

Simple Harmonic, Cycloidal, Constant Acceleration and Polynomial or Multi-harmonic.

Since in most cases cams rotate at constant speed it is convenient to express the function in terms of the angle of rotation of the cam and for simple Harmonic function this will be

$$\theta = (\psi/2).(1-\mathrm{Cos}(\pi\,\phi/\varepsilon)) \qquad (6.7)$$

### 6.1.2.2.2. Calculation of the Cam Profile

To determine the shape of the profile it is more convenient to assume that the cam remains stationary and the frame with the follower rotate in the opposite direction. First the X and Y coordinates of the roller centre are calculated for successive positions. These form the locus of the centres which is parallel to the cam profile and displaced from it by the roller radius. To find the position of the point of contact Q between the roller and cam profile it is necessary to offset a distance R3 along the normal to the profile N which is at right angles to the slope of the locus at B, see Fig. (6.9). This is repeated for a number of positions and the complete profile of the cam is obtained.

The coordinates of B are:

$$x = R1\text{Cos}\,(\alpha + \phi) + R2\,\text{Cos}\,(\alpha + \phi + \beta + (\psi/2)*(1-\text{Cos}(\pi\phi/\varepsilon))) \quad (6.8)$$

$$y = R1\,\text{Sin}\,(\alpha + \phi) + R2\,\text{Sin}\,(\alpha+\phi+\beta+(\psi/2)*(1-\text{Cos}(\pi\phi/\varepsilon))) \quad (6.9)$$

The slope at B is

$$\frac{dy}{dx} = \frac{dy/d\phi}{dx/d\phi} = \frac{U}{V}$$

where $U = R1\,\text{Cos}\,(\alpha + \phi) + R2\text{Cos}\,(\alpha + \phi + \beta + (\psi/2) * (1-\text{Cos}\,(\pi\phi/\varepsilon)))* (1 + (\psi\pi/2\varepsilon) * \text{Sin}\,(\pi\phi/\varepsilon))$

and $V = -R1\,\text{Sin}\,(\alpha + \phi) - R2\,\text{Sin}(\alpha + \beta + \phi + (\psi/2) * (1-\text{Cos}\,(\pi\phi/\varepsilon)))* (1 + (\psi\pi/2\varepsilon) * \text{Sin}\,(\pi\phi/\varepsilon))$

(6.10)

The angle of the slope : $\gamma = \tan^{-1}\dfrac{dy}{dx} + 180^{\circ}$ if $dx < 0$

The coordinates of Q are:

$$X = x + R3\cdot\text{Cos}\,(\gamma - 90) \quad (6.11)$$

$$Y = y + R3 \cdot \sin(\gamma + 90) \qquad (6.12)$$

The coordinates of the centre of the cutter S are given by:

$$x_1 = X + R4 \cdot \cos(\gamma - 90) \qquad (6.13)$$

$$y_1 = Y + R4 \cdot \sin(\gamma - 90) \qquad (6.14)$$

#### 6.1.2.2.3. Programming

Formulae (6.7) to (6.14) were implemented in a computer program by which the cutter centre-line coordinates at certain intervals to give the necessary accuracy are calculated.

The parameters required as input being :

Centre distance $R1$
Length of the follower arm $R2$
Radius of the roller $R3$
Radius of the cutter $R4$
X-axis - centre-line (OAo) angle $\alpha$
Initial arm inclination $\beta$
Total angle lift of the arm $\psi$
Angle of rotation of cam during lift $\varepsilon$
Angular steps in calculation of coordinate $\delta$

in (6.7) $\phi = \delta * n$ where n represents the nth coordinate calculated on the profile.

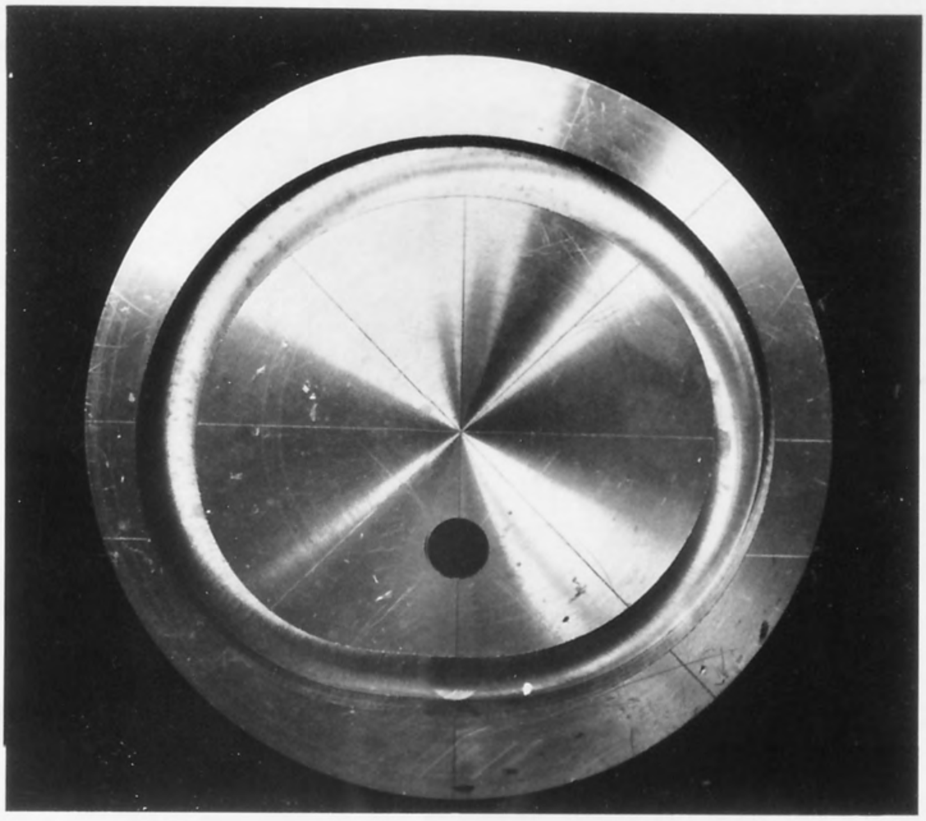


Fig. (6.10) An Industrial Cam with Simple Harmonic Motion

In addition to the above parameters the usual machine requirements such as feed rate, depth of cut, etc., are also needed as input.

Fig. (6.10) shows a typical industrial cam profile for a roller follower oscillating with Simple Harmonic motion produced in the DNC system.

## 6.2. The Use of the DNC System for Four Axis Machining

The component selected for machining to evaluate the effectiveness of the completed fourth axis system was a conical finned impellor having equally spaced helical fins.

### 6.2.1. Analysis of Impellor

The drawing of the component contained a table of coordinate data at every 15 degrees for defining a typical fin. This information should be used to relate the axial translation of the cutter to the rotational movement of the workpiece in producing the fins. As the coordinate data related to widely separated points, it was necessary to interpolate between the data values. The original information was rearranged and axis transformation was carried out so that it could be suitable for part program generation, Fig. (6.12).

It was realised that a single mathematical equation for the interpolation of the data values would not produce a satisfactory result within acceptable tolerance. Graphical analysis, using the pitch differences between adjacent data points (tangents), indicated that three individual curves could be distinguished.

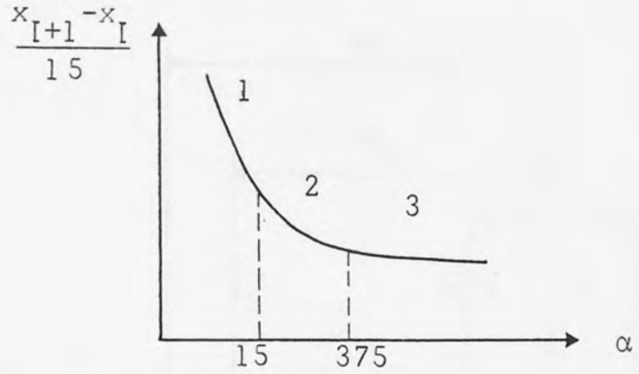


Fig. (6.11) Schematic Display of Pitch Differences for Successive Data Points

It was found that the first and the third curves were second degree polynomials and the second a power curve.

$$X_1 = C_{11} + C_{21}\,\alpha + C_{31}\,\alpha^2 \qquad 0 \leqslant \alpha \leqslant 150 \qquad (6.15)$$

$$X_2 = C_{21}\alpha^K \qquad 150 \leqslant \alpha \leqslant 375 \qquad (6.16)$$

$$X_3 = C_{31} + C_{32}\,\alpha + C_{33}\,\alpha^2 \qquad 375 \leqslant \alpha \leqslant 690 \qquad (6.17)$$

Applying least squares curve fitting method, the coefficients and the power K were found. The resulted equations gave excellent tangency at the curve intersection points and produced coordinates well within 50% of the general tolerance on the drawing.

Noting that

$$\frac{dx}{dt} = \frac{dx}{d\alpha} \cdot \frac{d\alpha}{dt}$$

where $\frac{dx}{dt}$ = Axial feed rate = F

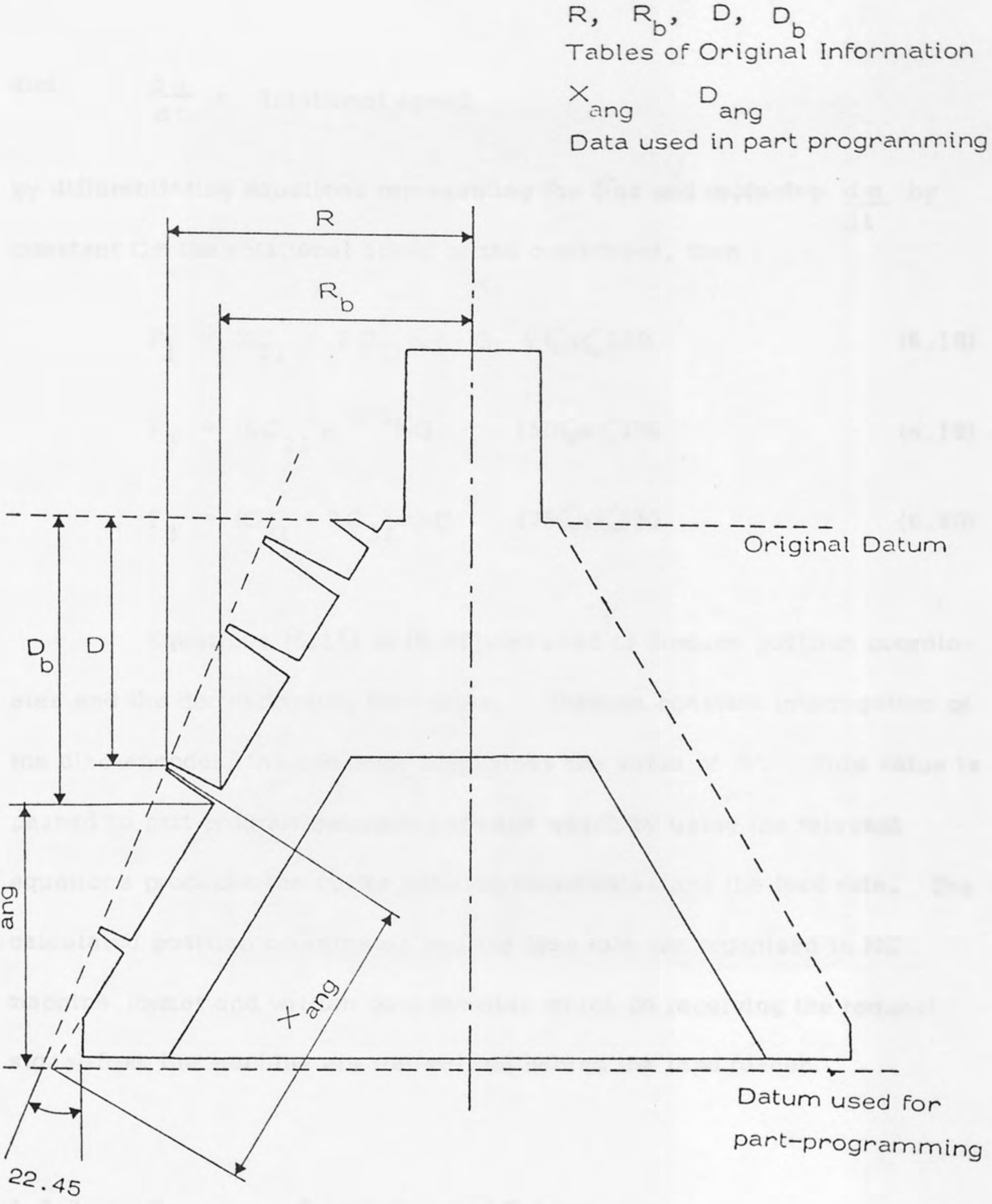

R, R_b, D, D_b
Tables of Original Information
X_ang D_ang
Data used in part programming
R
R_b
Original Datum
D_b
D
D_ang
X_ang
Datum used for part-programming
22.45

Fig. (6.12)

and $\frac{d\alpha}{dt}$ = Rotational speed,

by differentiating equations representing the fins and replacing $\frac{d\alpha}{dt}$ by constant C = the rotational speed of the component, then :

$$F_1 = (C_{21} + 2C_{31}\alpha) \cdot C \qquad 0 \leqslant \alpha \leqslant 150 \qquad (6.18)$$

$$F_2 = (KC_{21}\alpha^{K-1}) \cdot C \qquad 150 \leqslant \alpha \leqslant 375 \qquad (6.19)$$

$$F_3 = (C_{32} + 2C_{33}\alpha) \cdot C \qquad 375 \leqslant \alpha \leqslant 690 \qquad (6.20)$$

Equations (6.15) to (6.20) were used to produce position coordinates and the corresponding feed rates. Through constant interrogation of the disc encoder, the computer determines the value of $\alpha$. This value is passed to part program generating routine which by using the relevant equations produces the cutter position coordinates and the feed rate. The calculated position coordinates and the feed rate are organised in NC machine format and written onto the disc which on receiving the request signal from the machine are transmitted behind the tape reader.

