COMPUTER AIDED DESIGN AND MANUFACTURE OF FORM-ROLLS

by

TAK-NAM WONG

MSc, AP (HK)

A Thesis submitted for the Degree of

DOCTOR OF PHILOSOPHY

to

THE UNIVERSITY OF ASTON IN BIRMINGHAM

Department of Production Technology and

Production Management

October 1983

THE UNIVERSITY OF ASTON IN BIRMINGHAM

COMPUTER AIDED DESIGN AND MANUFACTURE OF FORM-ROLLS

TAK-NAM WONG

(A Thesis submitted for the PhD Degree, 1983)

SUMMARY

Cold roll forming of thin-walled sections is a very useful process in the sheet metal industry. However, the conventional method for the design and manufacture of form-rolls, the special tooling used in the cold roll forming process, is a very time consuming and skill demanding exercise. This thesis describes the establishment of a stand-alone minicomputer based CAD/CAM system for assisting the design and manufacture of form-rolls. The work was undertaken in collaboration with a leading manufacturer of thin-walled sections.

A package of computer programs have been developed to provide computer aids for every aspect of work in form-roll design and manufacture. The programs have been successfully implemented, as an integrated CAD/CAM software system, on the ICL PERQ minicomputer with graphics facilities. Thus, the developed CAD/CAM system is a single-user workstation, with software facilities to help the user to perform the conventional roll design activities including the design of the finished section, the flower pattern, and the form-rolls. A roll editor program can then be used to modify, if required, the computer generated roll profiles. As far as manufacturing is concerned, a special-purpose roll machining program and postprocessor can be used in conjunction to generate the NC control part-programs for the production of form-rolls by NC turning.

Graphics facilities have been incorporated into the CAD/CAM software programs to display drawings interactively on the computer screen throughout all stages of execution of the CAD/CAM software.

It has been found that computerisation can shorten the lead time in all activities dealing with the design and manufacture of form-rolls, and small or medium size manufacturing companies can gain benefits from the CAD/CAM technology by developing, according to its own specification, a tailor-made CAD/CAM software system on a low cost minicomputer.

Keywords : Cold roll forming
form-rolls
CAD/CAM
single-user workstation
NC turning

A C K N O W L E D G E M E N T S

I wish to express my gratitude to everyone involved in the successful completion of this research. In particular, I would like to thank :

Dr D A Milner, my supervisor, for his guidance, advice and encouragement.

Hadley and Evans (Sections) Ltd, for sponsoring this research and financially supporting me throughout the course with a studentship.

Mr Geoff Deeley, Works Director of the company, and his colleagues, for their consistent interests and their assistance in the work.

Last but not least, Ann for typing the thesis.

CONTENTS

Page

Page

Page

LIST OF FIGURES, PLATES, TABLES AND CHARTS

Page

Figures (Cont/d)

Page

<u>Figures</u> (Cont/d)

Page

Figures (Cont/d)

Plates

Page

Plates (Cont/d)

Tables

Page

Tables (Cont/d)

Charts

CHAPTER 1

CHAPTER 1

INTRODUCTION

It has been said that the computer age will do for man's mind what the industrial revolution did for his muscle. In the past twenty to thirty years, developments in digital data processing systems have progressed rapidly and, presently, there are many application areas with a wide variety of implementation. Manufacturing companies were introduced to the computer in the early 1960's. The first applications were the recording of routine financial transactions, but gradually computers were applied to other tasks, such as the control of inventory, the scheduling of production and the routing of a part from one process to another on the factory floor. Besides, falling productivity, worsening inflation, and scarce engineering resources are accelerating the demand for manufacturing organisations to implement computer technology diversely and to support functions company-wide.

Tremendous breakthroughs in interactive computer graphics, engineering applications, and relatively inexpensive 'minis' and 'micros' have hastened the proliferation of the Computer-Aided Design (CAD) and Computer Aided Manufacture (CAM) technology. During the last decade we have seen the emergence of CAD/CAM technology as a viable alternative to the conventional design and manufacturing technology to meet the demands of current and future industrial production, characterized by the increases in complexity and variety of products, the need for high productivity, and complicated information management requirements [1].

The largest suppliers of CAD/CAM systems offer 'turnkey' products. These are complete systems comprising both hardware and software which can be operated by the user following installation, virtually 'at the turn of a key'. To date, there are various vendor-supplied 'turnkey' CAD/CAM systems available in the UK market, for example, those from Computervision, Calma, Applicon, IBM, Ferranti and so on [2].

However, there are several factors which make the small and medium size scale mechanical industries to prefer a versatile, intelligent, and dedicated tailored CAD/CAM system to 'turnkey' systems. Although for a company embarking on CAD/CAM to opt for these 'turnkey' systems, it does usually save time and costs in the short term, a firm which has developed a system in-house or had one developed to his own specification by an outside organisation is more able to become flexible in its operation, and can specify peripherals of his own choice. With this in mind and after judging the requirements of a leading West Midlands company in the sheet-metal section manufacturing industry, the Department of Production Technology and Production Management at the University of Aston in Birmingham has undertaken a project, in collaboration with the company, to develop computer-aided facilities for the design and production of form-rolls, the special toolings used in Cold-Roll

Forming of thinwalled metal sections.

This research is concerned with the development of an integrated CAD/CAM system, being implemented into a low cost stand-alone minicomputer based single user workstation, to support a multitude of functions for the automation of design, design draughting, and NC part programming of form-rolls. In essence, the following activities are considered:

1. A study on the production of form-rolls by CNC.

2. Implementation of the CAD programs on a stand-alone minicomputer based CAD/CAM workstation.

3. Design and development of a roll editor program with the interactive graphics facility.

4. Design and development of a special purpose processor and postprocessor for the generation of NC tapes for machining the form-rolls using a NC turning machine.

CHAPTER 2

CHAPTER 2

REVIEW OF COLD ROLL FORMING

2.1 INTRODUCTION

Cold Roll Forming of thin sections is a very useful process in the sheet metal industry. In definition [3, 4, 5, 6], cold roll forming is a process for forming metal sheet or strip stock into desired shapes of uniform cross section by feeding the stock longitudinally through a series of roll stations equipped with contour rolls (sometimes called roller dies) - two or more rolls per station. Most contour roll forming is done by working the stock progressively in two or more stations until the finished shape is produced.

The applications to which cold roll formed sections are put are widely diversed. Generally, all metals that can be successfully formed by other common forming processes can be successfully cold roll formed [4]. The materials used in the manufacture of cold roll formed sections include hot and cold finished carbon steel, stainless steel, aluminium and its alloys, copper, brass, bronze, and zinc etc. However, stainless steel and aluminium are often roll formed for decorative and for architectural applications - often starting as pre-painted, plated, anodized or polished stock. The gentle action of the progressive bending rolls does not usually mark the prefinished materials.

Dissimilar strips of metals, or of metals and non-metals, can be fed into a roll forming machine at the same time, to produce composite sections with multiple properties, such as corrosion resistance or decorative finish with strength [6]. Normally metal as thin as 0.1 mm (0.005 in) and up to 20 mm (0.75 in) of thickness can be satisfactorily formed [7, 8], and the raw materials used may come in the form of coil stock or flat strips or sheets.

2.2 ADVANTAGES OF COLD ROLL FORMING

Some of the advantages of cold roll forming are as summarised in the following:

(a) The main advantage of this process is its high production capacity. For simple shapes, a production rate of 900 to 1500 m (3000 to 5000 ft) per working hour can be achieved.

(b) Roll formed products are essentially uniform in cross section.

(c) Virtually any profile can be produced by the process.

(d) Wide range of ferrous and non-ferrous metals can be readily formed by the process.

(e) Pre-coated materials can be roll-formed without damaging the surface finish.

(f) The sections can be produced to exact lengths thus reducing waste.

(g) High strength to weight ratio of the sections, especially thin-walled sections, means greater value and less expenses.

2.3 ALTERNATIVE PROCESSES

The press-brake and extrusion processes are the two main alternatives to cold roll forming. The press-brake process is not as versatile in terms of the section it can cope with, and is usually suitable for short-run production items. Extrusion, however, involves high tonnage presses, introduces lubrication problems and is usually carried out with material in a hot or warm state; it is, of course, fundamentally different to cold roll forming in that it transforms solid rather than sheet stocks.

2.4 ROLL FORMING MACHINES

A typical roll forming machine (Plate 2.1) usually consists of a fabricated base on which a number of roll stations are successively mounted. Each roll station contains two roll spindles, for holding a pair of rolls horizontally. Roll spindles are supported by housings either

(a) at one side, as in outboard type machines (fig. 2.1), or

(b) at both sides, as in inboard type machines (fig. 2.2).

The outboard type of machine, of which overhung roll spindles are easily accessible and the form rolls can thus be set up more easily, is usually used in the forming of narrow sections. The inboard type machines, on the other hand, supporting the spindles at both ends, are capable of handling much thicker and wider strip and sheet materials in forming most structural shapes. Plate 2.2 shows the inboard type roll stations. For this type of roll stations, the distance between the mounting can be adjusted and so are the heights of top and bottom rolls, so as to cope with various stock widths and roll diameters. Occasionally, when forming intricate shapes, one or two side-rolls may be mounted vertically to blend to the top and bottom rolls in the same roll station, to help to perform part of the forming operation.

For producing a particular finished section, a roll forming machine has to be selected according to its table width, the maximum number of roll stations it carries and the size of the stations. For forming large sections of thicker materials, roll stations must be large enough to accommodate rolls of large diameters. More roll stations, on the other hand, will reduce the amount of forming work required by each pair of rolls in each roll station.

The rolls of each station are usually power driven through a system of gears and have their peripheral speeds synchronised. The speed of forming or the speed of material travel may vary from station to station and usually that too requires synchronisation. Roll surfaces are usually lubricated to reduce friction between roll surface and material surface so as to maintain a good surface finish of the material and also to cool both the rolls and the material which may expand as a result of heat generated during forming.

2.5 SOME ASPECTS OF ROLL DESIGN

Cold Roll Forming of each shape presents individual problems in roll design, and the success of the cold roll forming process is mainly dependent on the shape of rolls used. However, there is a lack of scientific principles for roll designers to use. In fact, the roll design process requires experience, good judgement and a knowledge of the bending of metals. Thus it is not surprising that people treat the designing of rolls as an art rather than an exact science.

The final cross-section of the metal strip after forming, which is the product of the whole roll forming process, is called the finished section (Fig. 2.3). Design of the finished section is basically application orientated and is a job for the section designers [9]. However, it should be understood that the geometry of the finished section comprises only linear and circular segments.

2.5.1 Determination of Strip Size

Given the finished section drawing of the desired cold roll formed product, it is necessary to calculate the strip size, or width of the raw material required. Firstly, the meanlength of each individual element, either linear or circular, has to be established. This strip size S, for m linear elements and n circular elements, can then be calculated using the equation:-

$$S = \sum_{i=1}^{m} l_i + \sum_{j=1}^{n} r_j \theta_j \qquad (2.1)$$

where l_i is the meanlength of individual linear element,

r_j is the meanlength of individual circular element,

and θ_m is the final angle of bend (in radians) of the jth circular element.

The meanlength of an element, as a matter of fact, refers to the length of that part of the element which remains constant throughout forming.

Theoretically, linear elements do not undergo deformation in any way during forming since bending does not occur in them. For most practical purposes, such an assumption is valid even though it is not strictly true. It is possible for slight deformation to occur in linear elements situated adjacent to circular elements being bent as a result of the rolling action.

The determination of the meanlength of circular elements (or called 'Bending Allowances'), on the other hand, is not so straightforward. As shown in equation 2.1, the meanlength of a circular element is based on its mean radius, the radius of the element measured from its centre point to the neutral-axis. Different methods and different empirical formulae have been recommended by different experts in calculating the mean radius [3, 4, 5, 9]. For instance, Angel [3] recommended the following formula:-

$$BA = (0.01743R + 0.0078T)A \qquad (2.2)$$

where BA = the bend allowance,

R = the inside radius of the bend,

A = the angle of bend in degrees,

T = the material thickness.

Besides, the American Society for metals [4] recommends the following:-

For $r = 0$

$$r_m = r + t/3 \quad \text{(normal metal)} \qquad (2.3)$$

$$r_m = r + t/2 \quad \text{(less formable metal)} \qquad (2.4)$$

For $0 < r < t$,

$$r_m = r + t/3 \qquad (2.5)$$

For $r < 2t$,

$$r_m = r + t/2 \qquad (2.6)$$

where r_m is the mean radius

r is the inside radius of bend,

t is the material thickness.

However, the method of calculation used by the company is similar but is based on different criteria:-

$$r_m = r + kt \qquad (2.7)$$

where k is a factor based on the magnitude of the angle of bend (Table 2.1)

Angle of Bend	0^{o} to 80^{o}	Above 80^{o} to 100^{o}	Above 100^{o}
k	0.3	0.4	0.5

Table 2.1 Mean Radius Factor Based on Angle of Bend

2.5.2 Forming Sequence

With a section design being available, the first consideration for the roll designer is to determine the number of roll passes required to produce the desired shape. A 'Forming Angle Method', suggested by Angel [3], has been found to be a dependable guide for the determination of the number of roll stations required for a particular bend (Fig. 2.4). The strip length over which the bend is to be completed is determined from the following:

$$F = H (\cot \phi)$$

where F = forming length,

H = height of the leg to be bent up,

$\phi = 1^{o}\ 25'$ for carbon steel.

The number of stations needed is F, the forming length, divided by the horizontal centre distance of the roll forming machine, rounding up if necessary. The forms for intermediate rolls are then decided on the basis that bend height should increase linearly with length position as depicted in Fig. 2.4. For multiple bends on complicated shapes, the section must be broken down and each bend or pair of bends calculated individually to determine the number of passes required to perform that bend. Then, after combining passes wherever possible, the total number of passes can be determined.

Some roll design engineers, however, does not believe the above described 'Forming Angle Method' can always give optimum results, and there are in fact objections to this approach [10]. Thus, taking into consideration of those material behaviour such as springback, thinning, stretching, and persistence of flow; roll designers generally use their experience to determine the number of stages required.

After determining the number of stages, the roll designer then consequently determines the sequence and degree of bending at each roll station for the circular elements. The forming sequence is essentially a series of transitional shapes of the section which the strip progressively assumes from one stage of forming to another, and the centre-line (or neutral-axis) representation of these shapes is called the Flower Pattern. A typical flower pattern is shown in Fig. 2.5.

2.5.3 Wire Templates

With the required forming sequence, which is in the form of the flower pattern, a set of wire templates (Plate 2.3) can then be designed and manufactured accordingly. The wire templates are created as intermediate simulated sections of the strip at each successive forming stage and are used subsequently as the master contours or shapes in roll machining. Hence they in fact form an integral part of roll design and manufacture.

Base on the flower pattern, the shape of each element bent to the required angle at a particular forming stage is worked out and all of the element shapes are then grouped together to form the required section shape at that stage. The simulated section shapes are usually drawn ten times as large as the actual size. With the help of a shadow-graph projector, or perhaps some other similar equipment, straight wires are bent to the exact simulated shapes according to the 10-1 template drawings (Plate 2.4).

2.5.4 Design of Rolls

Concerning the rolls at each rolling station, it is necessary to decide on things like whether to use whole rolls or split-rolls, whether to use side-rolls, idler rolls or guides, the number of ironing stations and so on. Suitable allowance and adjustments to the roll contours must also be incorporated for smooth forming.

Having finalised the roll design, a set of roll drawings should then be produced. However, as a common practice, only simplified roll drawings are produced and used. The simplified roll drawings carry information such as the overall roll dimensions, roll materials, general tolerances for machining, pinch-difference surface allowances, the use of side-rolls or spigots and approximate shape of the rolls. Each roll drawing corresponds to each forming stage and is used to complement the wire template during machining of the rolls. Roll diameters are adjusted at the time of drawing to give a smooth variation of strip pass heights from station to station for better forming. Other means of guiding the strip flow are also incorporated at the same time if necessary.

2.6 MANUFACTURE OF ROLLS

The rolls are normally machined manually on conventional lathes (Plate 2.5). The accuracy of the roll contours so produced relies greatly upon the skill of the machinist whose only guides are the wire templates and the simplified roll drawings. If the shape of the roll is complex, a high level of skill is required and the procedure is very time consuming.

With the method described so far, detail roll design drawings are not necessary, only simplified drawings indicating the overall roll features are used. With some firms, the more elaborate approach of using detailed drawings for roll design without the use of wire templates is preferred and with others the use of other forms of templates or contour-tracing techniques are adopted instead. Despite the differences, one thing in common, that considerable manual skill is essential in manufacturing the rolls on conventional machines.

2.7 OTHER TOOLINGS AND ACCESSORIES

To aid forming operations, accessories may sometimes be used. Strip guides (Plate 2.6) are normally used before and in between the roll passes to keep the material in the right track. Inter-station idler-rolls may be used for the similar reason and in addition for doing partial forming on the material as well. A straightening device may be placed immediately after the last roll forming station to straighten the section. If coiled stock is used for forming while products of fixed length are required, then an additional cutting-off operation with a special machine (Plate 2.7) will be included at the end of the entire forming process.

2.8 AUXILIARY OPERATIONS

Auxiliary operations which can be incorporated in Cold Roll Forming include the usual press tool operations like piercing, blanking and notching, sweeping the finished section into circular radius if curved sections are required, seam welding for forming closed tubes and so on.

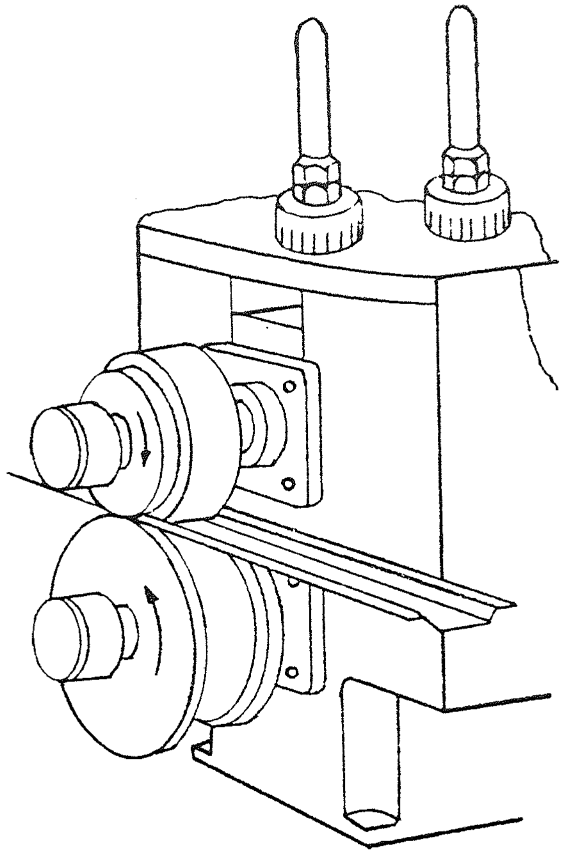


Fig. 2.1 ROLL STATION IN AN OUTBOARD TYPE ROLL FORMING MACHINE

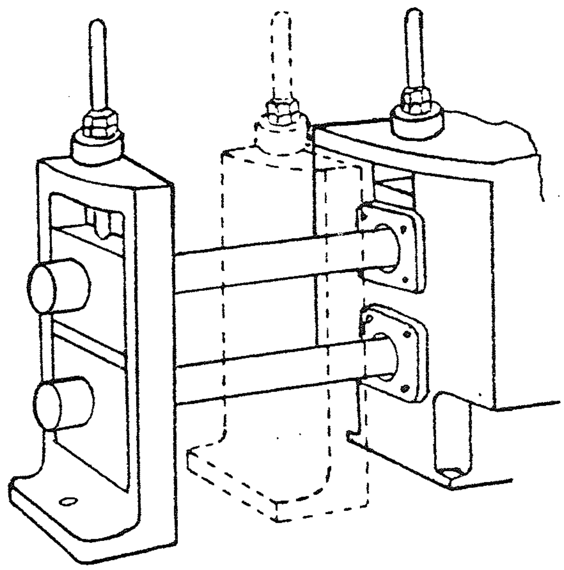


Fig. 2.2 ROLL STATION IN AN INBOARD TYPE ROLL FORMING MACHINE

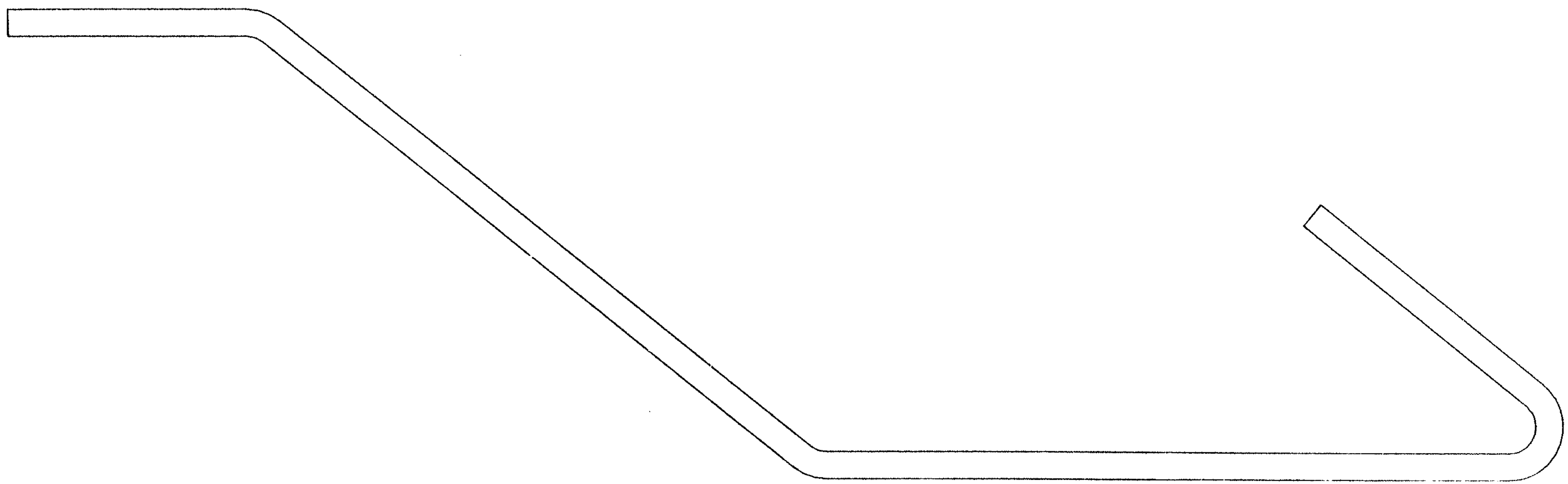
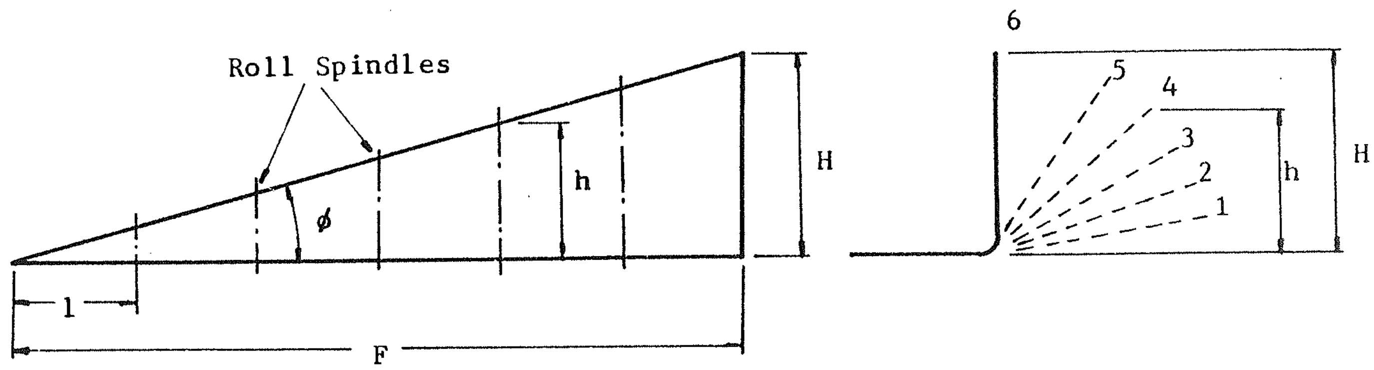