### 6.2.2. Component Orientation and Cutting

The values of $X_{ang.}$ (see Fig. (6.12)) along the base cone surface were used in analysis of the fins. Therefore, the cone for cutting was mounted with its axis in X-Z plane of the NC machine tool and inclined such that the base cone surface was parallel to the machine table.

Fig. (6.13) Impellor Cut Using the Fourth Axis

Now the component contained three equally spaced fins. With the existing arrangement, after finishing each fin the component was manually rotated 120° and the same procedure was repeated to complete the cutting. Fig. (6.13) shows the component cut in aluminium.

Due to the limitations of the existing equipment, a great deal of difficulties were encountered in cutting operation. Manual positioning with an acceptable precision was not possible and the resolution of the cyclic encoder used, being only one degree, was too low to give an accurate reading of the position. In addition the on line generation of part program added to the inaccuracies, as the axis remains rotating while the computation is in progress. Finally the stepping motor in this work was too small; consequently, to produce sufficient torque it was necessary to use a chain of gears, the back-lash due to the gear train was also responsible for the loss of some accuracy. Nevertheless, the exercise as a whole achieved its purpose in building a basis for future development of a viable fourth axis and its supportive software programs, however, for this to be of commercial value, obviously more stringent control is required.

## 6.3. Design and Machining using BCSURF Package

### 6.3.1. Curve Design with BSCURV

Using BSCURV curves are designed simply by giving, as input data, the coordinates of the control polygon vertices, the order of the curve and some control variables, see AppendixII . The output of the routine is in form of a series of point coordinates on the approximated curve at equal parametric intervals defined by the user. Gino-graph plotter routines have been used in this program to generate graphical output.

The result from the curve design routine proved that the method employed undoubtedly fulfils many important requirements of computer aided curve design in the engineering field. The parametric representation makes the design of closed curves as equally simple as the open ones. Furthermore, with little extra computation space curves are produced if the vertices of the defining polygon are given as points in three-dimensional space. Parametric representation with all its advantages, however, suffers from one drawback; this being the fact that equal parametric distances do not relate to equal scalar values at different sections of the curve. Therefore, for example, to produce points at equal arc intervals additional computing and data manipulation is needed.

As can be seen in the following figures, unlike many global schemes, curves designed using this method are free from extraneous undulation. Moreover, the shape and complexity of the curves is controlled using different "handles", each producing an additional effect and hence, more flexibility.

#### 6.3.1.1. Flexibility of BSCURV in Shape Control

Different types of control can be used to change the shape of a B-spline curve using BSCURV routine.

I) Changing the order of the curve

II) Changing the position and number of defining polygon vertices, and using repeating vertices.

In Fig. (6.14) the polygon defined by vertices ($P_0$, $P_1$, $P_2$, $P_3$) is used to construct curves of different orders. As shown second order curve results in straight lines that join the vertices, and as the order of the curve increases the resulting curve becomes smoother and looks tighter. Note that in any case the end slopes are not affected.

The fourth order curve corresponds to Bezier curve, because the number of the vertices is equal to the order of the curve.

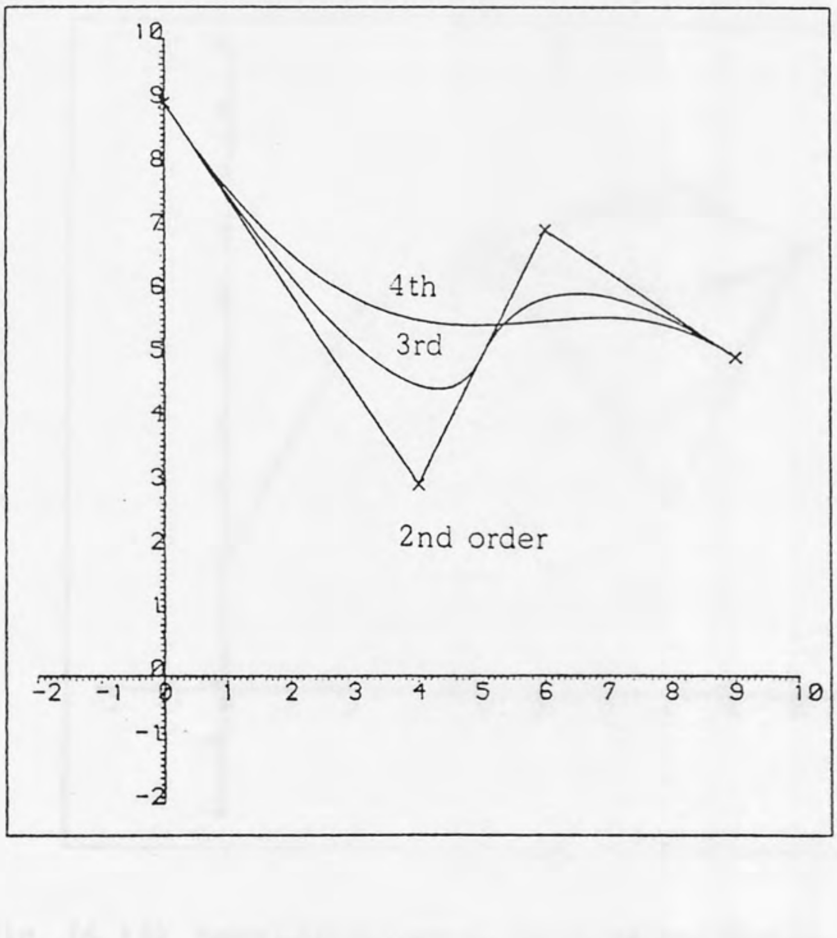


Fig. (6.14) B-spline Curves of Different Order

Fig. (6.15) illustrates how local modifications can be made by changing the position of one vertex. The non-global nature of B-spline is clearly shown here as this change only affects a limited part of the curve.

Unlike Bezier curve in that the number of vertices is fixed by the order of the curve, the B-spline only requires that

ORDER of the curve $\leqslant$ no. of vertices.

Therefore, in constructing a curve additional vertices can be inserted to modify the shape of the curve. In Fig. (6.16) by inserting an extra vertex the minor alteration caused can be seen.

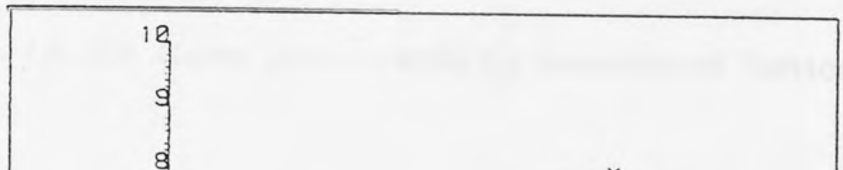


Fig. (6.15) Local Modification on B-spline Curves

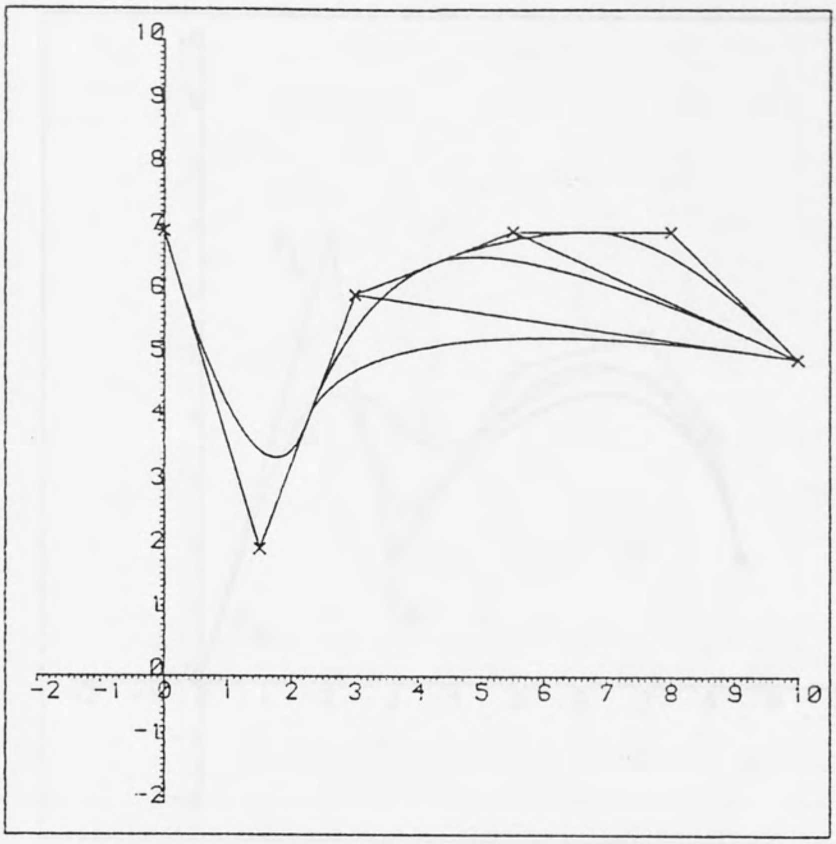


Fig. (6.16) Curve Modification by Insertion of Vertices

Finally, multiple vertices can play a useful role in controlling the shape of the curve. Referring to Fig. (6.17), for a set of fifth order curves, double, triple, and quadruple vertices are given at ( $P_2$ ). The curve is pulled towards the vertex as the multiplicity goes higher. The quadruple vertex for the fifth order curve creates a knuckle at this vertex since the slope becomes discontinuous.

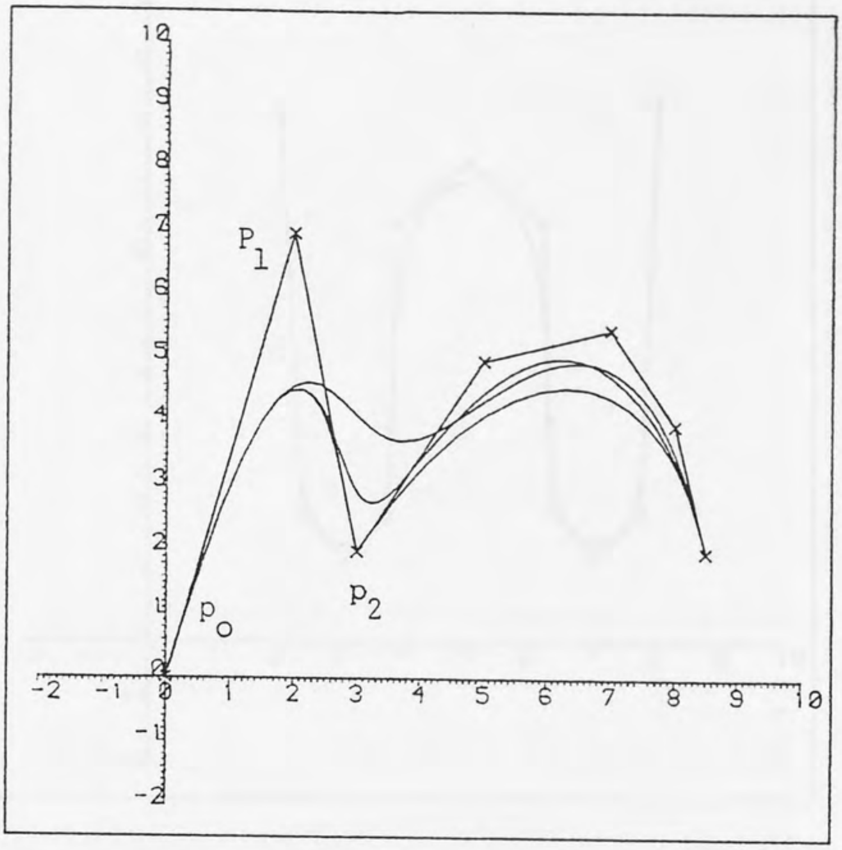


Fig. (6.17) The Effect of Multiple Vertices

Furthermore, by defining the same number of collinear vertices as the order of the curve, it is possible to embed linear segments in a curve, an example being shown in Fig. (6.18).

The storage required to implement this routine on a computer is very modest and in fact, the bulk of it is occupied by the graphical routines. The computational time depends on the number of the vertices of the defining polygon and the order of the curve. The following figures show examples of curves designed with BSCURV.

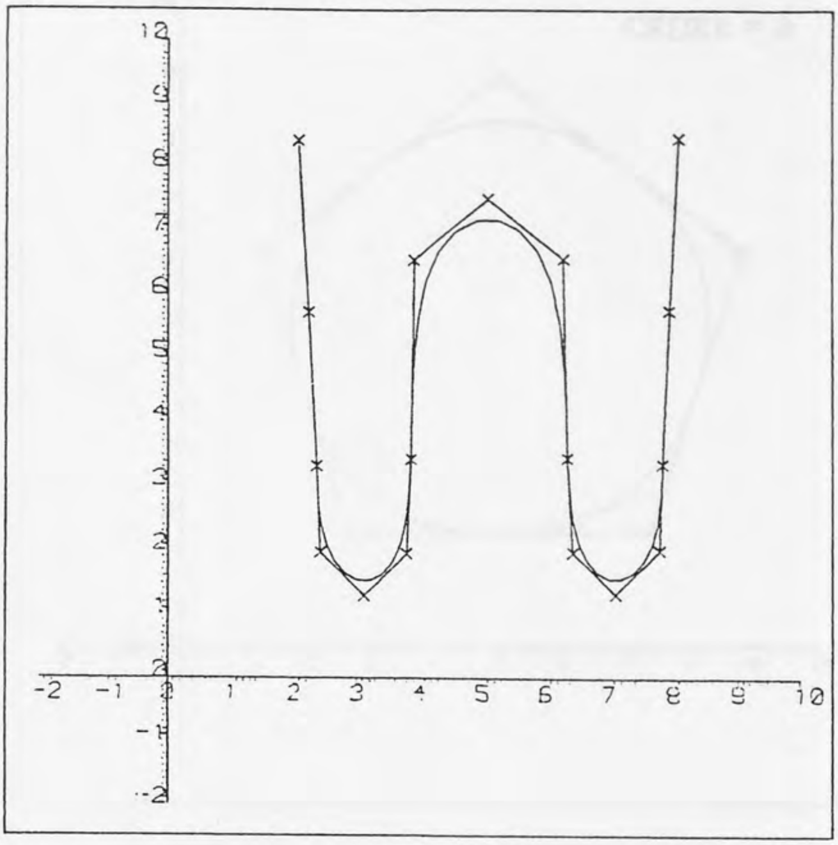


Fig. (6.18) Embedding of Linear Segments

### 6.3.2. Curve Fitting

Curve fitting routines developed in the package accomplish an essential aim of the system which was to include newly designed curves, existing curves, as well as constraints presented, as elements in surface design. The two fitting procedures investigated, namely, the least squares approximation and Reisenfeld's inversion technique, each demonstrated its importance in certain areas of applications.

Using BSLSQ based on the least-squares method, in search for finding the best fit to a set of data points, in addition to using different order B-spline curves, the knot set can also be changed so that the desired curve is produced. A spline curve which adequately fits the data and is

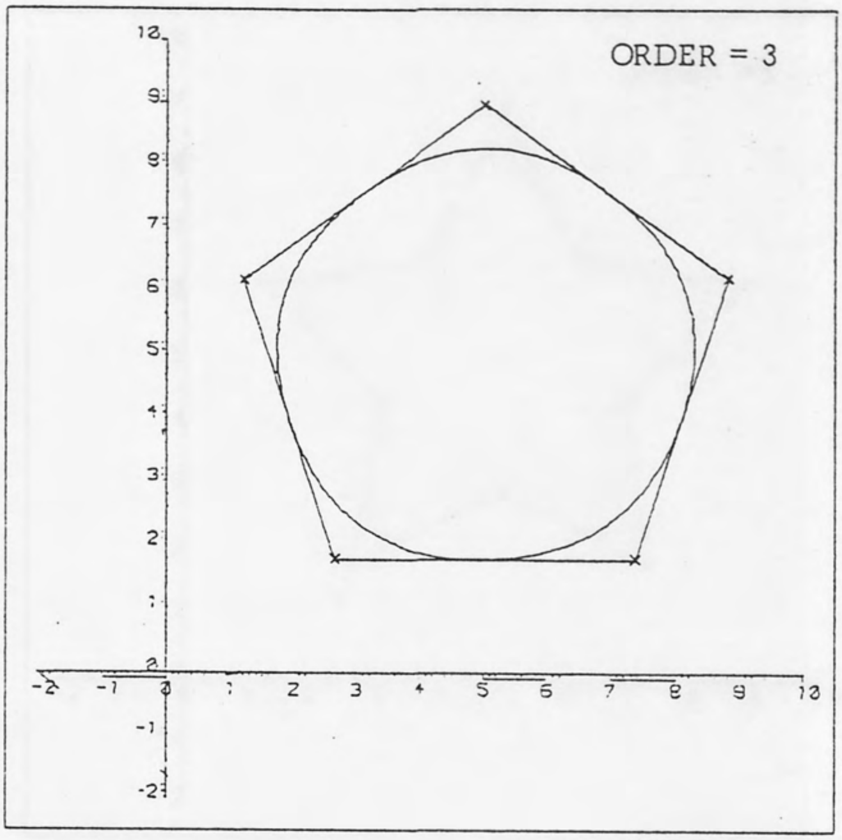


Fig. (6.19) A 3rd Order Closed B-spline Curve

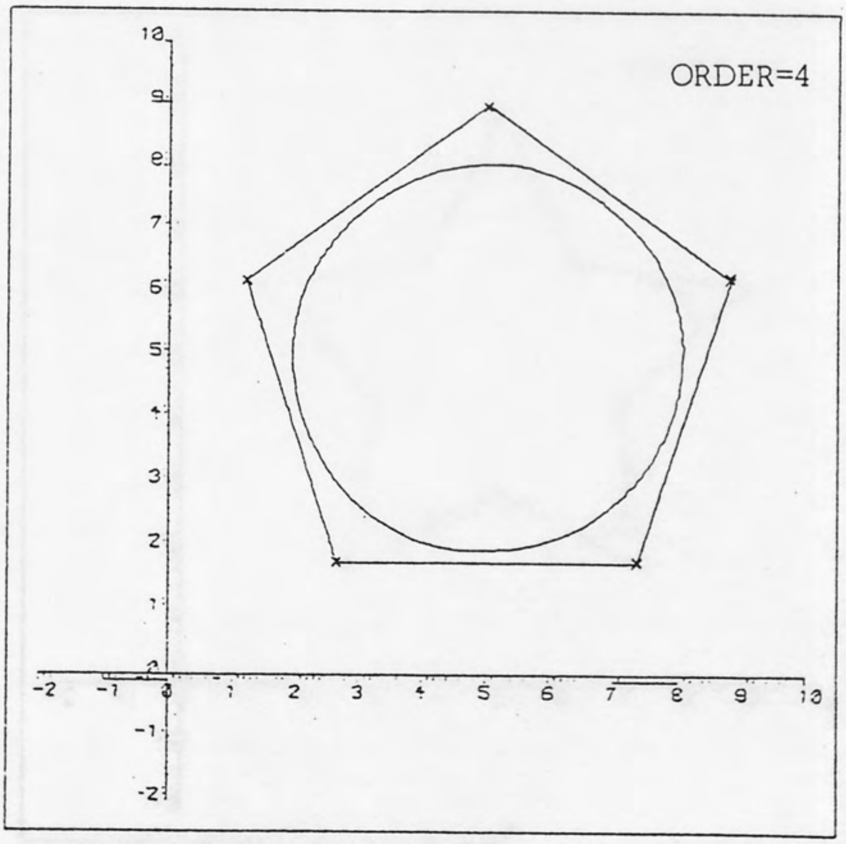


Fig. (6.20)

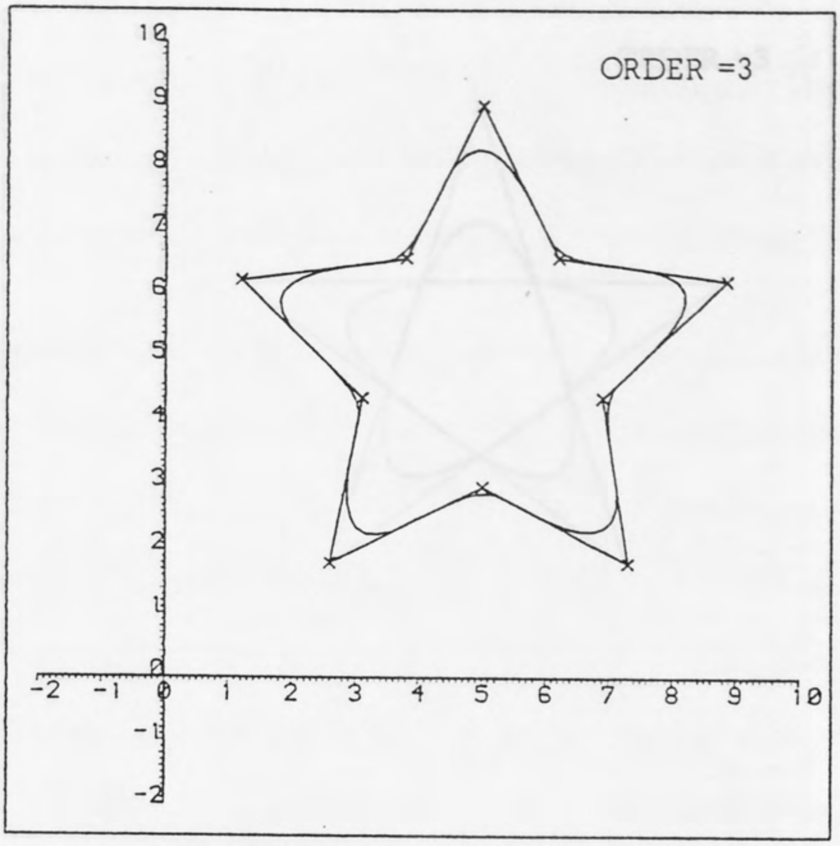


Fig. (6.21)

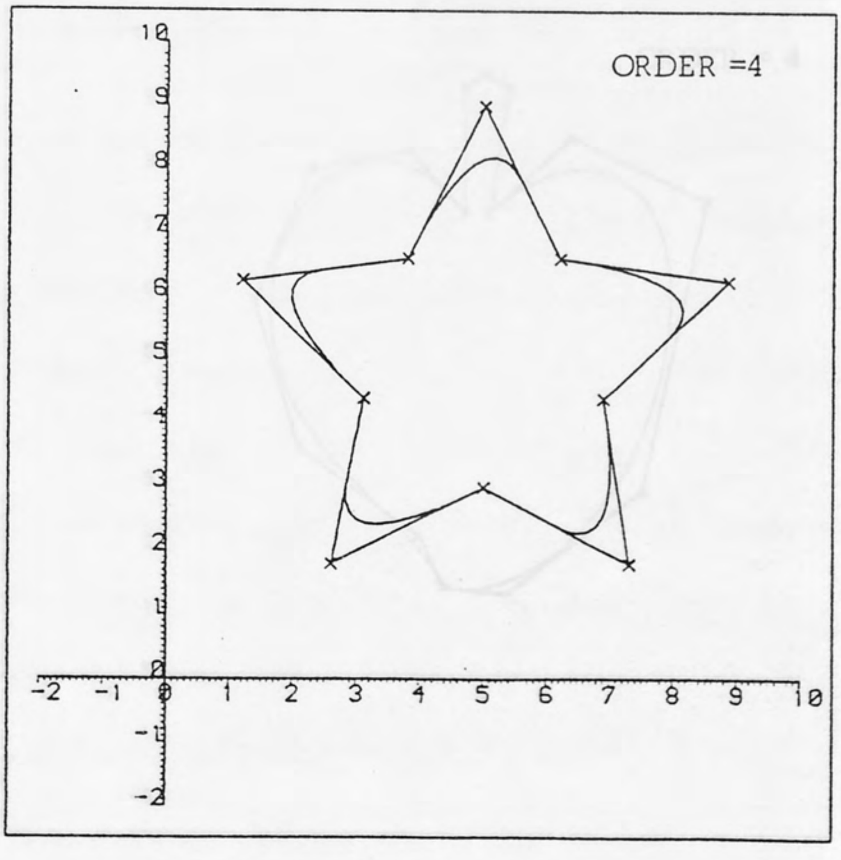


Fig. (6.22)