Fig. 2.3 A TYPICAL FINISHED SECTION



$$F = H \cot(\phi)$$

Fig. 2.4 THE FORMING ANGLE METHOD

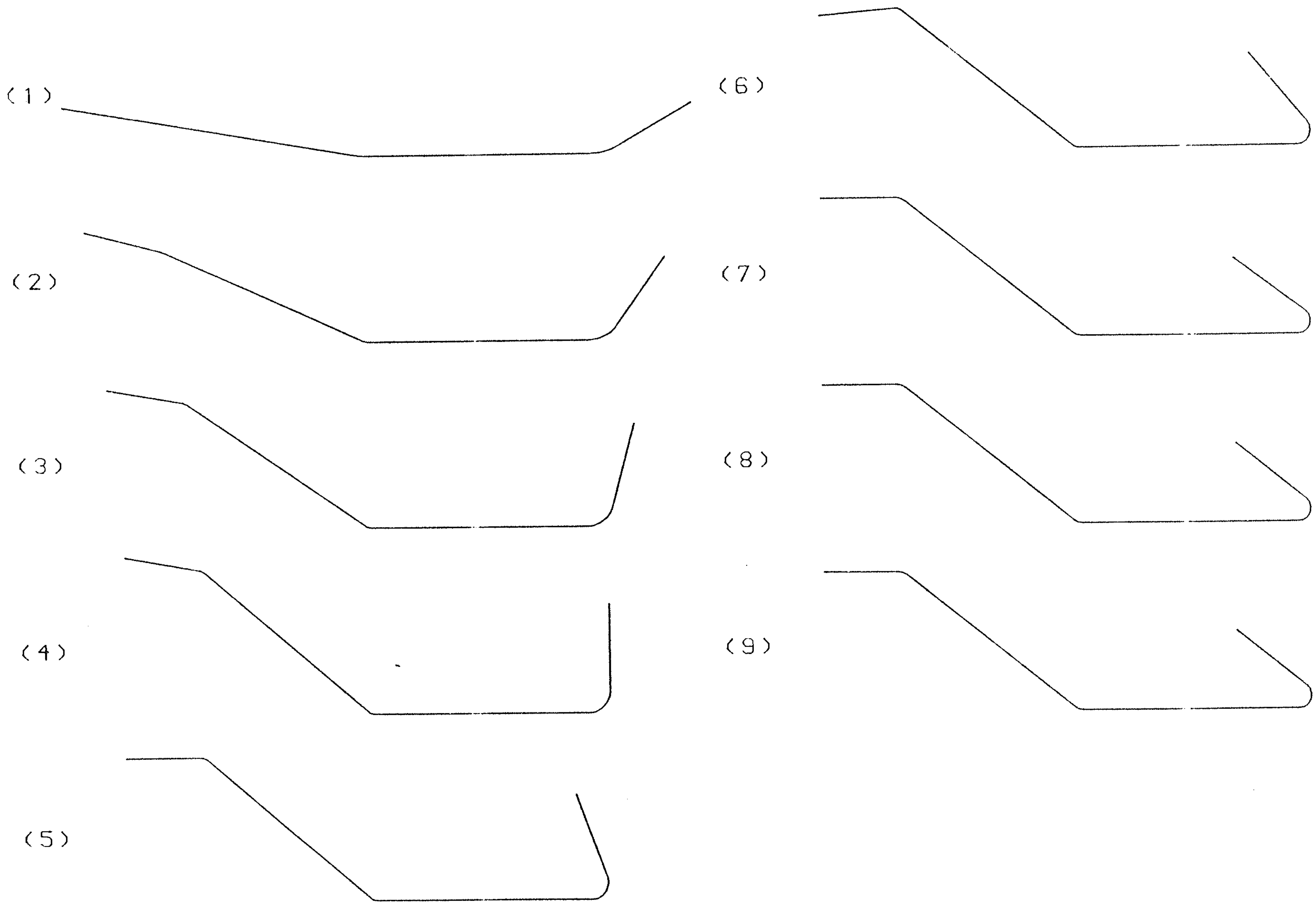


Fig. 2.5 A TYPICAL FLOWER PATTERN

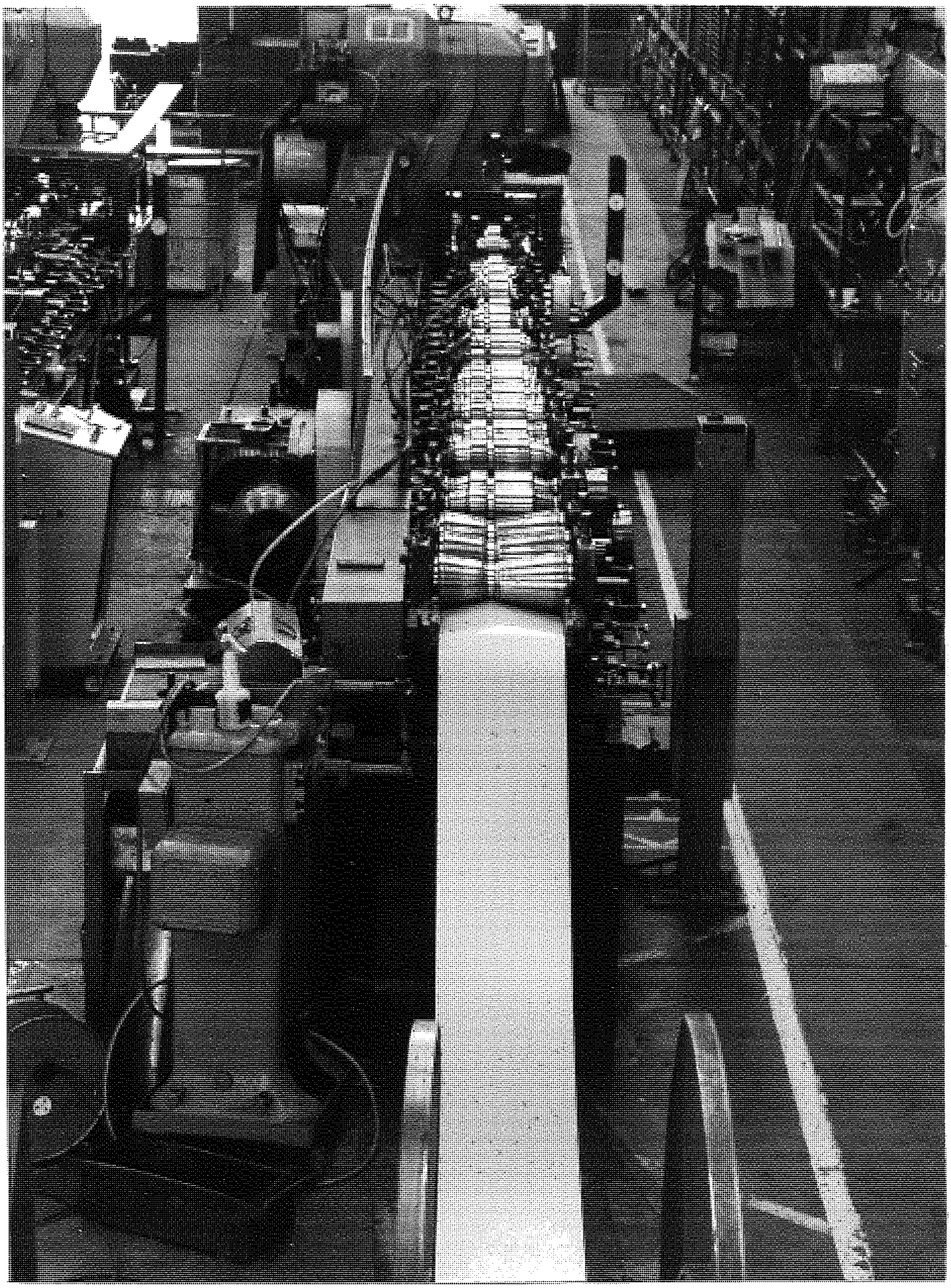


Plate 2.1 A TYPICAL ROLL FORMING MACHINE

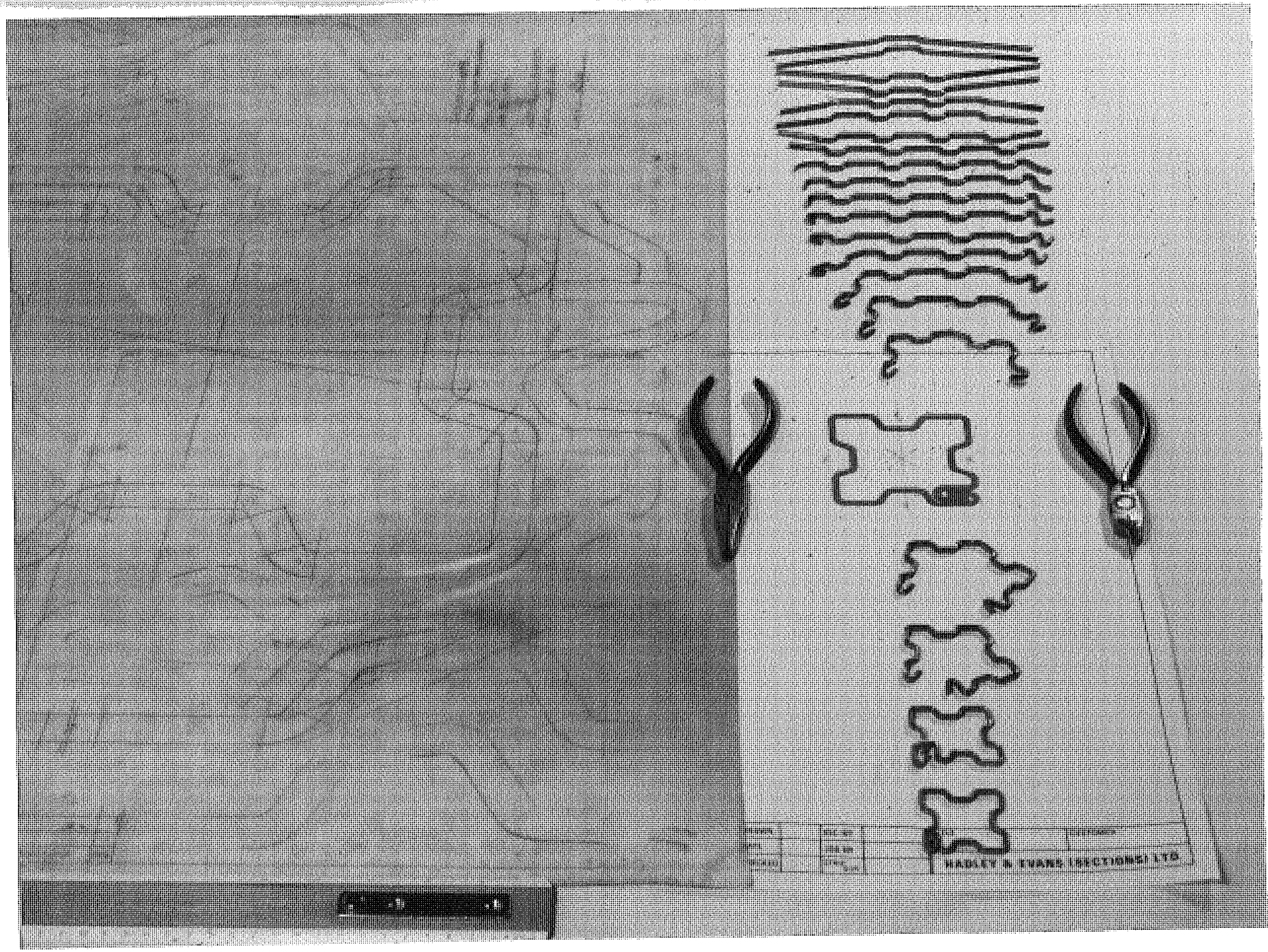


Plate 2.3 THE WIRE TEMPLATES AND A 10 TO 1 TEMPLATE DRAWING

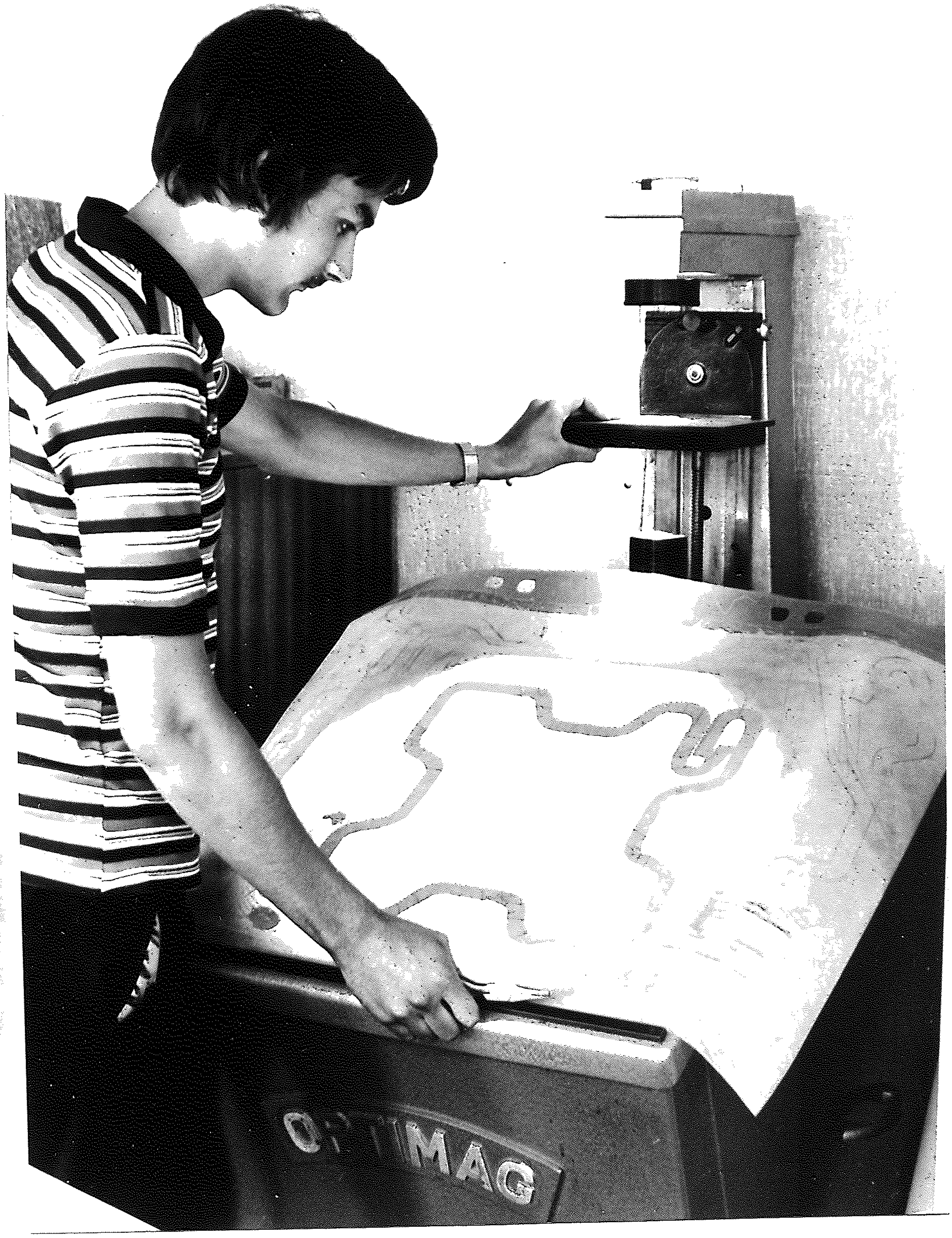


Plate 2.4 PRODUCTION OF THE WIRE-TEMPLATES ON A SHADOWGRAPH PROJECTOR

Plate 2.5 FORM-ROLL MACHINING ON A CONVENTIONAL LATHE

Plate 2.6 A STRIP GUIDE

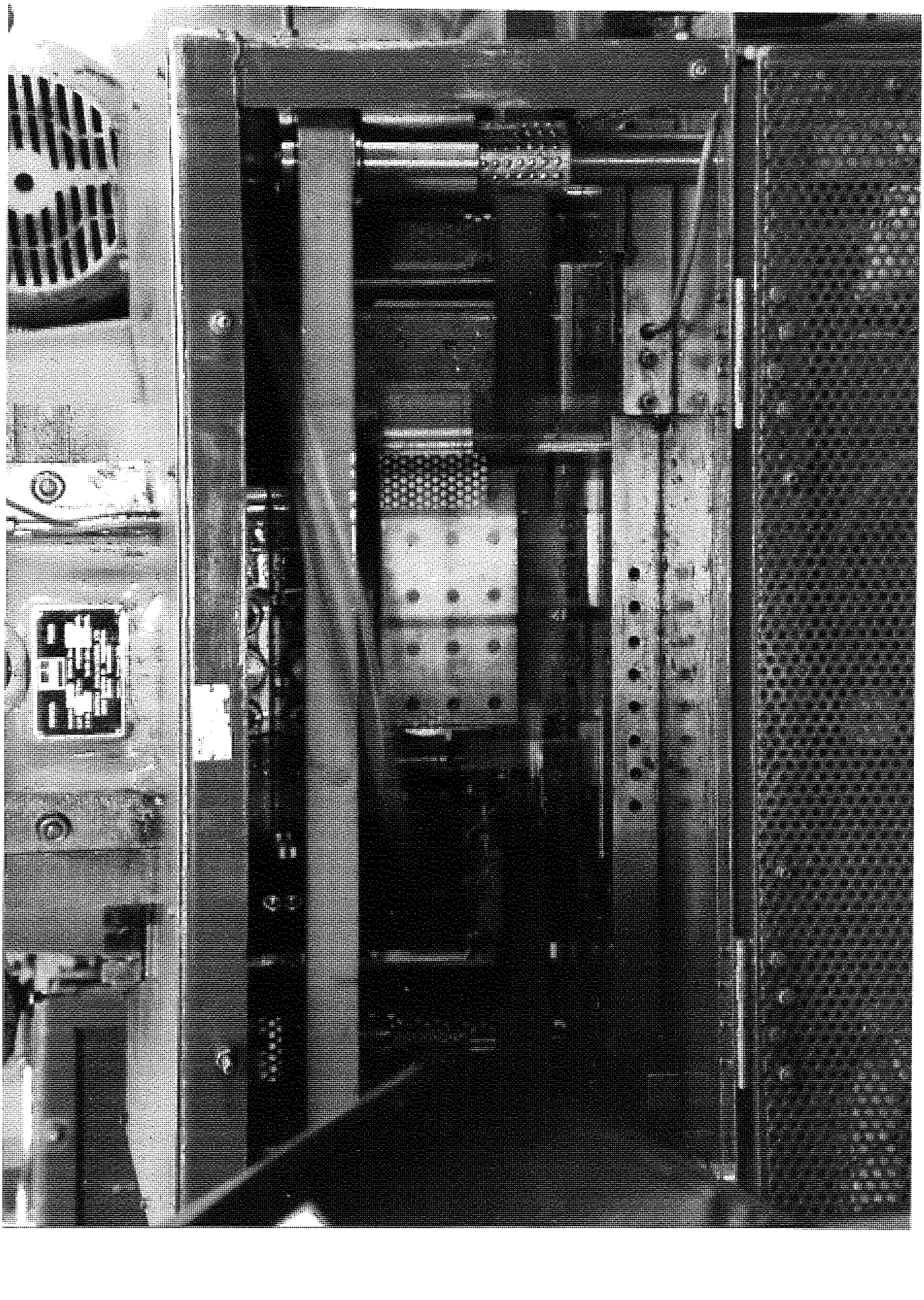


Plate 2.7 THE CUTTING-OFF OPERATION

CHAPTER 3

CHAPTER 3

THE CAD/CAM TECHNOLOGY

3.1 INTRODUCTION

Computer applications in manufacturing companies are now widely spread. Due to the development of microelectronics, computers have become far cheaper, more sophisticated and more powerful. In the present competitive commercial environment, the use of computer technology in the design of parts and production processes is perhaps the most promising technique for increasing productivity.

Both Computer-Aided Design (CAD) and Computer Aided Manufacture (CAM) technologies are currently advancing very rapidly in both their scope of influence and their technological expertise. The present CAD and CAM technologies have come about from firstly the development of NC machine tools during the World War II; and secondly the work of Ivan Sutherland in 1963 on 'SKETHPAD' [11], the world's first computer graphics system. To date, CAD is used to assist the product design engineer to develop designs, analyse them and graphically describe them. CAM, on the other hand, assists the production engineer to plan the method of manufacture, design tools, fixtures, test rigs and gauges, prepare factory plant layout and programme Numerical control (NC) machine tools and measuring machines.

3.2 THE CAD/CAM SPECTRUM

The CAD and CAM technologies are now being combined into unified CAD/CAM systems, where a design is developed and the manufacturing process controlled from start to finish with a single system. A design engineer can define a component, analyse stresses and deflection, check its mechanical action, and automatically produce engineering drawings - all from the same graphics terminal of a CAD system. Furthermore, production people can draw upon geometric description provided by CAD as a starting point to create NC tapes, determine process plans, instruct robots, and manage plant operations with CAM. The days of totally unmanned factories supervised by computers are in fact not very far off. The entire future commitment of computers in engineering design and production, as well as in many other spheres, is currently being enthusiastically pursued on an international scale.

As CAD/CAM is changing so rapidly, it is not surprising that there is not yet a precise definition for this new technology. There are some who regard CAD/CAM as a system which allows one to design mechanical parts to be manufactured on an NC machine. At the opposite extreme are those who include all engineering tasks performed with a computer as CAD/CAM (Fig. 3.1). In the author's point of view the CAD/CAM spectrum should not be limited by draughting engineering drawings and preparing NC tapes. Instead, a CAD/CAM system, according to the technical and economical feasibility of the organisation, can be built up gradually with one problem area and then move to the computerisation of other sectors.

3.3 THE EVOLUTION OF CAD

Prince [12] has traced the beginning of CAD to the work done by Sutherland [11] on his SKETCHPAD IN 1963. Since then, many efforts have been spent in exploring the mathematical domain for suitable theories and means of representing and generating geometric shapes, in the form of curves and surfaces, from which computerised design methods for specific applications may be developed. Typical pioneers in the field such as Coons [13], Bezier [14], Forrest [15] and many others had helped to firmly establish the rightful role of CAD in future.

3.3.1 CAD Hardware

Sutherland's SKETCHPAD [11], as an interactive graphics system, consisted of a Cathode Ray Tube (CRT) oscilloscope driven by a Lincoin TX2 computer, the user could then use the light pen or keyboard as the input device, to communicate with the computer and manipulate the pictures drawn on the screen. Systems of this kind, based on large mainframe computers, were expensive and thus were adopted mainly by the automobile and aerospace industries [14, 16]. Recently, due to the development of the relatively low cost minicomputers and microcomputers, the cost of computer memory has been reduced drastically. Concurrently, there has been great progress in the development of low cost CRT display for the Visual Display Unit (VDU) for use in computer graphics. As a matter of fact, these reductions in hardware costs have brought about the great availability of computer graphics and CAD in the present moment.

To date, there is a large variety of CAD workstations available in the market [17], ranging from those vendor-supplied CAD/CAM turnkey systems based on stand-alone minicomputers or time-sharing mainframes, to the microprocessor based inexpensive desktop graphics [18].