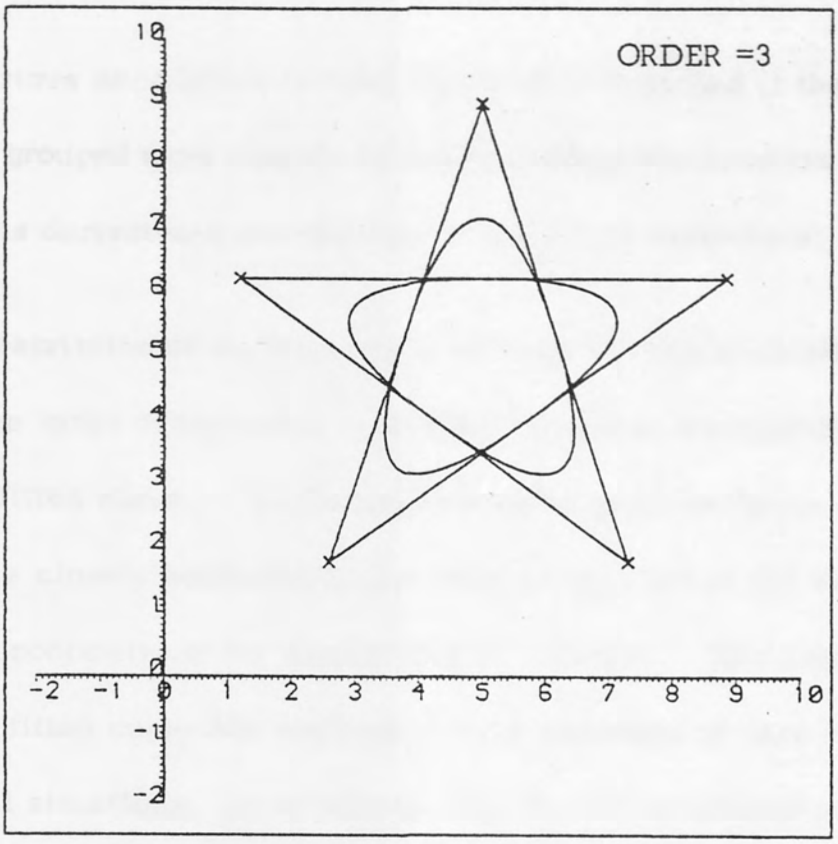


Fig. (6.23)

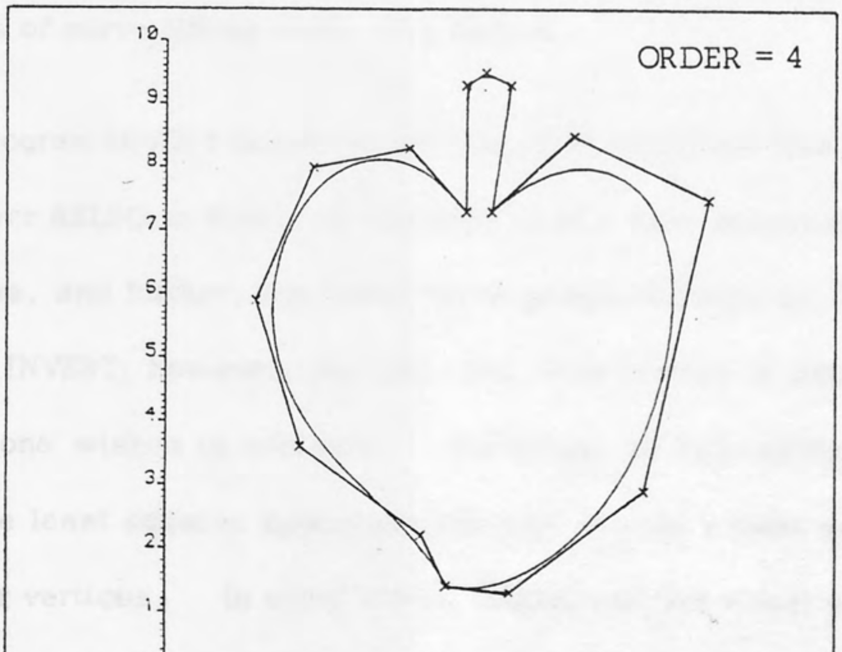


Fig. (6.24)

free from spurious oscillation is more likely to be obtained if the knots are chosen to be grouped more closely in regions where the function (underlying the data) or its derivatives change more rapidly than elsewhere.

Prescription of multiple knots as long as their multiplicity do not exceed the order of the curve is another handle in manipulating the shape of the fitted curve. In the regions where multiple knots are defined the curve more closely approximate the data points, but at the same time the degree of continuity of its derivatives is reduced. The computed values of the fitted curve has negligible error compared to data points in most practical situations, nevertheless, the flexibility of manipulation of the shape of the curve makes this method very satisfactory, particularly if used in CAD systems based on B-splines. Fig. (6.25) and Fig. (6.26) are examples of curve fitting using this method.

Program INVERT based on the inversion technique has the advantage over BSLSQ in that it is computationally more efficient and simpler to use, and further, the fitted curve passes through all the data points. In INVERT, however, one uses the same number of data points as the vertices one wishes to evaluate. Therefore, by increasing the number of points, the least squares approximation may provide a more satisfactory fit with fewer vertices. In other words, BSLSQ renders itself more easily in fitting a curve to a large number of empirical data points, whereas INVERT is more efficient in approximating an existing curve.

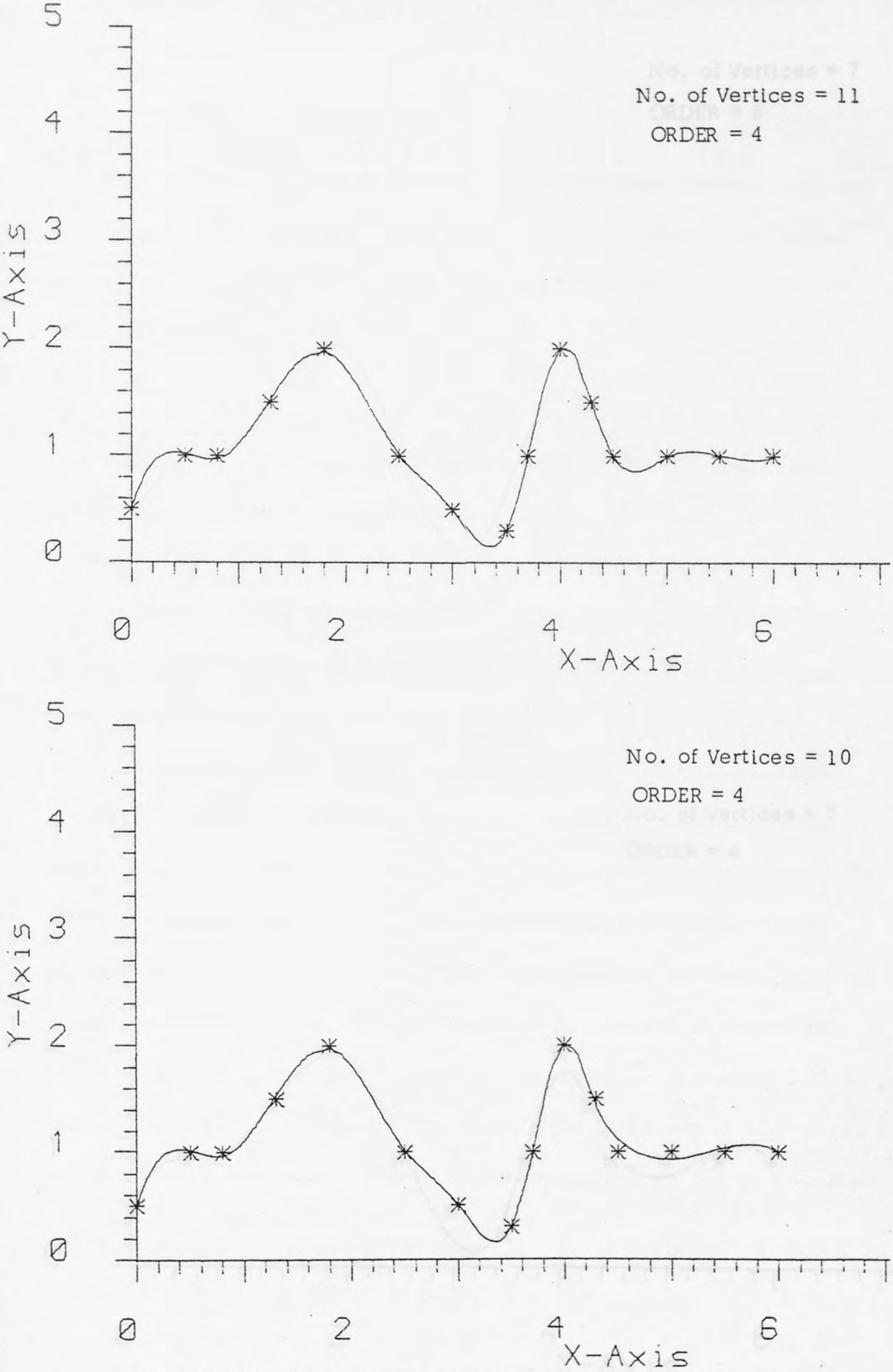


Fig. (6.25) The Effect of Selected Vertices on the Fitted Curve

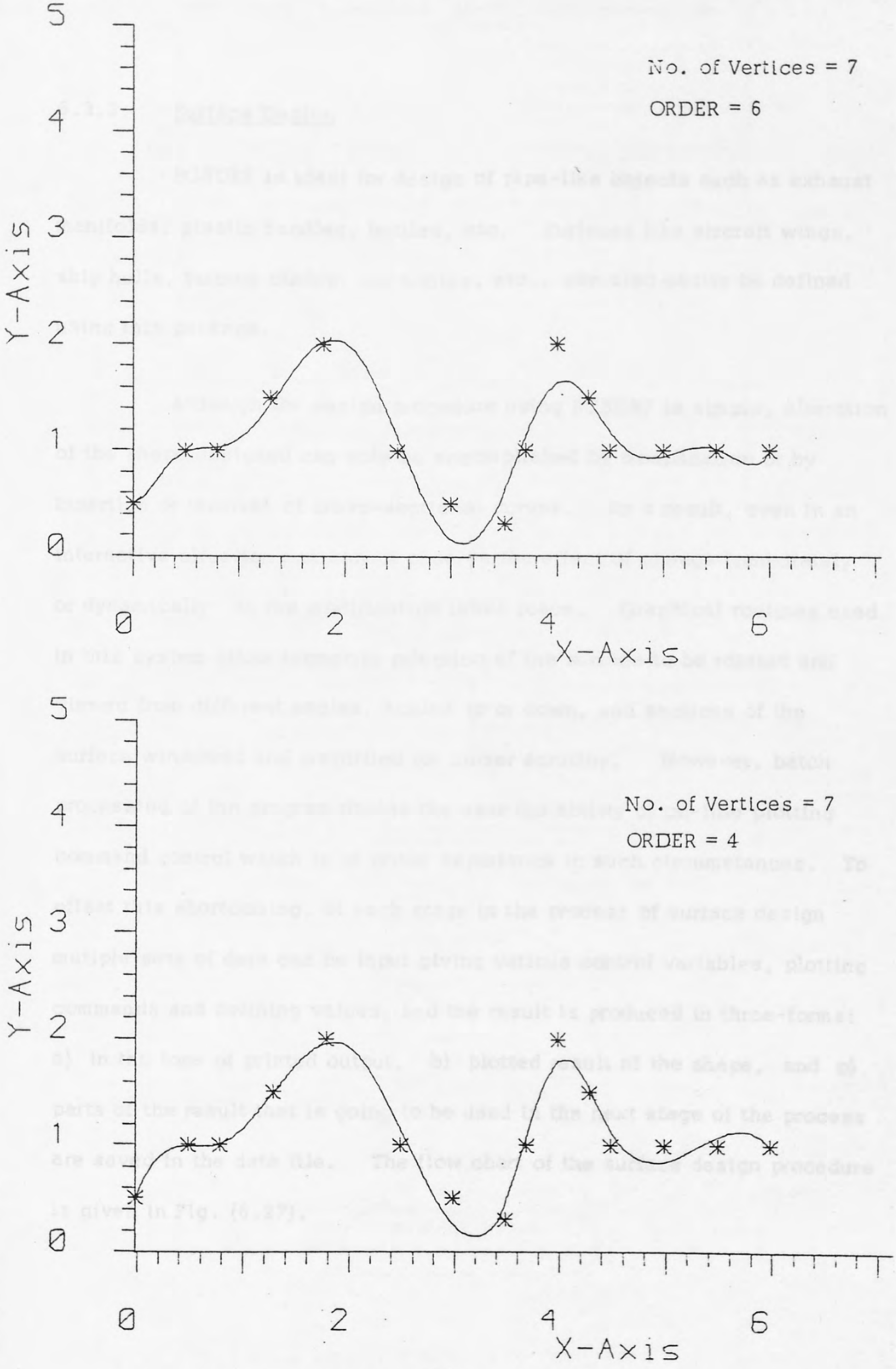


Fig. (6.26) B-spline Curve Fitting Using Different Orders

### 6.3.3. Surface Design

BCSURF is ideal for design of pipe-like objects such as exhaust manifolds, plastic handles, bottles, etc. Surfaces like aircraft wings, ship hulls, turbine blades, car bodies, etc., can also easily be defined using this package.

Although the design procedure using BCSURF is simple, alteration of the shape designed can only be accomplished by modification or by insertion or removal of cross-sectional curves. As a result, even in an interactive situation one cannot observe the effect of change immediately or dynamically as the modification takes place. Graphical routines used in this system allow isometric prjection of the surface to be rotated and viewed from different angles, scaled up or down, and sections of the surface windowed and magnified for closer scrutiny. However, batch processing of the program denies the user the ability of on-line plotting command control which is of prime importance in such circumstances. To offset this shortcoming, at each stage in the process of surface design mutiple sets of data can be input giving various control variables, plotting commands and defining values, and the result is produced in three-forms: a) in the form of printed output, b) plotted result of the shape, and c) parts of the result that is going to be used in the next stage of the process are saved in the data file. The flow chart of the surface design procedure is given in Fig. (6.27).

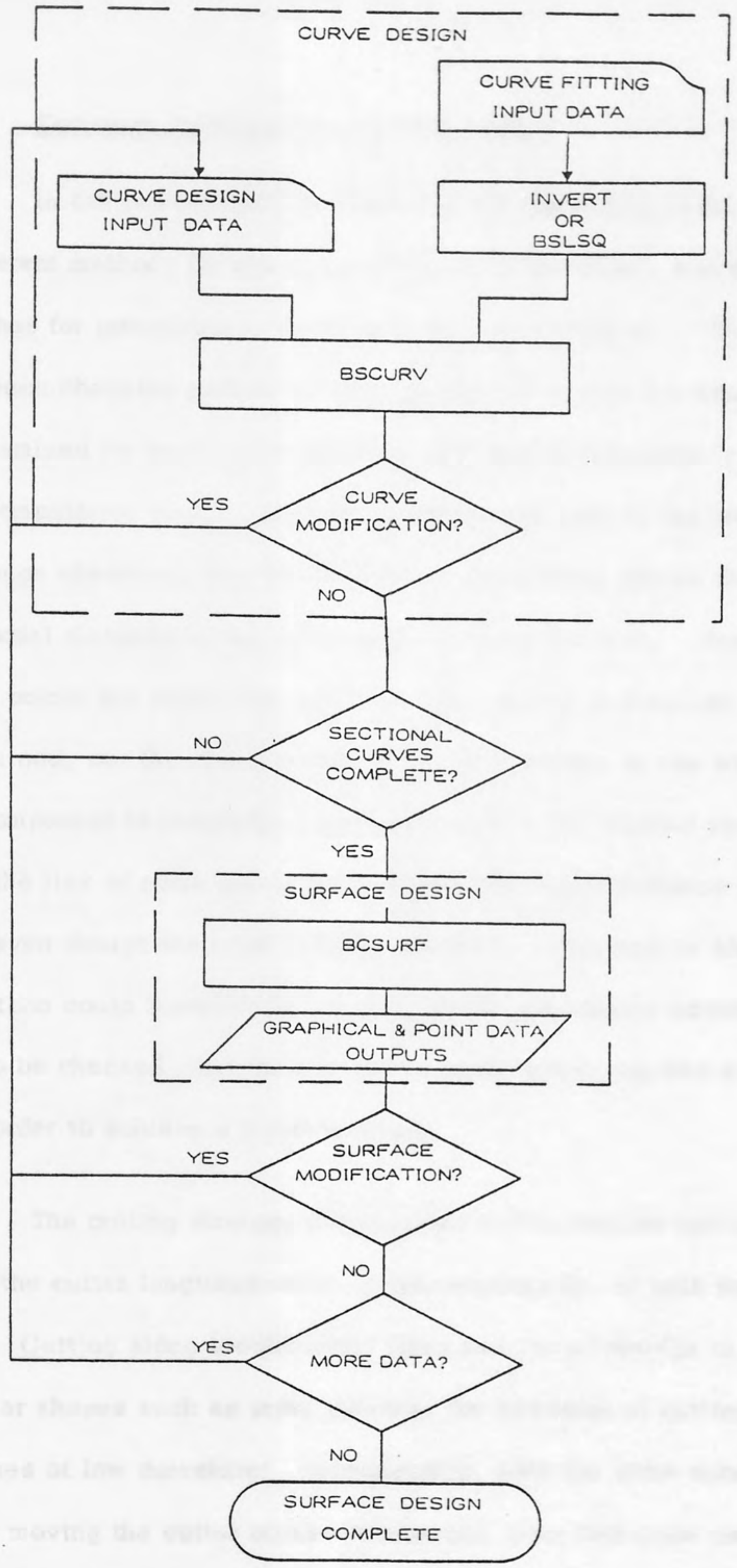


Fig. (6.27) Surface Design Procedure in the System

### 6.3.4. Computer Aided Machining with BCSURF

In the two routines developed to aid machining in this package, two different methods for the computation of cutter offset, and two different approaches for interference checking have been employed. The full-proof interference checking method adopted in ANAOFF allows the whole surface to be examined for each cutter position, but this will inevitably lead to higher computation cost. In order to reduce the cost in the process of interference checking, one would only consider those planes that are within the radial distance of the cutter and eliminate the rest. Assuming that the data points are organised in matrix form, points in a column or a row are examined, but the check moves to the next column or row when the edge of the component is reached. However, with a "U" shaped part, there is always the risk of some points being within the radial distance of the cutter, even though the edge is being reached. The routine ANAOFF in its general form could therefore be used for small components having fewer points to be checked, but their shape is such that it requires extensive care in order to achieve a reliable result.

The cutting strategy incorporated in this routine enables the user to move the cutter longtitudinally, cross-sectionally, or both for a better finish. Cutting along longtitudinal lines has the advantage in that, firstly, for tubular shapes such as pump volutes, the direction of cutting would be along lines of low curvature; consequently, with the same number of NC steps as moving the cutter cross-sectionally, finer tolerance can be accomplished. Secondly, the small ridges left between successive passes are now aligned with direction of the duct, so for such objects as pipe fittings where there is fluid flow the ridges cause little resistance to the flow.

Using the routine NUMOFF based on a numerical method of calculating cutter offset, as the tool is offset along the normal at the centroid of a triangle joining three neighbouring points specifying the surface, the ridges between successive passes are left over the data points. With ANAOFF, on the other hand, the cutter is directly offset over the data point position and hence, all of them will accurately lie on the machined surface. One would, therefore, assume that the analytical method used in ANAOFF produces a better tolerance. Nevertheless, surface finish must always remain a compromise between computing and NC machine cost, and putting the work into hand for polishing. Fig. (6.28) shows a ring designed using the package with normals drawn on the data points.

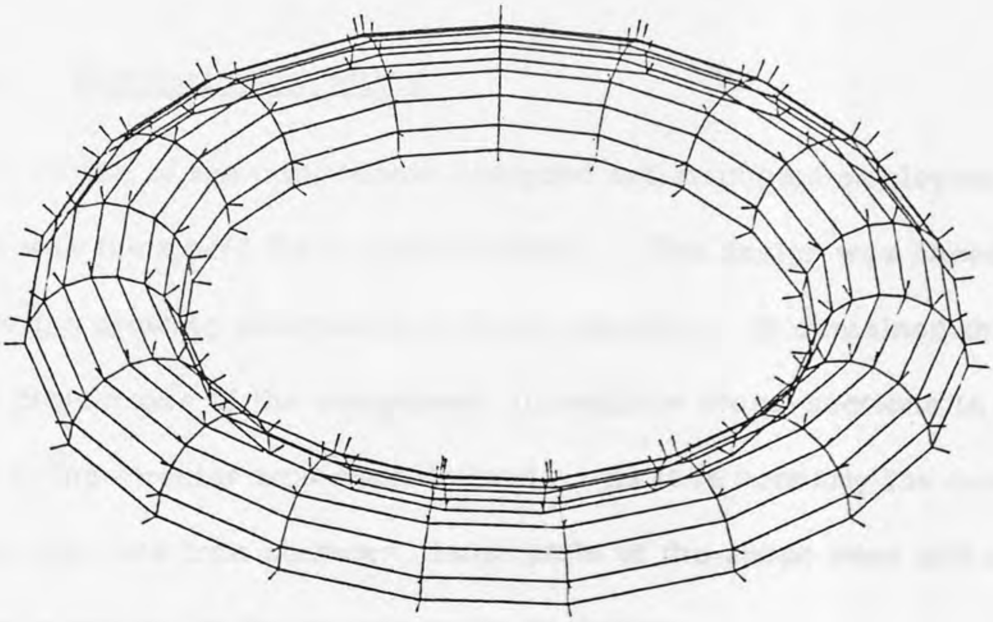


Fig. (6.28)

### 6.3.5. Flow of Data in the System

The Data General mini computer interfaced to the N C machine was limited in core memory capacity (16K of 16 bits), and furthermore, it was not equipped with any kind of graphics facilities. It was, therefore, decided to implement the surface design routines on the ICL 1904 computer at the computer centre of the University of Aston.

The data transfer between the two computers takes place in the form of punched paper tape. To achieve delimiter comptatibility the macro program "CONVERT" processes the output data and applies appropriate modifications such as insertion of "Carriage Return" character etc., which are required by the mini computer at the end of each line, and finally produces punched paper tape of the output. The paper tape is then loaded onto the magnetic disc of the mini computer for transfer to the N C machine. Block diagram of data flow is shown in Fig. (6.29).

### 6.3.6. Practical Applications

One of the components designed and machined employing the system was the mould for a plastic handle. The design was based on an engineering drawing provided by a local industry. It contained third degree projections of the component, in addition cross-sections in a few places using circular arcs were defined. As it is normally the case for objects with free form surfaces, large parts of the shape were left undefined and was a matter for the pattern maker to decide.

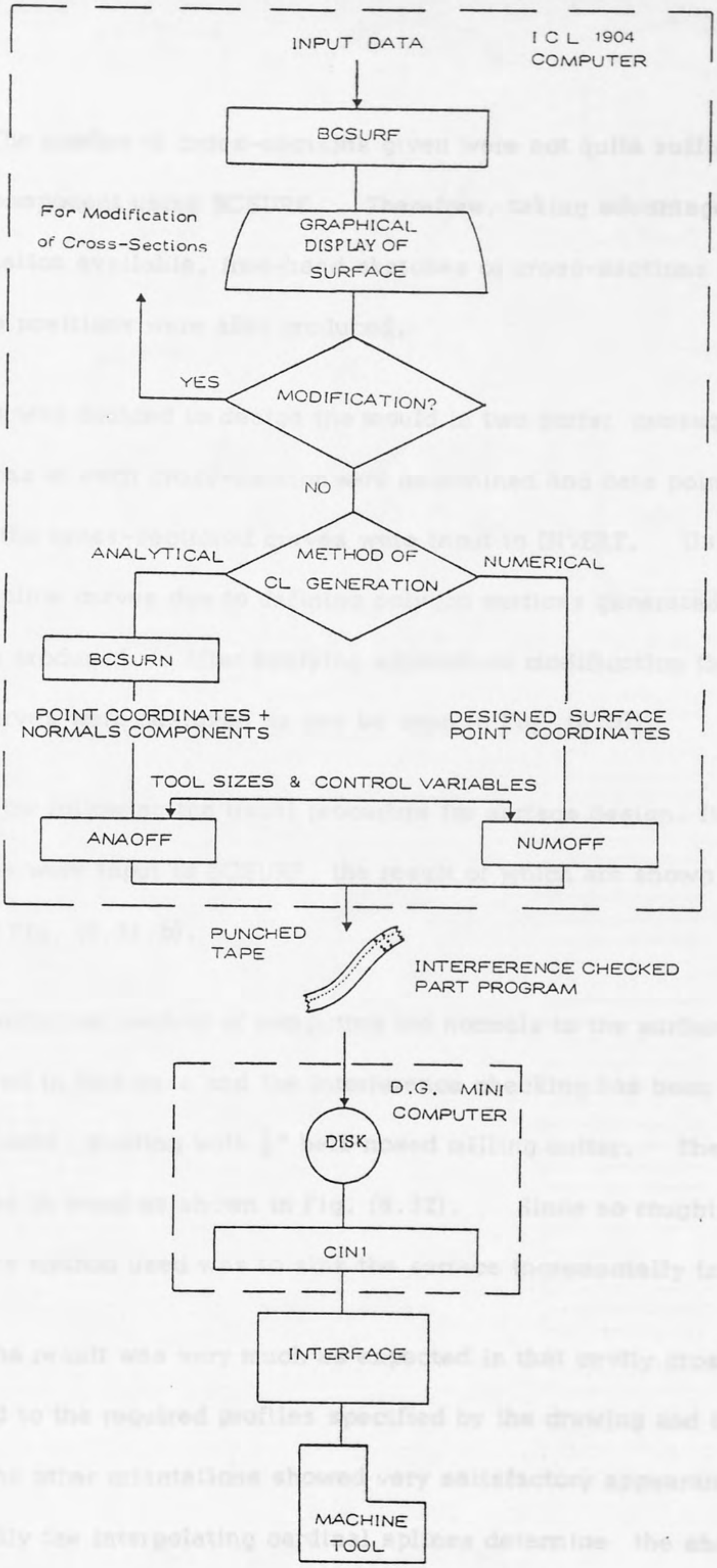


Fig. (6.29) Block Diagram of Data Flow Using BCSURF

The number of cross-sections given were not quite sufficient to define the component using BCSURF. Therefore, taking advantage of other information available, free-hand sketches of cross-sections at a few intermediate positions were also produced.

It was decided to design the mould in two parts; consequently, split positions at each cross-section were determined and data points extracted from the cross-sectional curves were input to INVERT. Using BSCURV B-spline curves due to defining polygon vertices generated by INVERT were produced. After applying appropriate modification the cross-sectional curves were designed as can be seen in Fig. (6.30).

Now following the usual procedure for surface design, final polygons vertices were input to BCSURF, the result of which are shown in Fig. (6.31.a) and Fig. (6.31.b).