Fig. 3.2 depicts an interactive graphics system for CAD applications. In general, a typical interactive graphic design station configuration includes a processor, a graphics display, input devices, and hardcopy ouput devices. The processor is the computer in the system which is used to drive all the peripheral equipments and run the programs. As mentioned before, central processors ranging from mainframes to desktop microcomputers are being used in different systems. The most visible part of the system is the graphics terminal. There are three main types of CRT used in graphics display terminals, namely, the storage tube which maintains a steady image on the screen, was introduced by Tektronix in the late 1960s; the refresh tube in which the picture is rewritten on the screen at a rate of between 10 and 60 frames a second; and the raster scan which uses techniques similar to those employed in a domestic television. On the other hand, input devices such as light pen, digitising tablet with pen, joystick, or keyboard can be used; while hardcopy ouput devices can be a printer and a plotter. Available in the present market are highspeed drum plotters or flatbed plotters using pens, electrostatic plotters, and microfilm plotters.

3.3.2 CAD Functions

Briefly, CAD functions may be grouped into four categories: geometric modelling, engineering analysis, kinematics, and automatic draughting.

The geometric model is literally the most critical feature of any CAD/CAM system. Many other CAD/CAM functions, such as finite-element analysis, automatic draughting and NC tape preparation, are depending on the geometric data of the model as a starting point. Ever since the emergence of the CAD concept, a lot of work has been done in developing the geometric model of a part [19]. Most modelling today is done with wire frame models with 2D, $2\frac{1}{2}$D or 3D capability. However, the more sophisticated 3D solid modelling technique has been developed to obtain a better representation of the part shape [20, 21, 22, 23, 24].

After the geometric model is created, some CAD systems can move directly to analysis, calculating the weight, volume, surface area, moment of inertia, or centre of gravity of a part. In some cases, by specifying material properties and the loading and boundary conditions, the CAD system can then generate the finite element model from a geometric model [25].

Besides, some CAD systems have kinematic features for plotting or animating the motion of linkage mechanism. Such analysis can ensure that moving compounds do not impact other parts of the structure.

On the other hand, with automatic draughting, detailed engineering drawings may be produced automatically, with automatic scaling and dimensioning features. The geometric data can be retrieved from the data base and from a menu with drawing functions such as size and location of lines, arcs, text, cross-hatching, and dimensions.

3.4 CAM IN GENERAL

In the broader sense, CAM refers to the computerisation of all the relevant activities in the manufacturing environment. However, activities in the present CAM technology may be divided into four areas: numerical control, process planning, robotics, and factory management.

3.4.1 Nummerical Control

The first NC machine tool was demonstrated in 1952, following the project developed in the Massachusetts Institute of Technology.

Since then, NC has been playing a substantial role in the manufacturing industry. Indeed, the early CAD systems were originally developed to incorporate and exploit the NC technology for manufacturing complex shapes [13, 16, 26]. The earliest computer involvement in NC was the preparation of control tapes using the Automatic Programmed Tool (APT) language. Then in the 1970s, more advanced systems were developed, using minicomputers in place of the hardwired logic in the control systems of NC machines, known as Computer Numerical Control (CNC). Further enhanced systems were implemented as DNC (Direct Numerical Control), where a number of machines were linked together and controlled by a computer. The concept of Flexible Manufacturing System (FMS), in fact, was evolved from the idea of adding automatic material handling facilities to a group of DNC machines [27, 28].

3.4.2 Process Planning

Applying the concept of group technology, which organises parts into families, process planning can be done with the same CAD/CAM system [29, 30]. In this regard, the information stored in the system that specifies the geometric design of a part and its manufacturing stage will be retrieved for developing the process plan of a similar part. That is to say, according to the classification system of the company, the process planner can retrieve from the computer, a list of old components that have some of the same characteristics of the new part. He can then plan for production of the new part simply by specifying that the manufacturing process is to be the same as that for the old part, with any difference noted. An example of this kind of system is the AUTOCAP (31) system for computer-assisted planning of turn parts.

3.4.3 Robotics

In short, a robot is a reprogrammable mechanical manipulator. In CAM systems, robots can be used to perform a variety of material handling functions. Robots may select and position tools and work-pieces for NC machine tools. Or they may use their 'hands' to grasp and operate tools such as drills and welders (32).

Industrial robots are being used increasingly as machine tool part loaders, often in conjunction with computer-controlled machining facilities. The use of robots to feed machine tools is seen as offering many advantages compared with custom-engineering loading devices incorporated into machine tools. This kind of load/unload robot systems are increasingly being integrated with machine tools. Automated machining systems can be constructed by combining a robot and a lathe, drilling machine, or a machining centre. The role of robots in such configurations is in loading/unloading of workpieces, tool changing, and swarf removal (33). This trend of robot application is now being developed as a key element in CAM systems.

3.4.4 Factory Management

Computers can be used to provide a variety of management aids in a manufacturing system. The production management aspects of such things as Material Requirements Planning (MRP) and scheduling has been well developed and refined [34, 35].

A central feature of computerised activity in production management is to integrate different management functions together. In fact, computerised factory management is more effective when it is implemented as a comprehensive system, embracing a wide range but closely inter-connected functions.

In the present computer software market, a wide range of software is available, for carrying out functions such as production control, inventory control, capacity planning, Material Requirements Planning (MRP) or the more enhanced Manufacturing Resource Planning (also MRP) and so on [36, 37]. Most of these individual packages can be implemented as modules in comprehensive commercial systems. In particular, Manufacturing Resource Planning can save much of the cost of financing inventories. Computers and MRP help plan every aspect of the production cycle, including material acquisition, worker assignments, machinery schedules and various inventory requirements.

3.5 THE INTEGRATION OF CAD AND CAM

The effectiveness and benefits of CAD and CAM, however, can be improved by linking individual tasks together, into an integrated CAD/CAM system, also termed by some experts as CAE (Computer Aided

Engineering) system. With an integrated CAD/CAM system, different design and manufacturing functions can be executed, by getting access to a common data base or data file. As a matter of fact, there is a high resemblance between the output from the geometric design process and the input for the manufacturing process. Thus, it is the idea of an integrated CAD/CAM system to ensure that any work carried out in the design process is not needlessly repeated during the stage of manufacture.

In general, the simplest way to achieve CAD/CAM integration is to generate NC part programs after the completion of the CAD process, while the more rigorous approach is to prepare the manufacturing process plan with the same system.

Traditionally, when preparing the control tapes manually, NC part programmers are required to identify and transfer the geometric data from the detailed drawings. However, Computer-Aided Part Programming systems have been developed to assist NC part programmers. In such systems the data processing capabilities of a digital computer are utilised to improve the efficiency and accuracy of part program preparation. Since the introduction of APT, many other systems have been developed and implemented and commonly used. Some of these systems are EXAPT, GETURN [38], MELTS [39], COMPACT II [40] and so on.

Due to the availability of APT and other Computer-Aided Part

Programming languages, a lot of CAD systems can then be extended into the CAM area by generating part programs for one of these systems, the actual NC machine tool control programs can then be generated by the processor and post-processor of the Computer-Aided Part Programming system. An example of this approach is the work done by Craig [41] for the design and manufacture of moulds for civil engineering applications.

On the other hand, with regard to process planning, although a number of systems have been developed, there is not yet a general purpose process planning program which can be widely applicable and easily transplanted. Apart from the group technology approach as mentioned in Section 3.4.2, some other reported systems are AUTAP [42], TOM [43], CAPSY [44], CAM-X [45], and others [30].

Furthermore, the more advanced way of integrating CAD and CAM is, for instance, to link the activities of industrial robots to the design and machining functions in Flexible Manufacturing Systems (FMS) [46]. This level of CAD/CAM integration, whereby all design and manufacture functions are linked together by the single computer system, would lead ultimately to the automated factory.

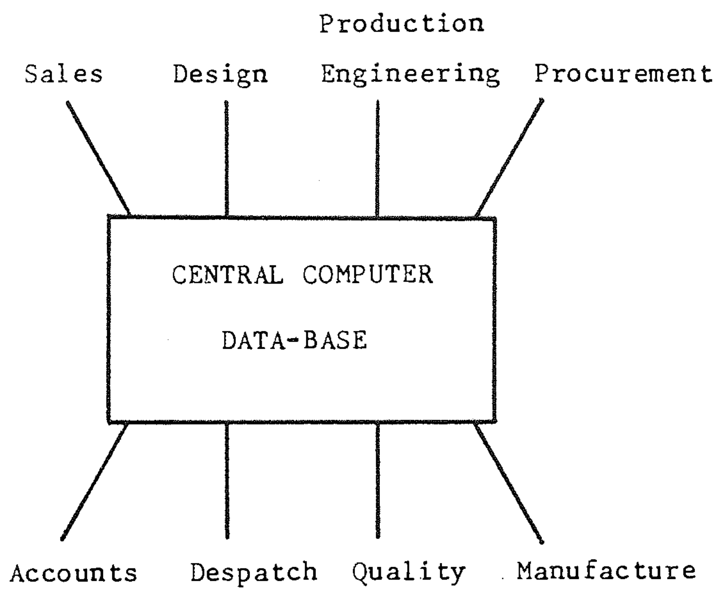


Fig. 3.1 A CAD/CAM SYSTEM

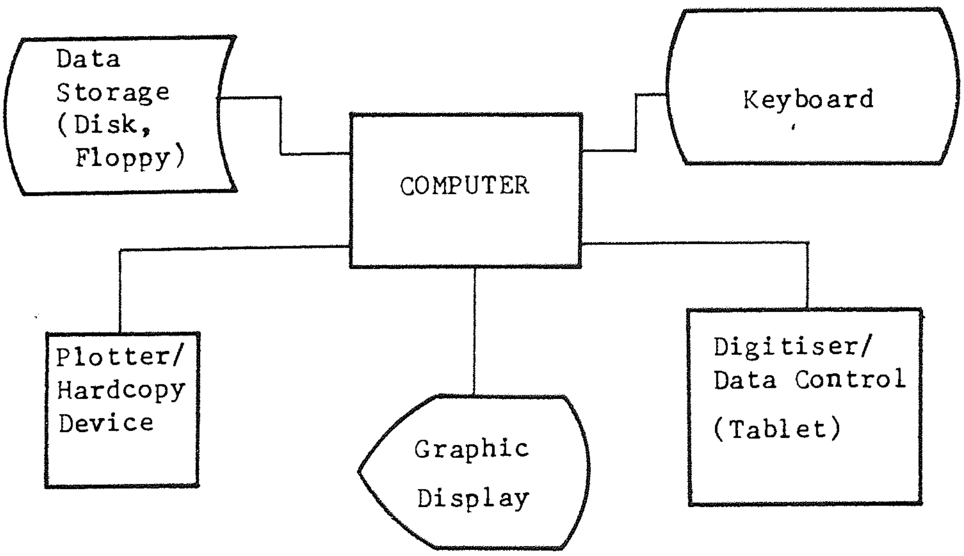


Fig. 3.2 AN INTERACTIVE GRAPHICS SYSTEM

CHAPTER 4

CHAPTER 4

COMPUTERISATION OF FORM-ROLL DESIGN AND MANUFACTURE

4.1 INTRODUCTION

As mentioned in the preceding chapter, the CAD/CAM technology is now having decisive influence in the manufacturing industry. The recognition of the potential of CAD/CAM is worldwide and many unexplored areas of engineering suitable for implementing CAD/CAM are swiftly being opened up to take advantage of the new technology. The design and manufacture of form-rolls, the special tooling used in Cold Roll Forming of thin-walled sections, is but one such area where the CAD/CAM approach can make a positive contribution.

The traditional approach for designing and producing form-rolls, as described in Chapter 2, relies heavily on the knowledge and expertise of the designer in determining the forming sequence. In addition, due to the lack of a well defined scientific principle, the forming sequence has to be established by trial and error. This may entail great expense in time and cost, as detail design drawings are usually required to ensure geometric accuracy.

On the other hand, for the conventional method of machining form-rolls, it is necessary to make by hand full scale metal templates. These templates can then be used as gauges for the turning process. This manual method for turning form-rolls relies greatly upon the skill of the machinist. If the shape of the roll is complex, then the demand on skill will become greater and also very time consuming. Thus NC machining should be applied to enhance the traditional manufacturing method.

Conclusively, the slowness of the manual process, coupled with the inconsistency in design style and human performance, are among the chief obstacles to the process of form-roll design and manufacture. Such inefficiency in work, being the result largely of human inadequacy, can to an appreciable extent be remedied or at least improved through some computerised methods and NC machining.

4.2 APPROACH FOR COMPUTERISATION

Concerning the CAD/CAM technology, there is now a wide range of systems and aids available. Some of these products are being supplied as 'turnkey' CAD/CAM systems, for which all hardware, software, and service can be purchased ready-to-use from a single vendor. Thus these 'turnkey' systems can be installed and operated quickly. However, the standard software of these 'turnkey' CAD/CAM systems usually cannot be modified readily. Likewise, there is little flexibility in the arrangement of the hardware, and these CAD/CAM 'turnkey' systems, above all, are relatively expensive.

Besides, most of the 'turnkey' systems are either too generalised or too specialised, and it is rarely possible to obtain a system which completely satisfies the requirements of a highly specialised job, like form-roll design and manufacture. These drawbacks outweigh the advantages of 'turnkey' CAD/CAM systems for some applications. Conversely, the required system can be established by developing specially designed hardware and software, or by installing standardised or specially designed application software on existing or separately purchased hardware. The application software itself can be purchased or hired if it is commercially available, otherwise it has to be designed and developed to specifications.

A great deal of standard software has been developed and is available for a wide range of engineering disciplines including electrical circuit design, civil engineering, printed-circuit layout, piping design, mechanical design, and so on. Similar to the 'turnkey' systems, those 'off-the-shelf' standard software packages, however, may not be suitable for the purpose or being too expensive to implement for specific applications. With regard to the field of roll design and manufacture, a CAD system can be available from Roll Data of Iowa Inc. (47) in the USA, and in the UK, the Machine Tool Industry Research Association (MTIRA) had developed a CAD/CAM software package (48). Another reported CAD package for this application had been done by Ellen and Yuen (49) in Australia. However, all these software packages were of limited capabilities and did not meet the specific requirements of the sponsored company for this research.

In the absence of suitable software for acquisition, the only suitable solution then was to computerise the roll design and manufacturing process by software design and development, so as to produce a special application software package which could be implemented on a computer system of the company's choice.

4.3 PLANNING FOR THE DEVELOPMENT OF THE CAD/CAM SYSTEM

After investigating the traditional roll design and manufacturing activities, based on the company's requirements, computerisation had been decided to be established in different stages. The whole development programme was planned in two distinct phases of software development which sequentially cover each relevant area of both the CAD and CAM functions. As an interim measure, existing conventional methods continued to operate while the computerised methods were being progressively developed and introduced, so as to avoid disrupting the existing roll design and manufacturing activities of the company.

Phase 1 of the development programme dealt with the establishment of CAD facilities for form-roll design. Due to the work done by Ng [50], this phase of the programme had been completed with the construction of software programs for the automatic draughting of the finished section, the flower patterns, the 10 to 1 template drawings, and the roll design drawings. All these programs were written in FORTRAN IV [51], utilising the GINO-F [52] library routines for processing graphics, and had been implemented as a batch processing system accessible via a remote terminal linked by telephone lines to the ICL 1904s computer at the University's Computer Centre.

Phase 2 of the programme, on the other hand, has been planned to design and develop the CAM facilities for form-roll production, and hence set up an integrated CAD/CAM system for the design and manufacturing of form-rolls. It is the theme of this research to cover the relevant tasks of Phase 2. According to this objective, the following activities were planned to be included in this research:

1. A study on the production of form-rolls by CNC.

2. Implementation of the CAD programs on a stand-alone minicomputer based CAD/CAM workstation.

3. Design and development of a roll editor with the interactive graphics facility.

4. Design and development of a special purpose processor and postprocessor for the generation of NC tapes for machining the form-rolls using a NC lathe.

All the above activities have been successfully carried out according to the plan in this research.

4.4 THE WORKSTATION

Although the ICL 1904s computer in the University's Computer Centre had been used by the sponsored company for Phase 1 of the development programme, it was the long term objective of this research, with regard to Phase 2 of the programme, to establish a stand-alone CAD/CAM system in the company.

The workstation is the most important part of any CAD/CAM system in industrial operation. As it had been decided that all application software programs should be designed and developed to suit the special requirements for form-roll design and manufacture, it was thus not justified to adopt those expensive 'turnkey' CAD/CAM workstations. Nevertheless, the workstation chosen for this task should bear the following salient features:

1. The memory of the system should be large enough to run the programs which are large in size, for instance, the Roll Design Program is of 100K words (24 bit word) in size.

2. Graphics display facilities should be available.

3. As all software programs were written in FORTRAN and using GINO-F library for graphics, the acquired system should therefore provide a FORTRAN compiler as well as the GINO-F software library.

4. It was considered not to be cost justified to install an expensive computer system in the sponsored company which is only a small organisation. Multi-programming or time-sharing were not the primary needs at this initial stage of CAD/CAM implementation for the concerned form-roll design and manufacturing process, thus a single user workstation of reasonable price was deemed to be the most appropriate choice.

4.4.1 The ICL PERQ Workstation

Bearing in mind the considerations mentioned, a PERQ workstation supplied by the International Computers Limited (ICL) was then purchased. The PERQ personal computer system (Plate 4.1) was originally developed by an American company - the Three Rivers Computer Corporation, as an economically viable single-user system. Under the commercial co-operation between Three Rivers and ICL, ICL began UK production of the PERQ in 1982.

PERQ is a high-powered, single-user computer system with a high-precision display. Its main features are the following [53]:

1. High-speed processor: The 16-bit CPU executes approximately 1 million high-level machine instructions (Q-codes) per second.

2. Memory: The main store of the acquired PERQ consists of 1 Mbyte of randomly accessible memory (RAM).

3. High-quality display: A 210 x 275 mm, 768 x 1024-pixel, high resolution black and white raster display (Plate 4.2) featuring a 60 Hz noninterlaced refresh rate enables pictures to be moved cleanly and rapidly.

4. Graphics tablet: The 2-D tablet (Plate 4.3) used in conjunction with a four-button cursor (Plate 4.4) enables a user-friendly man-machine interface.

5. Large virtual memory: The virtual memory system allows large application programs to run without requiring sophisticated overlay technique.

6. Local filestore: A 24 Mbyte Winchester disk and 1 Mbyte floppy disk give a user a large amount of local storage capacity.

7. FORTRAN 77 [54] compiler and GINO-F [55] graphics software library are both provided.

4.4.2 Other Peripherals

The ICL Perq has a General Purpose Instrumentation Bus (GPIB) interface (IEEE 488) as well as a RS232C interface. Thus a wide variety of equipment can be inerfaced with the PERQ. For instance, a HP7221C plotter (Plate 4.5) had been linked to provide hardcopy drawing output while a paper tape punch can be connected to produce the NC control tapes.

4.5 CNC LATHE

After a feasibility study of the production of form-rolls by CNC [56], a CNC lathe has then been installed by the sponsored company, to upgrade the conventional manufacturing process of form-rolls. The acquired lathe is the Japanese made Mori-Seki lathe equipped with a Fanuc CNC controller (Plate 4.6). More details concerning the production of form-rolls by CNC will be described in later chapters.

4.6 THE SOFTWARE SYSTEM

With regard to the entire computerisation plan, covering both of Phase 1 and Phase 2 of the development programme, the following application software programs had been designed and developed:

1. The Finished Section Program.

2. The Flower Pattern Program.

3. The Template Program.

4. The Roll Design Program.

5. The Roll Editor.

6. The Roll Machining Program.

7. The Postprocessor for NC Lathe.

8. The Tape Checking Program.

The first four units of the above listed software had been developed by Ng [50] corresponding to Phase 1 of the development programme. They constituted the roll design process to be computerised. The remaining units of software had been designed and developed in this research, corresponding to Phase 2 of the programme, which mainly concerns computerised roll manufacturing.

4.6.1 Implementation of the Software System in PERQ

As described earlier, the first four units of the software were originally written in FORTRAN IV for batch processing in the ICL 1904s computer. In this research, following the acquisition of the ICL PERQ, all these four program units had been modified firstly to adapt to the PERQ's FORTRAN 77 [54] compiler, and secondly to exploit the interactive and instant computing characteristic of the PERQ, by capturing data through man-machine dialogues. Naturally, the other four program units had been written in FORTRAN 77 and utilising the dialogue technique for data input as well. Besides, routines from PERQ's GRAFIKS [57] software had been used for generating interactive graphics on the screen.

4.6.2 <u>Software Functions</u>

1. THE FINISHED SECTION SOFTWARE

(a) Section drawing.

(b) Automatically generated meanlengths and strip size.

(c) Dimensioning.

(d) Paper size selection and scaling of drawing.

(e) Title block content printing.

2. THE FLOWER PATTERN SOFTWARE

(a) Flower pattern with common origin.

(b) Flower pattern with separate origin.

(c) Automatic bending radii control.

(d) Ability to process multiple element bending at the same stage.

3. THE TEMPLATE SOFTWARE

Template drawings (can be scaled up to 10:1) for all forming stages according to the flower patterns. This software was originally developed for the conventional roll manufacturing process, and thus it has then lost its main purpose following the implementation of NC turning of the rolls. Nevertheless, this software has been maintained and implemented in the PERQ system, mainly as a supporting feature, as templates can still be used for quality control purposes.

4. THE ROLL DESIGN PROGRAM

(a) Automatic generation of top and bottom roll contour based on the template contour.

(b) Automatic incorporation of pinch-difference surfaces based on the supplied clearance values.

(c) Automatic separation of side-roll contour from the top and bottom roll contour when required.

(d) Automatic addition of extension-contours when required.

(e) Generation of roll drawing incorporating the selected optional features for all forming stages.

(f) Generation of roll contour data output.

5. THE ROLL EDITOR

(a) Display the profile of the selected roll on the PERQ screen.

(b) Provide editing functions for modifying the roll profile. Facilities available at present are Deletion, Replacement, or Insertion of an element, and Corner Modification.

(c) Update the edited roll profile data.

(d) Generate a separate roll contour data ouput for the selected roll.

6. THE ROLL MACHINING SOFTWARE

(a) Generate the Cutter Location Data (CLDATA) file for turning a selected roll according to the inputting machining instruction.

(b) Automatic generation of cutter path for the chosen cutting cycle. At present, four types of lathe cutting cycles have been equipped, namely, roughing cycle, grooving cycle, pocketing cycle and fine turning cycle.

(c) Automatic display of the cutter path on the PERQ's screen.

7. THE POSTPROCESSOR

(a) Generate the NC part program from the CLDATA file of the selected roll for the Mori-Fanuc CNC lathe system.

(b) Automatic display of the cutter path on the PERQ's VDU by reading data from the NC part program.

8. THE NC TAPE CHECKING PROGRAM

Display the cutter path on the PERQ's screen while reading an NC part program for the Mori-Fanuc CNC turning machine. This can then act as a visual check of the NC tape before actually loading on the CNC lathe.

4.7 THE COMPLETE CAD/CAM SYSTEM SOFTWARE LAYOUT

While implementing the CAD/CAM software programs in the PERQ computer, a sequential approach had been chosen for operating the different program units. In other words, the program units should be executed in sequence. As a start, the Finished Section Program has to be run in the first place so as to produce a data file for the profile of the finished section, which should then be followed by operating the Flower Pattern Program. As templates are not required in NC turning of rolls, the Template Program is then an optional operation which will not influence further processing of the complete system. The Roll Design Program, however, can then be operated in response to the successful execution of the Finished Section and Flower Pattern Programs.

The Roll Design Program, as mentioned earlier, will generate a single roll contour data file for the whole set of rolls for all forming stages. After that, the Roll Editor has to be operated, to modify the roll profile if required, as well as to generate an updated roll contour data file for a selected roll. Lastly, the Roll Machining Program and the Post Processor can then be executed in succession to generate an NC control part program for turning the selected roll on the CNC lathe. On the other hand, the NC Tape Checking Program, independent of the integrated CAD/CAM software system, can be used to pre-check any NC part program before actually loading in the Mori-Fanuc CNC system.

The integrated CAD/CAM software system is depicted in Fig. 4.1.