Analytical method of computing the normals to the surface has been employed in this case and the interference checking has been carried out for four tools, starting with $\frac{1}{2}$" ball-nosed milling cutter. The mould was machined in wood as shown in Fig. (6.32). Since no roughing tape was produced, the method used was to sink the surface incrementally in Z.

The result was very much as expected in that cavity cross-sections corresponded to the required profiles specified by the drawing and the profiles at the other orientations showed very satisfactory appearance. Longtitudinally the interpolating cardinal splines determine the shape of the profile. Although they pass through control points, i.e. the cross-

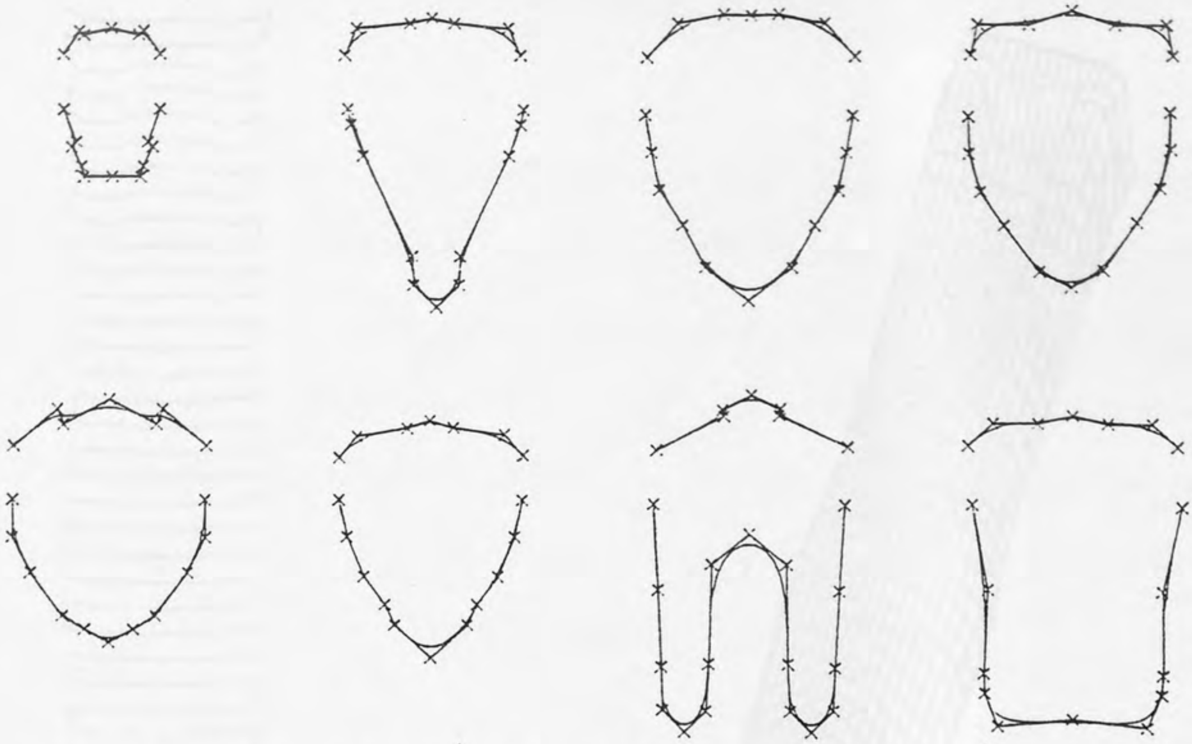


Fig. (6.30) Cross-sectional Curves for Upper and Lower Parts of the Mould

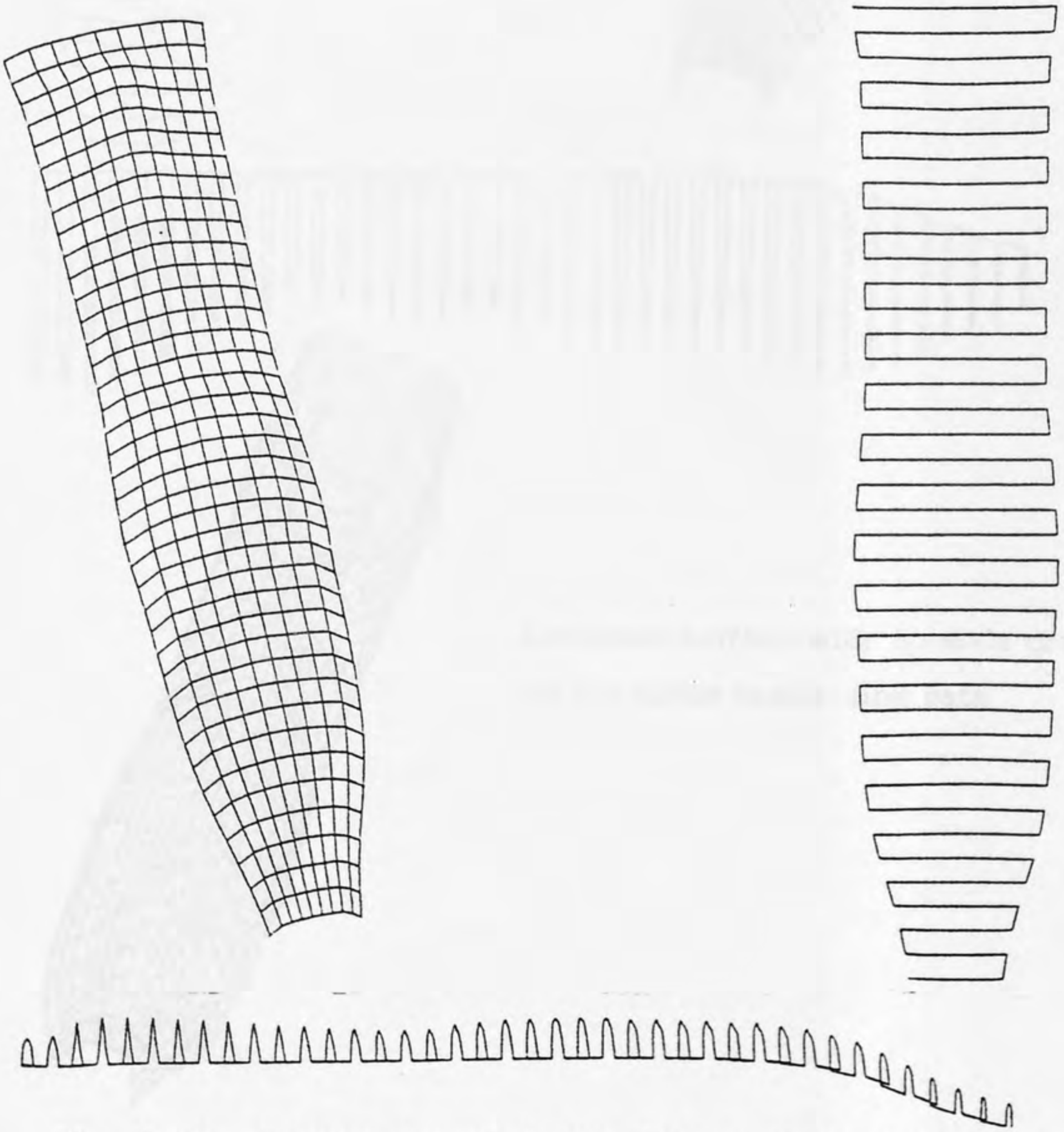
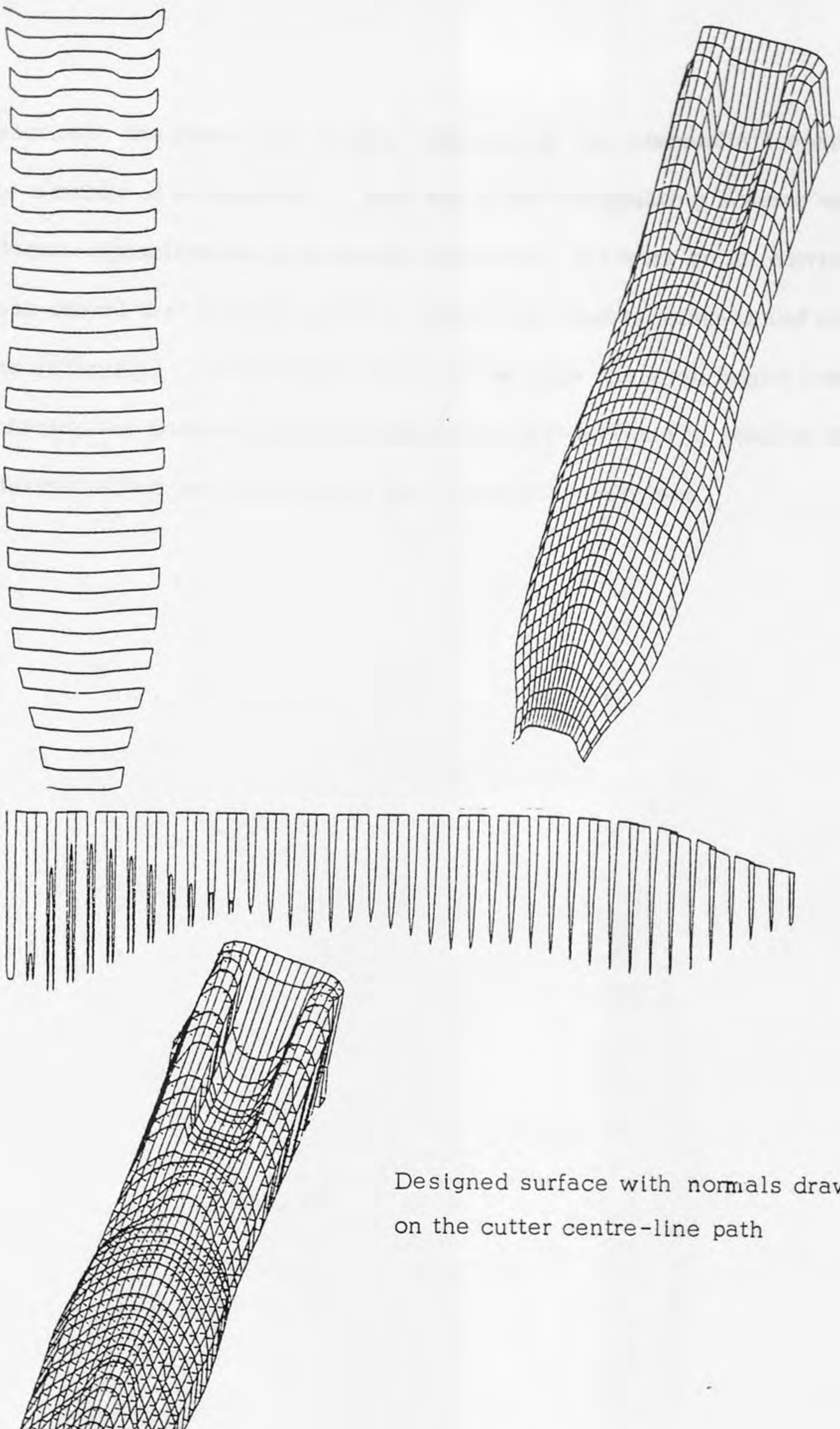


Fig. (6.31.a) Upper Part of the Mould Viewed from Different Angles



Designed surface with normals drawn on the cutter centre-line path

Fig. (3.31.b) Lower Part of the Mould

sections, the prediction of their behaviour at the intermediate positions is a matter of experience. Like any other interpolating scheme certain degree of oscillation is of course inevitable, but work so far carried out has proved that in most practical situations this is minimal and creates no difficulty. In addition, using the package one soon learns how by chaning the position and the number of cross-sections to produce the desired effect and to minimise the unwanted undulations.

Fig. (6.32) The Mould for Plastic Handle

Fig. (6.33) Exhaust Duct Machined in the System

# CHAPTER SEVEN

# CONCLUSIONS AND REMARKS

## 7.1. Working of the DNC System

### 7.1.1. Industrial Value of the System

The microprocessor based CNC systems are today the dominant form of NC design, and for the future method of production, there is a strong indication that the trend is towards modular integrated manufacturing systems. With the role of Direct Numerical Control in these systems, any contribution at any level and however small to general body of expertise in developing and running of DNCsystems is essential to achieving or maintaining industrial comp etitiveness.From a less futuristic point of view, and on a smaller scale, DNC cells similar to the one developed at the present work can bring considerable economical and technical benefits, particularly to those small and medium-sized industries which are using the older generation of NC machine tools and are now lagging their stronger competitors who have easier acces to new technology, improved production techniques and computer assistance. This is an inexpensive, but highly effective way of providing functional and managerial improvements and the advantages it brings by far outweigh the cost of its development.

### 7.1.2. Reliability of Data Transmission

The hardwired interface as designed, proved to be a very reliable data transmission link between the mini computer and the NC machine tool. The balanced differential lines connnecting the two parts of the interface were shown to be immune to the high electrical noise of the laboratory and well capable of carrying data over the relatively long distance between the two units in the system. Moreover, the sequencing of the data shift and the serial asynchronous form of transmission demonstrated that no loss of information occurs in the course of operation. At the initial stages of this work it was decided to use the parallel form of data transfer, i.e. to connect the data lines out of the computer bus directly to the machine control unit. But it was realised that in average there was one error out of every 250 characters transmitted; consequently, the present method was adopted and since the comissioning no problem has been encountered.