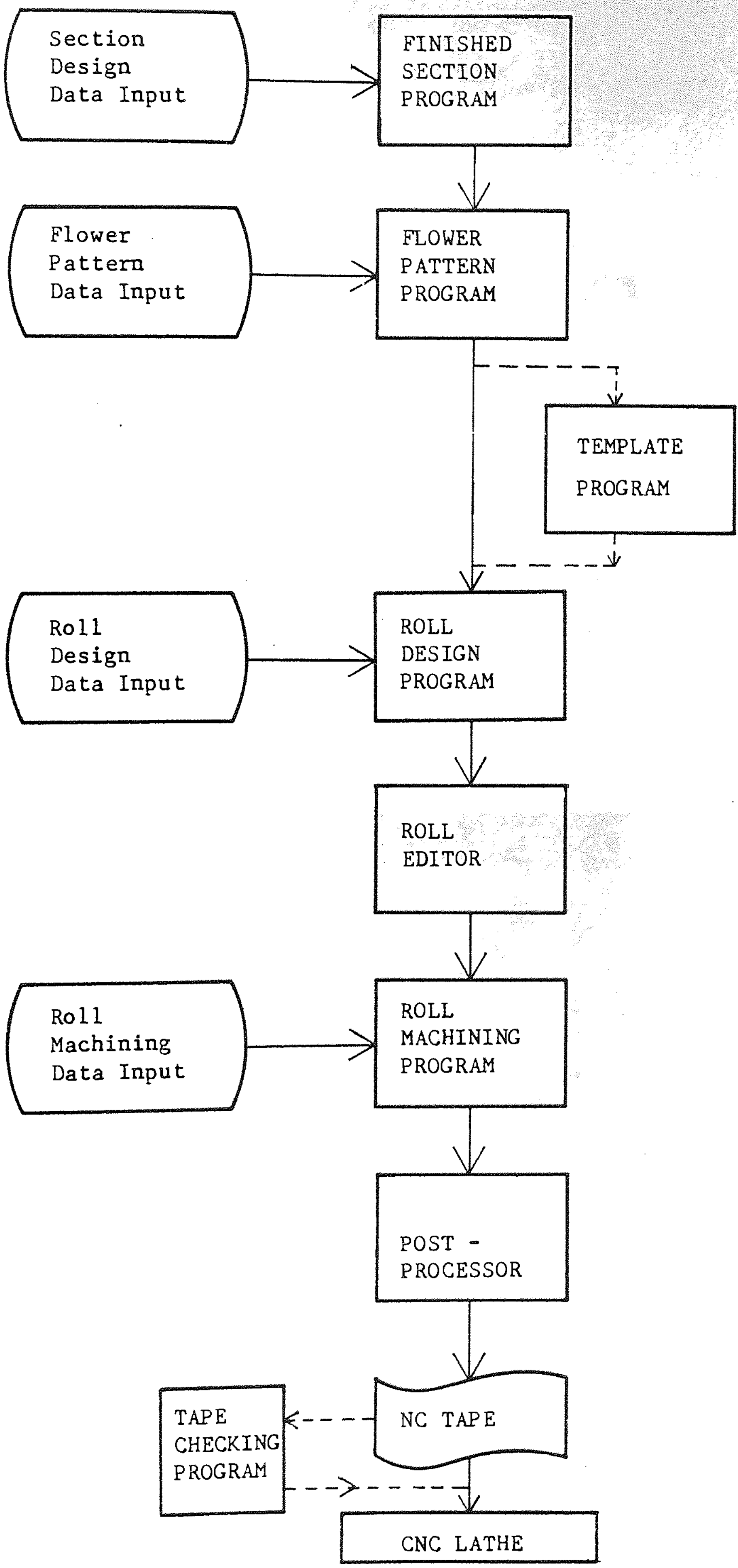


Fig. 4.1 THE INTEGRATED CAD/CAM SYSTEM FOR FORM-ROLLS

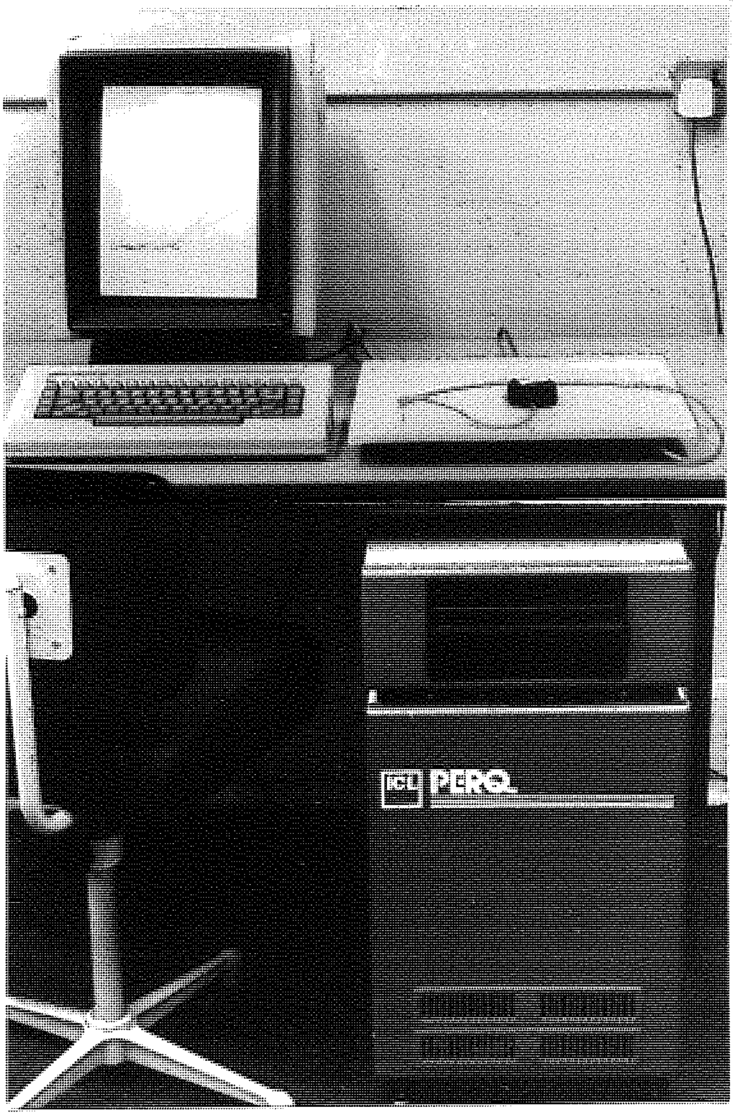


Plate 4.1 THE ICL PERQ COMPUTER

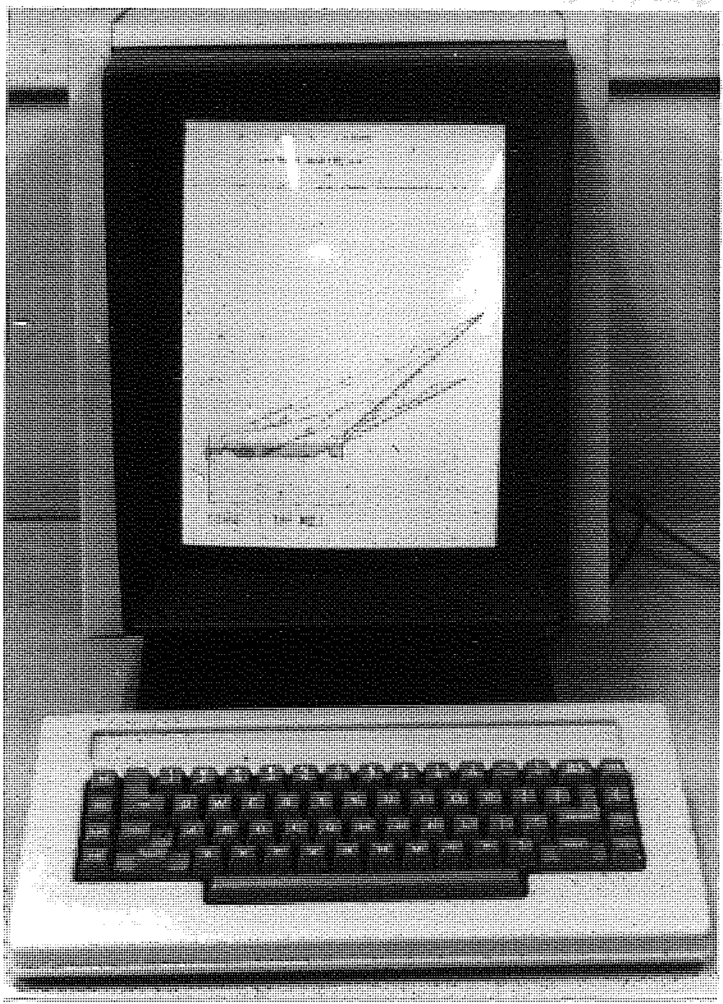


Plate 4.2 THE PERQ RASTER DISPLAY

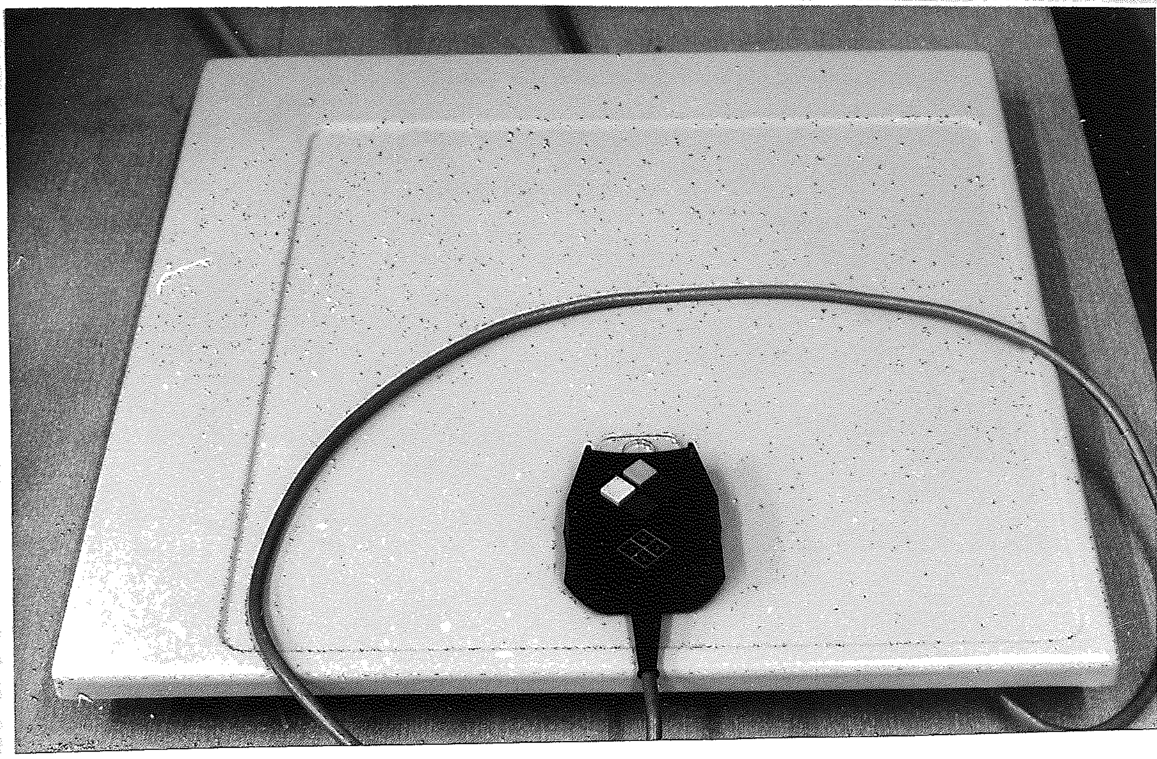


Plate 4.3 THE PERQ 2-D TABLET

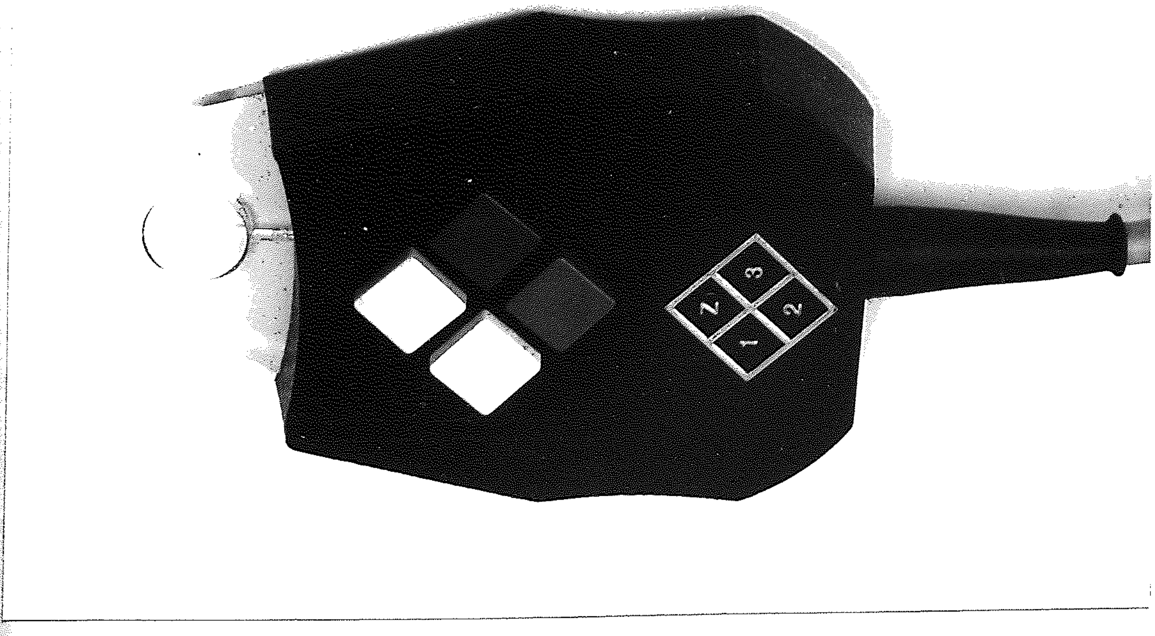


Plate 4.4 THE FOUR-BUTTON CURSOR

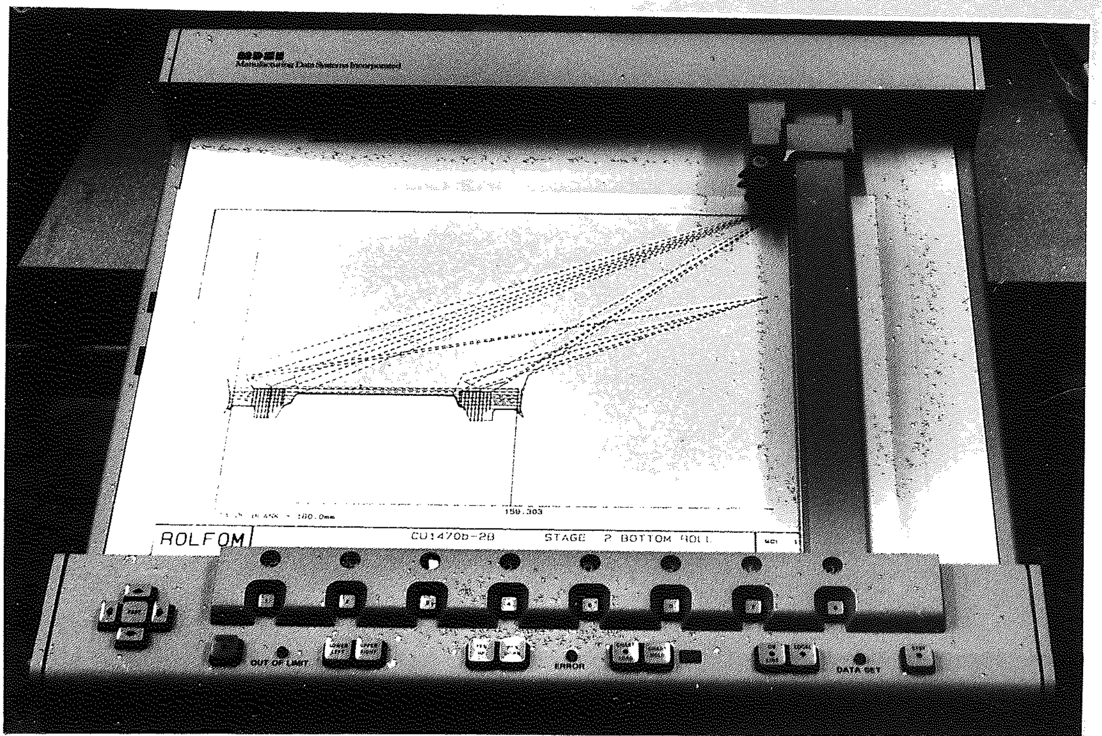


Plate 4.5 THE HP7221C PLOTTER

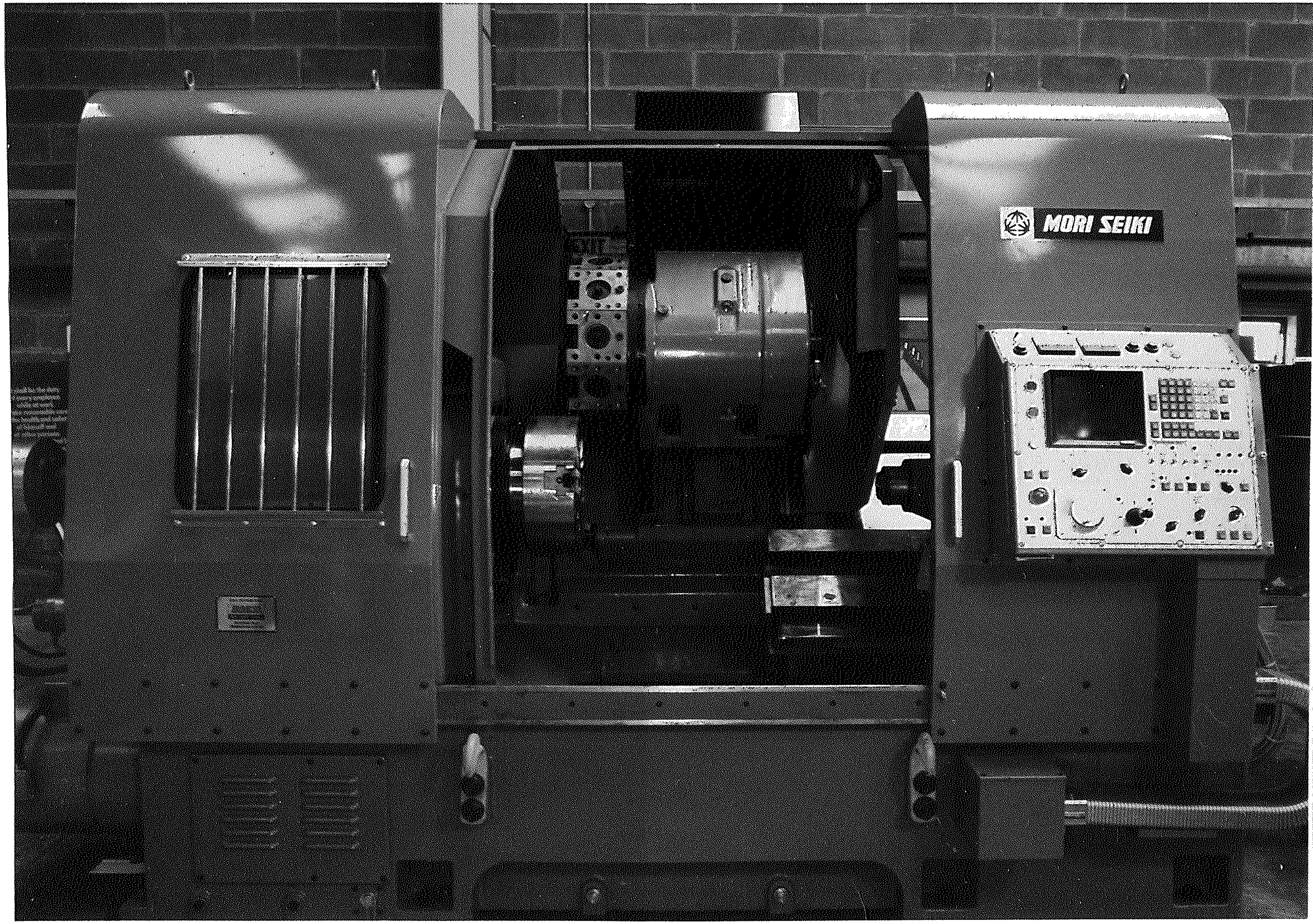


Plate 4.6 THE FANUC CONTROLLED MORI-SEKI CNC LATHE

CHAPTER 5

CHAPTER 5

COMPUTER SOFTWARE FOR FORM-ROLL DESIGN

5.1 INTRODUCTION

The software developed during Phase 1 of the entire computerisation programme [50], ranging from the finished section program to the roll design program, will be briefly described in this chapter. This suite of programs, written in FORTRAN, were originally established for batch processing on the ICL 1940s computer. Therefore, all data input for the programs had to be pre-entered in separate data files. By refering to the specified input data files, necessary data input could then be captured through the appropriate input/output channels.

In this research, however, all these programs had been modified for implementing on the PERQ workstation. Data acquisition could then be achieved, while running the program, directly through a question/answer sequence via the terminal screen and the keyboard. At the same time, the input information would be dumped in assigned files, which could therefore serve as a permanent record. Of course, these files could be recalled when re-running the programs, so that keyboard entry effort could be minimised. Besides, by using the system editor of the PERQ computer, these data files could be manipulated readily by the designer in the case of design change or modification.

In operation, the software is structured to lead the designer through the conventional stages of the design by the series of clear requests and prompts which appear on the terminal screen. As expected, the finished section program should be operated first, which should then be followed by the generation of the flower patterns by the flower pattern program, and thereafter, the roll design program could only be executed to produce the form-roll profiles.

A detailed description of the above mentioned programs has already been given by Ng [50], the following is, nevertheless, aimed simply at drawing attention to the general outline and some significant aspects of these programs, with special emphasis on their implementation on the PERQ workstation.

5.2 THE FINISHED SECTION SOFTWARE

The finished section is the final product of the cold-roll forming process, therefore its geometry must be accurately and adequately defined. It is the inherent aspects of the finished section program to enable the designer to distinctly define the section geometry, as well as to produce the finished section drawing automatically, either on the PERQ screen or to obtain a hardcopy by a plotter.

5.2.1 Section Definition

In general, a section geometry consists of only linear and circular elements with uniform thickness (Fig. 5.1). Non-linear type of elements other than circular elements are normally not used but if used they may be approximated with circular elements. Thus a section definition scheme that handles linear and circular elements is adequate. In essence, as the thickness is uniform, and should be defined beforehand, the only information needed for each element is its length if the element is linear (Fig. 5.2), or the inside radius if the element is circular (Fig. 5.3). The convention adopted for the directions of bending of circular elements is simple to use, being positive for upward bending and negative for downward bending, irrespective of which side of the section the bend is required (Fig. 5.4).

A terminal listing for data input for the finished section program is given in Fig. 5.5.

5.2.2 Automatic Strip Size Calculation

The finished section program also automatically calculates the individual meanlength of each element as well as the required strip size (or material width) for forming the desired section. Various methods for computation in this respect have already been mentioned in Section 2.5.1. In this program, equation 2.7 has been adopted in conjunction with Table 2.1 for calculating the mean radius of a circular element, while the strip size, straightforwardly, will be determined in equation 2.1.

5.2.3 Operating the Program

In this program, facilities for selecting the plotting frame size ranging from AO to A4 and for controlling the scale of the drawing are also incorporated. An optional dimensioning facility which enables the designer to keep track of the details of each element is also available. However, the size of the plotter drawings would be limited by the size of the plotter as a matter of fact, and the size of the printed drawing on the VDU is inevitably limited by the size of the PERQ screen.

The display of the finished section drawing on the PERQ screen is shown in Plate 5.1.