### 7.1.3. The Importance of Buffer Store

The buffer memory incorporated in the interface is of the type FIFO with the total capacity of 288 characters. Each interface board in the interface has its own buffer area, and as a result, the data transfer in three parts of the systems, i.e. from the computer to the computer interface board, from the machine interface board to the machine control unit, and between the two interface boards can take place independently, each operating at a different frequency.

The computer data out transfer rate is 250,000 characters per second, whereas the transmitter/receiver having a 200 KHZ clock can transmit approximately upto 1200 characters per second. The rate of data input to the Numerical Control unit on the other hand is much lower; that is at only 156 char./sec. Consequently, the successively increasing speed of data transfer at consequtive parts of the interface from the controller to the computer ensures that there is always data available for the machine. The Acramatic controller has its own internal memory that can store one block of part program, however, this arrangement provides an additional safeguard so as to make certain that there is no cutter dwell or a broken tool while cutting.

The interface as designed sends a data request to the computer only when 9/10 of the data stored in the computer interface board buffer memory is transmitted which could in average amount to 5 blocks of part program. Although the speed of data input to the Numerical Control unit depends on the time required for performing machining instruction presented in each block of the part program, nevertheless, with this system in most practical situations, the time intervals for requesting data compared to the computer time in filling the memories is considerably higher. Therefore, a very high priority could be given to servicing of the interface. In addition this means that in a time sharing computer used to control a DNC cell in a BTR mode, fewer interrupts take place and hence, savings is made on computer time which allows a larger number of machine tools or other peripherals to be controlled using a single computer.

Finally, it must be noted that the size of the buffer memory incorporated into the interface in this work proved to have satisfied the main requirements of the system at the present condition. However, the provision of a larger memory capacity and perhaps the use of a different type of memory unit could in other cases be needed or beneficial.

### 7.1.4. Functional Improvements and Flexibility

#### a. Elimination of Tape Handling

One of the main weaknesses of conventional numerically controlled machine tools has always been the problem of punched paper tape handling, and the tape reader has generally been regarded as the most troublesome link in these systems. With the direct transfer of data from the computer behind the tape reader in the DNC system, a much more reliable data input to the control unit is accomplished and further, a considerable saving in data preparation cost will be made by eliminating the paper tape. A punched tape, if needed as a permanent file, will be produced only when the program has been proved and the errors corrected. The importance of direct data transfer is better realised when producing part programs for components whose machining require software interpolation, as these usually result in very long programs.

#### b. Increased Capability

The successful production of components which required three-dimensional milling capability on the DNC system demonstrated that by software manipulation some control limitations can easily be overcome and the range of machining enhanced. Using software interpolation it is proved that a non-contouring NC machine can indeed be converted into a system capable of performing such functions in two or three dimensions. It should be noted that once the basis of software control has been established, owing to the inherited adaptability of this type of control and its versatility, software routines can be written to cover a wide range of machining operations and modified to suit new working environments as and when required. Moreover, development of such changes is simple, does not need special skill, and the cost of their implementation is low.

### 7.1.5. Ease of Part Programming

In applying computers in part programming, either a comprehensive APT like processor can be used or a simpler approach such as the method employed in this work can be adopted. Special purpose processors such as APT are usually very large and can only be run on big computers with high core memory capacity. In addition, post-processors are required to translate the output of these processors to NC machine format, and further, learning the programming with them is no easier than the commonly used computer languages. Therefore, when using special purpose processors, considering the initial capital cost of acquiring these packages, personnel training expenses and the high computation costs result in an expensive method of part programming that can economically be viable only for companies with large number of NC machines and a wide variety of component types.

With the rapid rate of reduction in price of computers, the mini and micro computers are becoming ever more widely available to small and medium-sized industries, and it is not perhaps incorrect to suggest that it is more advantageous to these industries to develop their own software with reference to a specific need.

The successful application of FORTRAN for part programming at the present work has proved that the use of general purpose computer languages on a small-sized mini computer can offer a simpler and less expensive approach to this problem. Using a purpose written routine in FORTRAN for part programming, cutter offset calculations, interpolation, inch/metric conversion, etc., can easily be carried out and at the same time NC machine formatted program produced. However, many industries mainly deal with components which are similar in shape, or contain parts which are similar or require the same type of computations. In such circumstances, 'family' type part program generating subroutines could be developed, a combination of which supplied with the input data specifying the basic dimensions of the component required would result in an efficient method of computer aided part programming.

The control programs developed and the method of part programming devised for the DNC cell in this research provide a high degree of flexibility and allow the computer to assume many different roles in the system. On the subject of part programming, the experience has shown that the edit facility is perhaps one of the most useful features. Here, the part program generated using FORTRAN routines, or input from an EIA or ASCII coded tape, or directly typed in through the keyboard can be stored onto the disc, where the powerful edit functions of the computer can then effectively be used to correct the errors without recourse to reprinting the whole program before transfer to the machine.

### 7.1.6. Viability of the Added Rotational Axis

The result from the present work proved that the incorporation of a rotational axis upon a Numerically Controlled machine tool within a DNC cell is feasible. Now, although the rotational axis developed showed many shortcomings, these could mainly be attributed to limitations of the equipment employed. Leaving these aside, the existing arrangement and design seem to present a satisfactory method in achieving a solution to the above problem.

By modifying the control programs, the data input from the encoder instead of being used for part programming could be used to establish a closed-loop positioning circuit. By doing so, part programming as usual prior to machining can be carried out. The part program for this four axis system should therefore include functions relating to the control of the rotational axis. This means the alteration of the present format and addition of new "word-address" data to the ordinary NC part program that on transfer is separated by the control routine and channelled to the stepping motor control.

## 7.2. BCSURF as a CAD/CAM Software Package

### 7.2.1. Design Philosophy

A major part of this work has been allocated to developing a prototype design and machining software package called BCSURF, short for B-spline Cardinal spline SURFace. The method employed is based on the so called computational geometry with particular reference to free form surfaces.

The incentive for developing computational geometry is largely due to the inadequacies of conventional techniques to handle the doubly curved surfaces and the need to minimise labour cost taking advantage of modern technology. In choosing a model for computer aided geometric design it is important to evaluate the utility of the power of a model and the control parameters. In addition the model must be compatible with the equipment upon which it is to be implemented. Although many design methods have been introduced in this field, hardware limitations has always been an obstacle on their successful implementations.

For years lofting has established itself as a practical method in design, and therefore, it was only logical that the importance of this technique be considered in making a decision on adopting a design philosophy. Lofting has successfully been used in this work and proved not only to be able to produce satisfactory result in design of a wide range of shapes, but also present no difficulty for shape representation using the existing two-dimensional cathode ray tubes, plotters etc.

### 7.2.2. Suitability of the Method for Design and Machining

The potentials of the B-spline basis in computer aided curve description could only be fully realised and appreciated when it is being applied in practice and primarily in the course of its development. In curve description, whether one is faced with the problem of design or representation, in other words, creating a new curve or approximating an existing one, B-spline basis provides simple algorithms which are computationally very efficient and easy to implement. It also gives multiple control parameters providing many degrees of freedom.

The general form of the B-spline curve description procedure used in design and fitting routines of BCSURF demonstrated to be well capable of fulfilling the requirements of surface design in a practical sense with the defining polygon being the main control handle. Ideally one prefers the control points to lie on the curve itself. This can, of course, be achieved by combining the inversion technique in the curve generating routine. In such a routine, however, any modification will require that the whole computation to be repeated. The use of polygon vertices as control parameter, nevertheless, showed to give a very good indication of the shape of the curve, and that it can effectively be used for subsequent modifications. Now, although the fourth order curve seemed to be best suited in most situations as it provides a compromise between smoothness and "localness", yet varying curve orders have a role to play when being used as a handle in design.

The routine for curve design allows space curves to be designed when the appropriate control parameters are specified and three-dimensional data is input. The experience, however, has shown that in the majority of cases one could limit oneself only to defining planar sectional curves, as judging the shape of a doubly curved contour on the two-dimensional graphics devices is not possible or at best easy.

It was found that the local property of B-splines for curve design is of great importance, because with lofting method cross-sectional curves can have very complicated shapes and it is often required to modify a small portion of these curves without altering the rest. Furthermore, designing such curves may need a polygon containing many sides. Compared to Bezier method, B-splines have the advantage that the degree of the curve is independent of the number of polygon vertices and therefore, difficulties involved with the Bezier curve does not occur here.

It was realised that in design of free form surface, the constraints presented are not purely aesthetic and therefore, fitting procedures were investigated and two routines both based on the mathematics of B-splines were developed to accommodate these. In dealing with large number of points, least squares method can provide a more satisfactory solution but it might involve guessing the position of vertices, and if the problem is to approximate an existing curve, inversion technique with its more efficient performance is suggested.

The most attractive feature realised with BCSURF is the simplicity of approach to design and the ease of determining the main input elements. In addition the mathematics of surface description used readily lends itself to providing a link between the design and manufacture ; consequently simplifying production of designed objects on NC machine tools. Here the analytical method of computing normals to the surface is more efficient than the numerical method, as both B-splines and cardinal splines bases allow the tangent vectors to be determined with very little extra computations in the surface design routine. In addition, one might argue that by offsetting the tool directly above the points specifying the surface, the resulting machined surface will more closely approximate the designed shape than in numerical method where the tool cuts between the points and leaves the ridges over the data points.

In BCSURF the object being defined in a bi-parametric space with B-spline sectional curve shaping the surface in one direction and cubic cardinal splines in the other, the behaviour of interpolating cardinal splines is an important factor in the outcome of the design. The result using BCSURF has, however, shown that cubic cardinal splines produce a smooth surface free from extraneous undulations, though the prediction of their behaviour between the control points is a matter for experience and a function of the position of sectional curves relative to each other.

### 7.2.3. Inadequacies of the Package

BCSURF is an experimental integrated design and manufacture software package developed to explore the problems of Computer Aided Design. It can be said that the main objectives of this work which was to study the required characteristics from the computer aided geometric design methods, and to examine the practical difficulties involved in their implementation have been accomplished. But the software produced for the present suffers from some functional shortcomings that puts demands on the user to acquire a more intimate knowledge of the package for a successful application. The main inadequacy being the non-interactive implementation of the software; consequently only through a time consuming process the result can be achieved.

Problems arising from parametrisation are also worth noting. It may be convenient or desirable, for purpose of machining or drawing, to interrogate a parametrically defined surface by working along isoparametric lines. Although this is undoubtedly a quick and simple method, does not always produce a satisfactory result, as it is required that the specified points be roughly equally spaced physically. In order to achieve this an iterative method has been adopted in BCSURF. This, however, has the disadvantage of being costly with regard to increased processing time and complexity. Further on the inadequacies of parametric method, given a position Z, one can not using parametric representation of the surface readily determine X and y at the cross-section through Z. To do so,again a lengthy iterative search is needed, and the package does not incorporate facilities for this purpose.

Lastly, BCSURF only caters for design and machining of a single patch. Although in many cases, using lofting, one patch could cover a complete component, but the high core memory required sometimes necessitates that a part to be divided into sub-sections having certain degree of continuity at the joints after assembly.

# CHAPTER EIGHT

## SUGGESTIONS FOR FURTHER WORK

### PART I

I) The functions of the computer in the DNC system include part programming and data preparation, data storage, and data distribution. One of the weaknesses of the present interface design is that once the part program is called on the computer, using the machine control console only a forward search on this data can be carried out. If for any reason, say in the course proving the tape, a reverse search is needed the whole process should be repeated from the start. With the rapid rate of reduction in price of micro processors and also memory chips, any future development of such systems should be approached somewhat differently. A microprocessor dedicated to a machine tool could be used as an intermediate data storage device. In addition this unit, put next to the machine, could be used for such functions as manual data input, on line editing and search, and format structuring to suit a particular machine. Already there are commercially available devices for retrofitting to the existing controllers for performing similar functions. These special purpose units or a general purpose micro-processor interfaced to the mini computer and connected to the machine control would give a much more practical and flexible system. The mini computer can therefore be used for part programming, design, mass storage of data, and also for file keeping.

II) It has been shown in this research that the capabilities of the hardwired NC systems can largely be enhanced by software manipulation. However, to gain the maximum benefit, the range of sub-programs written could be extended to cover a whole range of two and three-dimensional contours.

One of the areas that attracted some attention in the recent past and could be of much interest for future development is conversational or interactive part programming. Using such method, in a step by step manner, the user can input geometrical data relating to a path and immediately look at the resulted cutter centre-line or machined path on the VDU or plotter at his disposal. If it is acceptable, by moving to the next stage the post processed data will be generated, otherwise the input is modified until the desired path is produced.

PART II

I) This research has been successful in showing that the adopted design method and the mathematics used offer excellent properties for design and machining of curves and free form surfaces. However, complete benefit will only be obtained if the same principle is implemented in an interactive environment. The first step, therefore, for continuation of work in this line should be towards establishing the hardware basis upon which future software development will be carried out.