In addition to the drawing, an intermediate data file is generated to retain data which will be needed in later stages of processing. In fact, this intermediate file will be retrieved during the operation of the flower pattern program, as well as the roll design program, for obtaining the geometric data of the finished section.

5.3 THE FLOWER PATTERN SOFTWARE

The flower pattern program had been developed for generating accurate flower patterns to be used in the forming sequence to produce required section profiles. Basically, the designers are required to supply only information or data regarding the extent of bending of selected elements of the section in each stage of forming. In other words, the designers have to provide new bending information concerning only the affected elements, bending status of all other elements in the preceding stage will automatically be assumed in the current stage. A typical terminal listing for data input for this program is given in Fig. 5.6. The resulting flower pattern drawing will be displayed on the PERQ screen (Plate 5.2), hence, the distribution of bending of each element in relation to other elements can be easily inspected to locate areas of awkward bending and material interference that should be corrected. Radii of bending at successive stages are automatically generated using the opening-radii method.

5.3.1 The Opening-Radii Method of Forming

This is a computerised method which automatically generates suitable radius of bending for each circular element at successive stages. Provided that the relative positions of the elements adjacent to the particular circular element being bent can be precisely monitored at each bending stage, it is theoretically feasible to maintain a constant cross-sectional area and a uniform thickness for the circular element throughout forming, as illustrated in Fig. 5.7, such that:-

$$r_i \theta_i = r_j \theta_j = \dots = k \qquad (5.1)$$

where r_i is the inside radius of bend at stage i

θ_i is the angle of bend (in radians) at stage i

r_j is the inside radius of bend at stage j

θ_j is the angle of bend (in radians) at stage j

k is constant

As this method is based upon the principle of constant cross-sectional area of bent elements throughout forming, the tendency of material thinning during bending is greatly reduced.

5.4 THE TEMPLATE SOFTWARE

Before the installation of a NC lathe, wire templates were required as an aid in manual turning of the rolls. Thus a 10 to 1 template program had been developed for this purpose of generating template drawings, ten times full size, which are to be used in manufacturing the wire templates. However, a large size plotter should be used for plotting the 10 to 1 template drawings. When the program was initially run on the ICL 1904s computer, the drawings were produced on graph-plotting paper of width 860 mm and larger drawings had their scale automatically reduced to fit the paper width.

Nevertheless, wire templates are no longer required with the present NC roll turning method. The template program has thus been implemented on the PERQ system, merely as a supporting function, which will produce the template drawing for every forming stage, and the size of a template drawing will automatically be scaled down to fit on the PERQ screen. The template drawing can then be used by the designer as a reference for checking the shape of the product during each forming stage. Plate 5.3 shows a template drawing on the PERQ screen.

5.5 THE ROLL DESIGN SOFTWARE

The roll design program had been developed to automatically generate the roll drawings and profile data to design specification. Most of the tedious and routine roll design functions are now handled by computer software, leaving the designers to perform the more creative and subjective decision making work. Four major areas of the entire roll design function had been computerised, namely:-

1. Basic Roll Contour Design

2. The Pinch-Difference Option

3. The Side-Roll Option

4. The Extension-Contour Option

5.5.1 Basic Roll Contour Program

This involves automatically designing the top and bottom roll contours using the conventionally generated template profiles as the design basis. Parts of the template profiles not accessible by the roll material during forming are automatically modified to prevent interference between the roll and the strip material. Just like the template contours, a roll contour also consists of linear and circular elements. During contour modification, part or whole of the linear or circular element contour may have to be removed and replaced by a new contour. The new contour is always linear and vertical to avoid roll interference with the material being formed.

5.5.2 The Pinch-Difference Options

Pinch-difference is a term used to specify the amount of control required at the gap size of each bent part of the section. At certain parts of the roll periphery, firm contact between the roll surface and strip surface is necessary for drawing the strip forward while at other parts a clearance between the surfaces is more appropriate to avoid undesirable friction during forming. Facilities for designing the pinch-difference surfaces of the required type had therefore been included in the software.

5.5.3 The Side-Roll Options

Although most of the forming operations in practice can be handled with the use of top and bottom rolls alone, it is frequently desirable to use side-rolls to perform part of the forming, especially when bending legs to a near vertical orientation. The side-roll option enables designers to precisely define which parts of the section are to be formed by side-rolls instead of top and bottom rolls. Side-roll contours are then automatically generated with a side accessibility of roll material instead of the normal top and bottom accessibility. The definition scheme adopted is simple to use, for each side-roll contour required, it is necessary only to specify the starting elemental contour and the ending element contour involved.

5.5.4 The Extension-Contour Option

Another important feature in roll design is the use of special contours which interlocks one roll with the other to prevent axial misfits during forming. Such contours, sometimes referred to as gates, are termed extension-contours since they are contours extended from the break-points that separate one roll contour from the other. Gates frequently used in top and bottom pairs are commonly called spigots. A difinition scheme had been incorporated in the software to enable designers to specify extension-contours of the commonly used shapes and sizes to be used in the roll designs. Extension-contour definitions for roll pairs either with or without side-rolls can be easily specified.

5.5.5 Operating the Program

Again, while implementing on the PERQ computer, this program has been converted to read in data through a question/answer sequence. Hence, the roll designer, being guided by the requests and prompts on the terminal screen, can input the roll design data and choose the appropriate roll design options directly through the keyboard. A terminal listing for this program is given in Fig. 5.8. As a result of running this roll design program, roll design drawings will be produced on the PERQ screen. That is, the roll drawings will be generated in a stage by stage sequence, for all forming stages. In doing so, the roll profiles in each forming stage will automatically be scaled down to fit in the PERQ screen. Plate 5.4 is the roll drawing for one of the forming stage.

Besides, the successful running of the roll design program will produce, in conjunction with the roll drawings, a digitised roll profile data file. This file, which stores the geometric data for all the rolls in all forming stages, is required in further processing of the CAD/CAM process for providing geometric description of roll profiles.

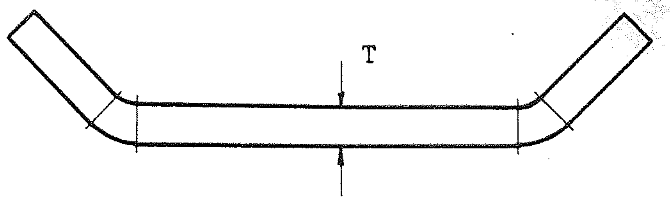


Fig. 5.1 ELEMENTS OF THE SECTION GEOMETRY

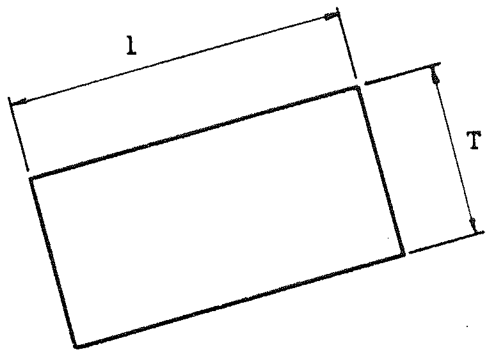


Fig. 5.2 DEFINITION OF A LINEAR ELEMENT

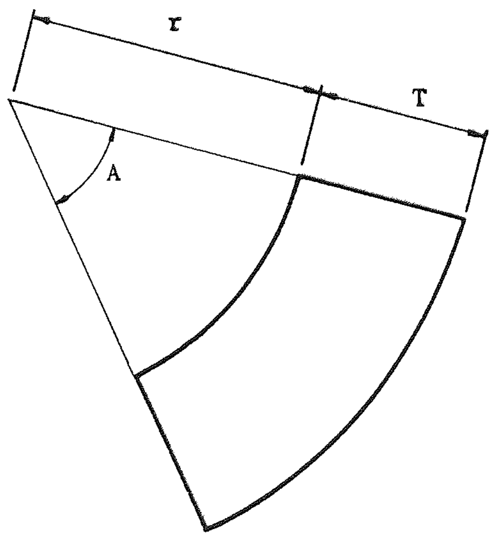


Fig. 5.3 DEFINITION OF A CIRCULAR ELEMENT

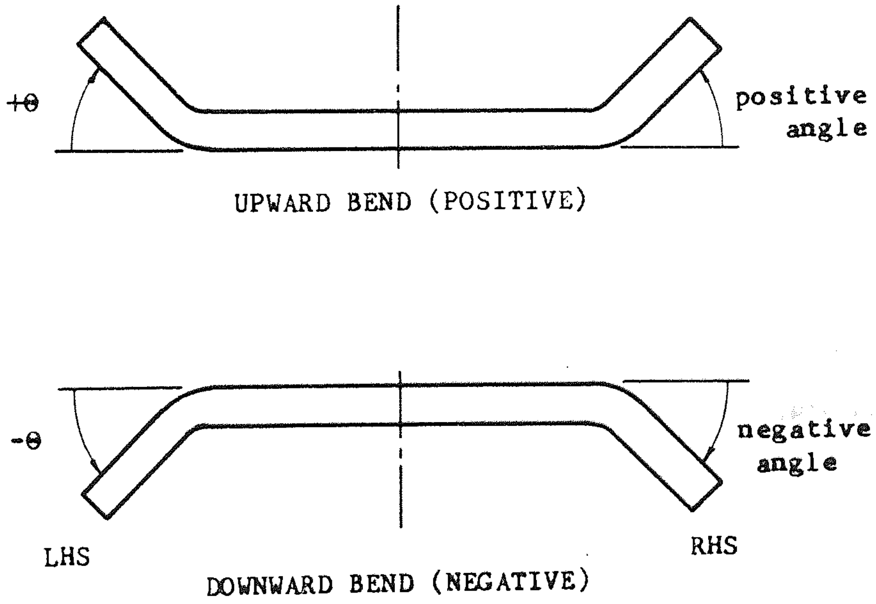


Fig. 5.4 CONVENTION FOR THE BENDING DIRECTION

```
*** ROLFOM ***
******

** FINISHED SECTION PROGRAM **


*** PLEASE INPUT THE SECTION NO ***
section no. ?
  1470b

        * PLOTTING PROGRAM NO. 1 FOR ROLLED-SECTION *

                  FINISHED SECTION
                  ----------------


PLEASE SUPPLY THE FOLLOWING DATA ACCORDING TO THE GIVEN FORMAT:-

INPUT UNIT        OUTPUT UNIT        THICKNESS        ORIGIN
----------        -----------        ---------        ------

(INTEGER)         (INTEGER)          (REAL)           (INTEGER)


WHERE,

INPUT UNIT IS IN INCH(1) OR MM(2),
OUTPUT UNIT IS ALSO IN INCH(1) OR MM(2),
THICKNESS IS THICKNESS OF STRIP,
ORIGIN IS THE STARTING POINT FOR ELEMENT DEFINITION SEQUENCE LATER ON.

ORIGIN=1 IF FROM LEFT TOWARDS RIGHT (ONE-SIDED).
ORIGIN=2 IF FROM RIGHT TOWARDS LEFT (ONE-SIDED).
ORIGIN=3 IF FROM CENTRE TOWARDS RIGHT AND THEN FROM CENTRE TOWARDS LEFT.
ORIGIN=4 IF FROM CENTRE TOWARDS LEFT AND THEN FROM CENTRE TOWARDS RIGHT.
ORIGIN=5 IF FROM CENTRE TOWARDS RIGHT (SYMMETRICAL SECTION).


** PLEASE INPUT IUNIT,OUNIT,THICK,ORIGIN **
IUNIT OUNIT    THICK   ORIGIN
  2     2       0.5000    5
```

Fig. 5.5 A TYPICAL TERMINAL LISTING FOR DATA INPUT FOR THE FINISHED SECTION PROGRAM (PART 1)

PLEASE ENTER THE DATA FOR ELEMENT DEFINITION SEQUENCE ACCORDING TO THE GIVEN FORMAT, ONE DEFINITION STATEMENT FOR EVERY ELEMENT ENTERED :-

SEQUENCE NUMBER (N)	ELEMENT TYPE (TYPE)	LENGTH OR RADIUS	ANGLE OF BENDING
(INTEGER)	(INTEGER)	(REAL)	(REAL)
START FROM 1, INCREMENT BY 1 UP TO 50, TO TERMINATE ENTER 0.	1 = LINEAR, 2 = CIRCULAR, THE REST ARE ILLEGAL.	POSITIVE LENGTH FOR TYPE 1, INSIDE RADIUS FOR TYPE 2 (MAY BE 0).	FOR TYPE 2 ELEMENTS ONLY, ZERO FOR TYPE 1, NON-ZERO FOR TYPE 2. *

*THE EXCEPTION BEING, WHEN N=1 AND TYPE=1 (AND ORIGIN NOT 5!), ANGLE OF BENDING IS TAKEN AS THE ANGLE OF INCLINATION BETWEEN THE HORIZONTAL-AXIS AND THE FIRST LINEAR ELEMENT, PERMISSIBLE RANGE IS FROM -90 TO +90 DEGREES.

NOTE THAT WHEN ORIGIN IS 3 OR 4 (DOUBLE SIDED DEFINITION), 2 SETS OF DEFINITION SEQUENCE ARE REQUIRED, ONLY 1 SET IS REQUIRED WHEN ORIGIN IS 1,2 OR 5.
FIRST ELEMENT AND LAST ELEMENT OF THE SEQUENCE MUST BE LINEAR.
MAXIMUM NO. OF ELEMENTS IN EACH SEQUENCE MUST BE LESS THAN 50.

1	1	40.6499	0.0000
2	2	1.5000	-23.0000
3	1	0.0010	0.0000
4	2	1.5000	-30.0000
5	1	0.0010	0.0000
6	2	1.5000	-20.0000
7	1	0.0010	0.0000
8	2	1.5000	-25.0000
9	1	0.0010	0.0000
10	2	1.5000	-45.0000
11	1	1.9499	0.0000
12	2	2.5000	38.0000
13	1	0.0010	0.0000
14	2	2.5000	25.0000
15	1	0.0010	0.0000
16	2	2.5000	25.0000
17	1	0.0010	0.0000
18	2	2.5000	25.0000
19	1	1.7500	0.0000
20	2	0.0600	-180.0000
21	1	1.3000	0.0000
0	0	0.0000	0.0000

Fig. 5.5 (PART 2)

```
PLEASE SELECT THE PAPERSIZE AND THE SCALE
ACCORDING TO THE GIVEN FORMAT :-


          PAPERSIZE          SCALE
          ---------          -----

          (INTEGER)          (REAL)

PAPERSIZE = 0,1,2,3,4 FOR A0,A1,A2,A3,A4 SIZES RESPECTIVELY.
SCALE= ANY POSITIVE VALUE LESS OR EQUAL TO THE FOLLOWING
       LIMITS BASED ON THE SELECTED PAPERSIZE.


FOR PAPERSIZE A0, MAXIMUM PERMISSIBLE SCALE = 12.31
FOR PAPERSIZE A1, MAXIMUM PERMISSIBLE SCALE =  8.09
FOR PAPERSIZE A2, MAXIMUM PERMISSIBLE SCALE =  5.28
FOR PAPERSIZE A3, MAXIMUM PERMISSIBLE SCALE =  3.16
FOR PAPERSIZE A4, MAXIMUM PERMISSIBLE SCALE =  1.75
  4    1.0000


DO YOU WANT ANY DIMENSIONING?

ENTER 1 IF YES, OR 0 IF NO.
 1


DO YOU WISH TO SUPPLY ANY TITLE-BLOCK INFORMATION ?

ENTER 1 IF YES, OR 0 IF NO.
 0
```