II) At the present an object designed with BCSURF can be machined by incrementally sinking the die in Z. Using a roughing tape can reduce the machining time to a great extent. The package to be of any practical value, inclusion of a routine for generation of roughing tape is essential.

III) The convensional methods for design of free form surfaces cannot produce an exact description of the object, both because of complexity of the shape and also because of the presence of the blended regions. Indeed it is only after the making of the pattern that one can say the object is fully defined. Consequently, at the design stage only approximate analysis of such features as stresses, volumes, weights, areas, cross-sections, etc., can be performed. Computer aided defined surfaces usually have the potentials to facilitate the analysis of the object at the design stage, programs therefore could be developed in conjunctions with BCSURF to cater for such requirements. One could also develop routines to aid determining the split lines when a three-dimensional object is to be cast in a two-part mould.

IV) As discussed in the previous Chapters, the parametric method although having great many advantages, presents some practical difficulties that limit its effectiveness in application. In order to overcome these problems, more investigations are needed in finding a relationship between parametric values, physical values, and the factors affecting the relationship. This investigation could be carried out for both B-splines and cardinal splines. Finally, on the question of curve fitting using inversion technique, there are many factors affecting the shape of the fitted curve. These being the position of data points, the number of vertices and the order of B-splines. Further study is required in determining the role of these elements and finding a relationship between them.

## APPENDIX I

Listing of the control programs for the DNC cell (CIN1, CIN2, CIN3), Users' guide, and examples of input or output where appropriate are given in the microfiche attached.

## APPENDIX II

Listing of the programs included in BCSURF, users' guide, and examples of input and printed output are given in the microfiche which also contains APPENDIX I.

## REFERENCES

1. Charnley, C.J. : Numerical Control, Present Status and Future Potentials of Numerical Engineering Methods, Part 6: Current Trends and Future Potentials. Engineer's Digest. Vol. 39, NO.5 May 1978, p 9-17

2. Hessey, M.F. : The Application of Mini-computers and microprocessors to Manufacturing Processes. Engineer's Digest, Vol. 39, No. 11 Nov. 1978, p 35-38

3. Wilkinson, D.G. : Links Between Design and Manufacture. National Engineering Laboratory, East Kilbride, Glasgow 1976

4. Deam, J.B. : Direct Numerical Control of Machine Tools. Machining and Prod. Eng. March 1973, p 319-323

5. Crossley, T.R. : A Decade of Direct Numerical Control. Annals of the CIRP, Vol. 27, No. 1, 1978, p 405-408

6. Milner, D. A., Brindley, J.D. : Hardware and Software Developments for a DNC Manufacturing Cell. Int. J. Prod. Res., Vol. 16, No. 6, 1978, p 441-452

7. Wightman, E.J. : Developments in Computer Control Systems for Machine Tools. Proc. Instn. Mech. Eng., June 1974 p. 49

8. Johnson, A.R. and Sturge, D.P. : The Duct System Design and Manufacture for Patterns, Moulds and Dies. Proc. of The 20th Int. Machine Tool Design and Research Conference. Sept. 1979. p 17-23

9. Clark, J.H. : Some Properties of B-splines. Second USA - Japan Computer Conference, 1975 Session 26. 1.1 p 452-454

10. Forrest, A.R. : II Current Developments in the Design and Production of Three Dimensional Curved Objects- Computational Geometry. Proc. Roy. Soc. Lond. A 321 1971 P 187-195

11. Riesenfeld, R.F. : Applications of B-splines Approximation to Geometric Problems of Computer-Aided Design. PhD Thesis,Syracuse University, U.S.A. Computer Science 1973

12. Ferguson, N.C. : A History of Numerical Control. British Numerical Control Society Vol. 10, No. 1, Feb. 1979, p 18-19

13. Olesten, O.N. : "Numerical Control" Wiley Interscience Pub., 1970

14. Simon, W. : "The Numerical Control of Machine Tools", Edward Arnold Pub. 1973

15. Hollingum, J. : Microprocessor Takes Over. The Engineer, Vo. 243, Oct. 1976, p 30-32

16. Wightman, E.J. : Computer Control System for Machine Tools. Chartered Mechanical Engineer March 1974

17. Martin, S.J.: " Numerical Control of Machine Tools" English University Press Ltd, 1970

18. Plas, J. and Blommaert, J. : A Stepping Motor Drive Assembly Especially Designed for DNC Systems. 13th International Machine Tool Design and Research Conference, 1972, p 183-189

19. Poulos, D.G. : Some Aspects of On-line Computer Control of a Lathe. PhD Thesis, Bristol Polytechnic, Dec. 1979

20. Puckle, O.S. and Arrowsmith, J.R. : "An Introduction to NC Machine Tools" Chapman and Hall Pub., 1968

21. Sweeney, G. and Bruce, J.W. : Prediction of the Future of Computer Controlled Machine Tools. Symposium on Computer Control of Machine Tools, Birnie Hill Institute. NEL, East Kilbride, Sept. 1971

22. Barnett, C. and Davies, B. : A Review of Developments. Symposium on Computer Controlled Machine Tools, Birnie Hill Institute, NEL, East Kilbride, Sept. 1971

23. Milner, D.A. and Oliver, A.F. : Computer Involvement in Numerical Control of Machine Tools. The Prod. Eng., Aug/July 1974 p 240-247

24. Savage, R. : The Influence of Microprocessors on the Evolution of Numerical Control Systems. 14th Int. M.T.D.R. Conf. 1973, p 143-147

25. Carter, W.A. : Computer Numerical Control Software. 14th Int. M.T.D.R. Conf., 1973, p 359-365

26. MTIRA Report : Getting the Most Out of CNC. The Prod. Eng., Oct. 1977, p 39-40

27. Cole, L. : Acramatic Improved Man/Machine Communication. The Prod. Eng., June 1975, p 367-369

28. Milner, D.A. and Rackic, M. : Software Program Implementation in DNC System. Autumn Control Conf. 15th Proc. University of Texas, Austin, June 1974, p 675-686, Pub. by Aicher New York

29. Barden, W., Jr. : "The Z-80 Microcomputer Handbook" Howard W. Sams & Co. Inc. 1978

30. Bolinger, J. : The Role of Microprocessors in Future of CNC Systems. Journal of the CIRP Vol. 25, No. 1. 1976, p 323-328

31. Kean, C.G. and Savage, R. : Microprocessor in Machine Operation. The Prod. Eng., Vol. 56, No. 4, April 1977 p 32-36

32. Van Brussel, H. and Simons, J. : Microprocessor in Hierarchical Control Systems. Annals of CIRP, Vol. 271, 1978 p 265 - 269

33. Pritty, D.W. : A Continous Path Microprocessor Numerical Control System. Dept. of Computer Science, University of Strathclyde, Livingstone Tower, Glasgow, 1979

34. Tipton, H. : Numerical Control. Frontiers of Technology, Engineer's Digest, 1979 p 69 - 74

35. Collins, J. A. : Integrated Manufacturing Systems - The World Scene. British Numerical Control Society, 14th Technical Conf. April 1980 p 1-9

36. Spur, G. : Flexible Manufacturing Cells in Multi Station Production - Realisation of a Modular Flexible Manufacturing System for Rotational Parts. Annals of the CIRP, Vol. 25, No. 1, 1976, p 329

37. Duncan, J. P. and Mair, S.G. : Polyhedral NC Program Documentation Report, Dept. of Mech. Eng., University of British Columbia, Vancouver, Canada, 1975

38. Allan III, J.J. : Foundation of the Many Manifestation of Computer Augmented Design. IFIP Working Conference on Principles of Computer Aided Design. Eindhoven, The Netherlands, Oct. 1972, p 29-58

39. Newman, W.M. and Sproull, R. : "Principles of Interactive Graphics", McGrawhill Book Company, New York, 1973

40. Ferguson, J.C. : Multivariable Curve Interpolation. Journal ACM, 11, 2, 1964, p 221-228

41. Curry, H.B. and Schoenberg, I.J. : On Polya Frequency Functions IV, The Fundamental Spline Functions and Their Limits, J. Analy. Math., 17, 1966, p 71-107

42. Coons, S.A. : Surfaces for Computer Aided Design of Space Forms. Report MAC-TR-41, Project MAC, M.I.T., 1967

43. Bezier, P. : Example of an Existing System in Motor Industry, The UNISURF System, Proc. Royal Society London, A321, 1971, p 207-218

44. Gordon, W.J. and Riesenfeld, R.F. : B-splines Curves and Surfaces."Computer Aided Geometric Design", Academic Press, 1974, p 95-123

45. Schweikert, D.G. : An Interpolation Curve Using a Spline in Tension. J. Math. & Phys. 45, 1966, p 312-317

46. Mehlum, E. : Curve and Surface Fitting Based on a Variational Criterion for Smoothness. Central Institute of Industrial Research, Oslo, 1969

47. Sabin, M.A. : An Existing System in the Aircraft Industry. The British Aircraft Corporation, Numerical Master Geometry System. PROC. Roy. Soc. Lond., A 321, 1971, p 197-205

48. Flutter, A. G. and Rolph, R. N. : POLYSURF: An Interactive System for the Computer - Aided Design and Manufacture of Components. CAD 76 Proceedings, 1976 p 150-158

49. Bloor, Susan, M.,De Pennington, A., Swift, J.S. and Tan, S.T. : Interactive Design and Manufacture of Sculptured Surfaces Using the B-spline Approach. Proceedings of the Twentieth International Machine Tool Design and Research Conference, Sept. 1979, p 3 - 15

50. Kishi, H. Oyake, I., Shimura, Y., Nagai, K. and Hatta, T. : The Okisurf System. Proc. 11th Ann. Meeting of Numerical Control Soc. Toronto Canada, March 1974

51. Forrest, A.R., : Mathematical Principles for Curve and Surface Representation. In Curved Surfaces in Engineering,

(Continued....)

51 (Cont.) Proc. Conference at Churchill College, Cambridge, 1972, IPC Science and Technology Press Ltd.

52. Higdon, A. Ohlsen, E., Stiles, W., and Weese, J. "Mechanics of Materials" 2nd Edition John Wiley & Sons Inc., New York, 1967

53. Schoenberg, I.J.: Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions. Quartery of Applied Math., V 4, 1946, p 45-

54. Faux, I.D., and Pratt, M.J.: "Computational Geometry for Design and Manufacture" Ellis Horwood Ltd., 1979

55. Nutborne, A.W. : A Cubic Spline Package, Part 2 - The Mathematics. Computer Aided Design, V5, No. 1, Jan. 1973, p 7-13

56. Adams, J.A., : A Comparison of Methods for Cubic Spline Curve Fitting. Computer Aided Design, Vol. 6, 1974, pl-9

57. Bezier, P.E. : "Numerical Control: Mathematics and Application" John Wiley & Sons, 1972

58. Forrest, A.R. : Curves and Surfaces for Computer Aided Design. PhD Thesis, University of Cambridge, 1968

59. Forrest, A.R. : Coons Surfaces. Appendix to Numerical Control by Bezier, John Wiley & Sons, 1972, p 228-235

60. APT Encyclopedia : Published by UNIVAC, 1963

61. Bezier, P.E. : Mathematical and Practical Possibilities of UNISURF. In Computer Aided Geometric Design, Academic Press, 1974, p 120-150

62. How to Use Nova Computer: Data General Corporation, Southboro, Massachusetts, USA, Oct. 1972

63. Townsend, R. : "Digital Computer Structure and Design" Newnes-Butterworths, London, 1975

64. Gordon, W.J. : Spline Blended Surface Interpolation Through Curve Networks. J. Math. Mech. 18, 10, 1969, p 931-952

65. The Gino-Graph Plotter Routines Users Handbook: Computer Centre, The University of Aston, 1980

66. de-Boor, C. : On Calculating with B-splines. J. Approx. Th., 6, 1972, p 50-62

67. Cox, M.G. : The Numerical Evaluation of B-splines. NPL DNAC4 Report, National Physical Laboratory, Teddington, Middlesex, Aug. 1971

68. Wilson, E.N. : Cubic Splines on Uniform Meshes. Communication of the ACM, Vol. 13, No. 4, April 1970 p 255-258

69. Hayes, J.G. and Halliday, J. : The Least-Squares Fitting of Cubic Spline Surfaces to General Data Sets. J. Institute of Math. and Its Application, 14, 1974, p 89-103

70. James, M.J., Smith, G.M., Wolford, J.C. : "Applied Numerical Methods for Digital Computation with FORTRAN and CSMP" Harper & Row Publishing Inc., 2nd Edition, 1977

71. Clark, J.H. : Designing Surfaces in 3-D. J. Communications of the ACM, Vol. 19, No. 8, Aug. 1976, p 454-460

72. Martin, E.N. : Man-Machine Communication in Three Dimensions. In Computer Aided Geometric Design, Academic Press, New York, 1974, p 303-315

73. Duncan, J.P. and Mair, S.G. : Polyhedral NC Machining Report, Dept. of Mech. Eng. University of British Columbia, Vancouver, Canada, 1975

74. Husemeyer, C., Wong, M.S., Duncan, J.L. and De Matherbe, M.C. : Programming a Point-to-Point NC Machine for Contouring Operation. 13th Int. M.T.D.R. Conf., 1972, p 171-176