Fig. 5.5 (PART 3)

```
                    **** ROLFOM ****
                         ******

        * PLOTTING PROGRAM NO. 2 FOR ROLLED-SECTION *

                  --- FLOWER PATTERNS ---

** PLEASE INPUT THE SECTION NO. **
section no. ?
1470b
** INPUT JRUN = 1 IF FLOWER PATTERN ONLY,OR
                2 IF ROLLER-PLOTTINGS ONLY,OR
                3 IF BOTH OF THE ABOVE
JRUN = ?
 3
** INPUT NSTAGE (NSTAGE = TOTAL NO. OF PASS,  MAXIMUM 50)
NSTAGE = ?
 3
** INPUT JBEND (SELECTION OF CIRCULAR-ELEMENT  BENDING OPTION)
(TYPE 0 IF SIMPLE ELEMENT DEFINITION, OR
      1 IF COMPOSITE PERCENTAGE ELEMENT DEFINITION)
JBEND = ?
 0
** THE SECTION IS A SYMMETRICAL SECTION **
-----THUS INPUT THE FOLLOWING :
PASS     ELEMENT TO         ANGLE OF
 NO.     BE BENT(RIGHT)     BEND
(terminate input by typing
 0        0                 0.0)
 1       10                  -45.00
 1       18                   25.00
 2        8                  -25.00
 2       16                   25.00
 2       20                  -50.00
 3        6                  -20.00
 3       14                   25.00
 3       20                  -75.00
 0        0                    0.00
** DO YOU WANT THE OPTION FOR SHARPENING OF INSIDE RADII ?
-- input JSHARP = 0 if radii-sharpening is not required at all, or
                  1 if stage no. are to be entered individually, or
                 -1 if all stages except last stage require sharpening

JSHARP =?
 0
```

Fig. 5.6 A TYPICAL TERMINAL LISTING FOR DATA INPUT FOR THE FLOWER PATTERN PROGRAM

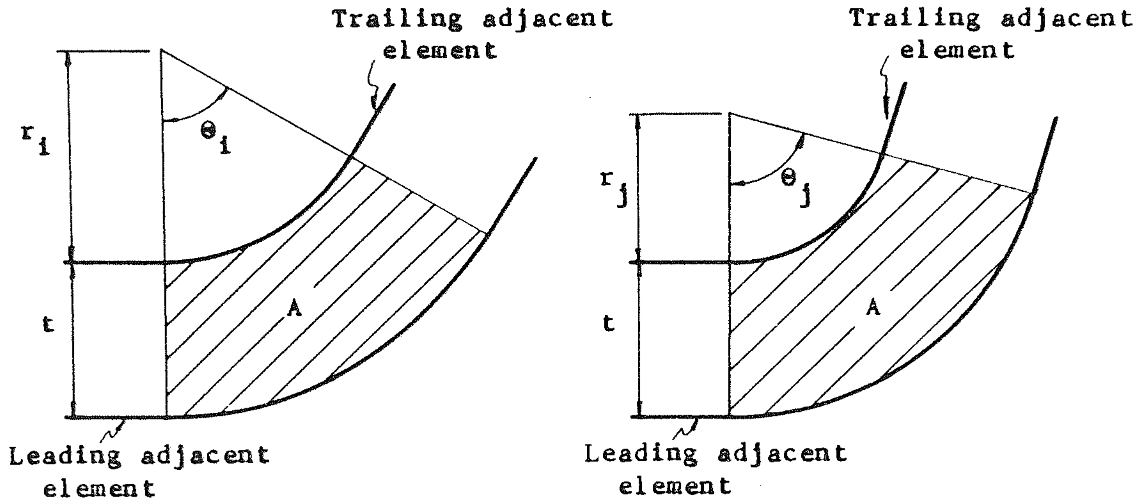


Fig. 5.7 THE OPENING-RADII METHOD OF FORMING

```
                         *** ROLFOM ***

                     ** ROLL DESIGN PROGRAM **

** INPUT THE SECTION NO., PLEASE! **
section no. ?
1470b
**DO YOU WANT CONSISTENT PASS-HEIGHTS & C-C DISTANCE?
**INPUT IPASHT =  1 IF YES,OR
                 -1 IF YOU WANT INCONSISTENT PASS HTS AND C-C DISTANCE
IPASHT = ?
-1
**PLEASE INPUT THE PASS HEIGHT (PASHT) AND C-C DISTANCE (CTOC)
INPUT PASHT & CTOC FOR STAGE  1 PLEASE
  71.755      128.143
INPUT PASHT & CTOC FOR STAGE  2 PLEASE
  74.295      130.683
INPUT PASHT & CTOC FOR STAGE  3 PLEASE
  76.200      132.296
**INPUT 1 FOR ITOC IF YOU WANT CONSISTENT LH & RH TOLERANCE,OTHERWISE
**INPUT -1 FOR INCONSISTENT TOLERANCES
ITOC = ?
 1
**NOW INPUT THE TOLERANCES, WHERE
  TOLLH = LH TOLERANCE BETWEEN ROLLER & LAST ELEMENT,
  TOLRH = RH TOLERANCE BETWEEN ROLLER & LAST ELEMENT.
   TOLLH         TOLRH
   0.149         0.150
**INPUT JTEMP = 1 IF YOU WANT COMPONENT DRAWING IN HIDDEN LINE,OTHERWISE 0
      RSCALE = DESIRED SCALE FOR THE ROLLER DRAWINGS
      RGAP = THE GAP DIMENSION BETWEEN TOP AND BOTTOM ROLL
JTEMP  RSCALE   RGAP
1      1.000    10.000
**INPUT JSTAGE(1) =-1 IF ALL STAGES REQ'D FOR ROLLERS,OR
      JSTAGE(1) = 1 IF NO STAGES REQ'D ,OR
      JSTAGE(1) = ANY POSITIVE INTEGER LESS THAN NSTAGE
                  IF SELECTED STAGES REQ'D FOR ROLLERS
JSTAGE(1)=?
-1
**DO YOU WANT THE PINCH DEIFFERENCE OPTION?
**INPUT JPINCH = 0 IF NO EXTERNAL PINCH DIFFERENCE DIMENSION
                   DEFINITION IS SUPPLIED, OR
                 1 IF ONLY ONE PINCH DIFFERENCE DIMENSION DEFINITION
                   IS SUPPLIED FOR ALL BENDING STAGES, OR
                -1 IF PINCH DIFFERENCE DIMENSION DEFINITION IS SUPPLIED
                   INDIVIDUALLY FOR EACH BENDING STAGE, OR
                 2 IF SINGLE THICKNESS OPTION WITH ONE COMMON CLEARANCE
                   VALUE FOR ALL STAGES SELECTED, OR
                -2 IF SINGLE THICKNESS OPTION WITH INDIVIDUAL CLEARANCE
                   VALUE FOR EACH STAGE SELECTED.
JPINCH = ?
-1
```

Fig. 5.8 A TYPICAL TERMINAL LISTING FOR DATA INPUT FOR THE ROLL DESIGN PROGRAM (PART 1)

```
**INPUT THE PINCH DEFFERENCE DIMENSIONS,WHERE
  PDIM1 = THICKNESS BETWEEN THE DRIVE SURFACES
  PDIM2 = THICKNESS BETWEEN THE CLEARANCE SURFACES
INPUT PDIM1 & PDIM2 FOR STAGE  1 PLEASE
     0.500     0.560
INPUT PDIM1 & PDIM2 FOR STAGE  2 PLEASE
     0.500     0.610
INPUT PDIM1 & PDIM2 FOR STAGE  3 PLEASE
     0.530     0.560
**NOW INPUT ELEMENT NO. WHICH DEFINES DRIVE SURFACE FOR EVERY STAGE,
  2 DEFINITION STATEMENTS PER STAGE REAUIRED,WHERE
  ISQP=CURRENT BENDING STAGE SEQUENCE NO.
  IEPL = L.H.S. ELEMENT DEFINING THE DRIVE SURFACE CONTOUR
  IEPR = R.H.S. ELEMENT DEFINING THE DRIVE SURFACE CONTOUR
  (TERMINATE INPUT BY TYPING  0  0  0)
ISQP  IEPL  IEPR
   1     1     1
   1     9     9
   2     1     1
   2     7     7
   3     1     1
   3     5     5
   0     0     0
**ARE YOU GOING TO CHOOSE THE SIDE ROLL OPTION?
  ENTER ISROL = 1 IF YES,OR
              = 0 IF NO
ISROL = ?
 1
**ENTER IOTOS = 1 IF CONSISTENT ORIGIN TO SIDE-ROLL AXES,OR
               -1 IF INCONSISTENT ORIGIN TO SIDE-ROLL AXES DISTANCES
IOTOS = ?
-1
**INPUT ORIGIN TO SIDE-ROLL AXES DISTANCE,WHERE
  OTOSL = DISTANCE BETWEEN ORIGIN AND LEFT SIDE-ROLL CENTRE, AND
  OTOSR = DISTANCE BETWEEN ORIGIN AND RIGHT SIDE-ROLL CENTRE.
INPUT OTOSL & OTOSR FOR STAGE  1 PLEASE
   100.000   100.000
INPUT OTOSL & OTOSR FOR STAGE  2 PLEASE
   100.000   100.000
INPUT OTOSL & OTOSR FOR STAGE  3 PLEASE
   115.000   115.000
**INPUT THE SIDE-ROLL CONTOUR DEFINITION, WHERE
  ISQS  = CURRENT BENDING STAGE SEQUENCE NO.,
  IESL1 = L.H.S. ELEMENT CONSTITUTING L.H.S. SIDE-ROLL CONTOUR,
  IESL2 = WHICH FACE OF THE L.H.S. ELEMENT IS DESIRED,
  IESR1 = R.H.S. ELEMENT CONSTITUTING R.H.S. SIDE-ROLL CONTOUR,
  IESR2 = WHICH FACE OF THE R.H.S. ELEMENT IS DESIRED.
INPUT ISQS,IESL1,IESL2,IESR1 & IESR2 PLEASE
(TERMINATE INPUT BY TYPING 0  0  0  0  0 )
   1   0   0   0   0
   1   0   0   0   0
   2   0   0   0   0
   2   0   0   0   0
   3   6   2   6   2
   3  21   2  21   2
   0   0   0   0   0
```

Fig. 5.8 (PART 2)

```
**INPUT SELECTION FOR EXTENSION-CONTOUR OPTION,ENTER
  IEXTN = 1 IF EXTENSION CONTOUR OPTION WILL BE USED
        = 0 IF EXTENSION CONTOUR OPTION WILL NOT BE USED
IEXTN = ?
 1
**INPUT TYPE AND DIMENSION FOR EACH DESIGNATED SIDE-CONTOUR EXTENSION,
  WHERE
  ISC1 = SIDE-CONTOUR NO.
  ISC2 = SIDE-CONTOUR TYPE
  SCDIM1 = DIMENSION DEFINITION FOR SIDE-CONTOUR PART 1
  SCDIM2 = DIMENSION DEFINITION FOR SIDE-CONTOUR PART 2
  SCDIM3 = DIMENSION DEFINITION FOR SIDE-CONTOUR PART 3
  SCDIM4 = DIMENSION DEFINITION FOR SIDE-CONTOUR PART 4
(TERMINATE INPUT BY TYPING 0  0  0.0  0.0  0.0  0.0 ),NOW INPUT
ISC1  ISC2   SCDIM1  SCDIM2  SCDIM3  SCDIM4
  1    2     8.000   0.000   8.000  13.000
  2    2     8.000  13.000   8.000  13.000
  3    1     0.000   0.000  12.000   9.000
  4    1     0.000   6.000  13.500   7.500
  0    0     0.000   0.000   0.000   0.000
**INPUT SIDE-CONTOUR SELECTION FOR EACH BENDING STAGEWHERE
  ISQD  = BENDING STAGE NO. FOR WHICH SIDE-CONTOUR DEFINITION IS
          INTENDED
  IELD1 = TYPE OF SIDE-CONTOUR ON L.H.S.
  IELD2 = SIDE-CONTOUR NO. WHICH HAS BEEN DEFINED PREVIOUSLY AND
          IS TO BE SELECTED FOR THIS STAGE NO. ON L.H.S.
  IERD1 = TYPE OF SIDE-CONTOUR ON R.H.S.
  IERD2 = SIDE-CONTOUR NO. WHICH HAS BEEN DEFINED PREVIOUSLY AND
          IS TO BE SELECTED FOR THIS STAGE NO. ON R.H.S.
(TERMINATE INPUT BY TYPING 0  0  0  0  0 ),NOW ENTER
ISQD  IELD1  IELD2  IERD1  IERD2
   1     2      1      2      1
   2     2      2      2      2
   3     1      3      1      3
   3     1      4      1      4
   0     0      0      0      0


                    ** END OF INPUT **
```

Fig. 5.8 (PART 3)

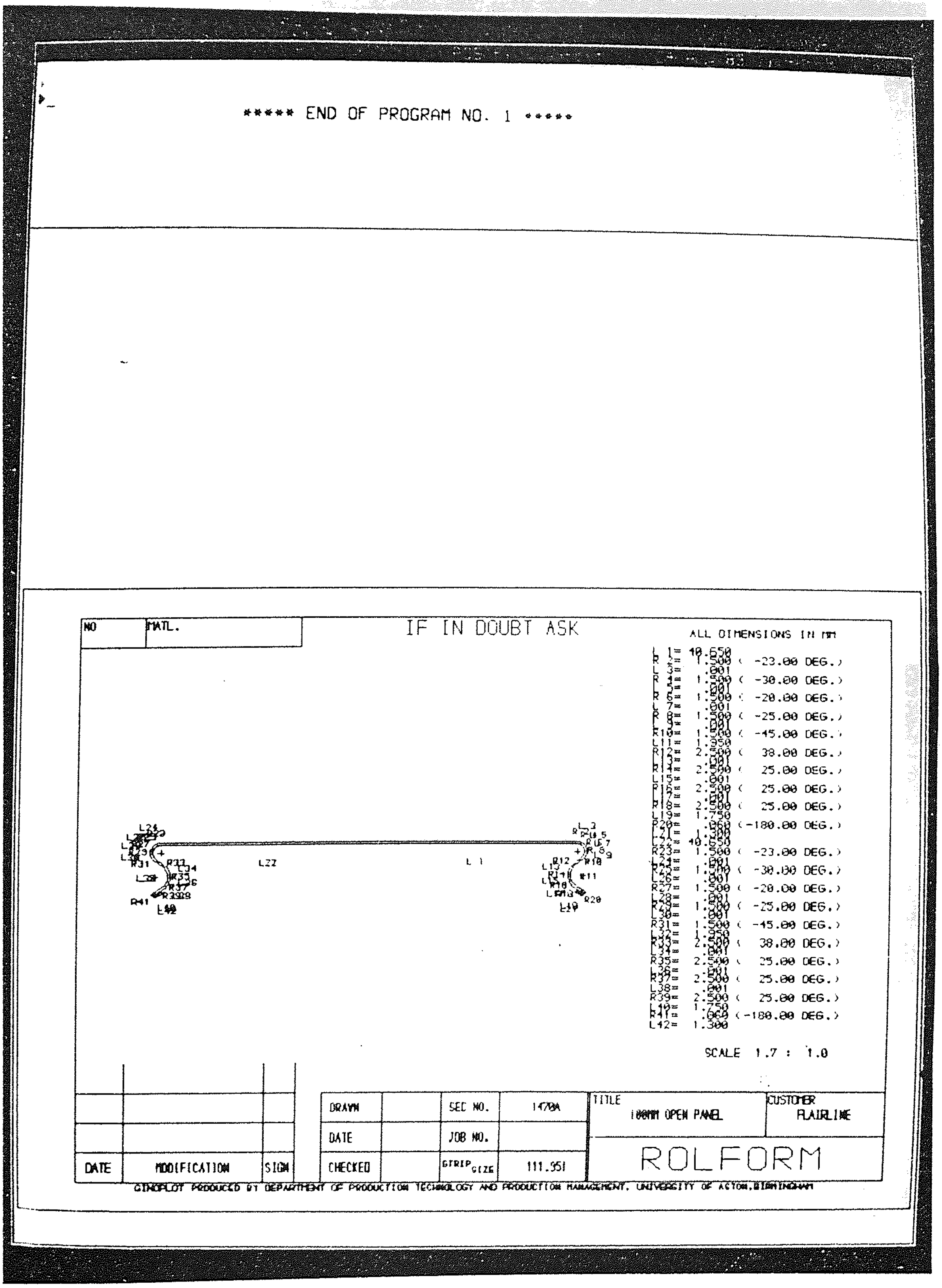


Plate 5.1 PERQ SCREEN DISPLAY OF A FINISHED SECTION DRAWING

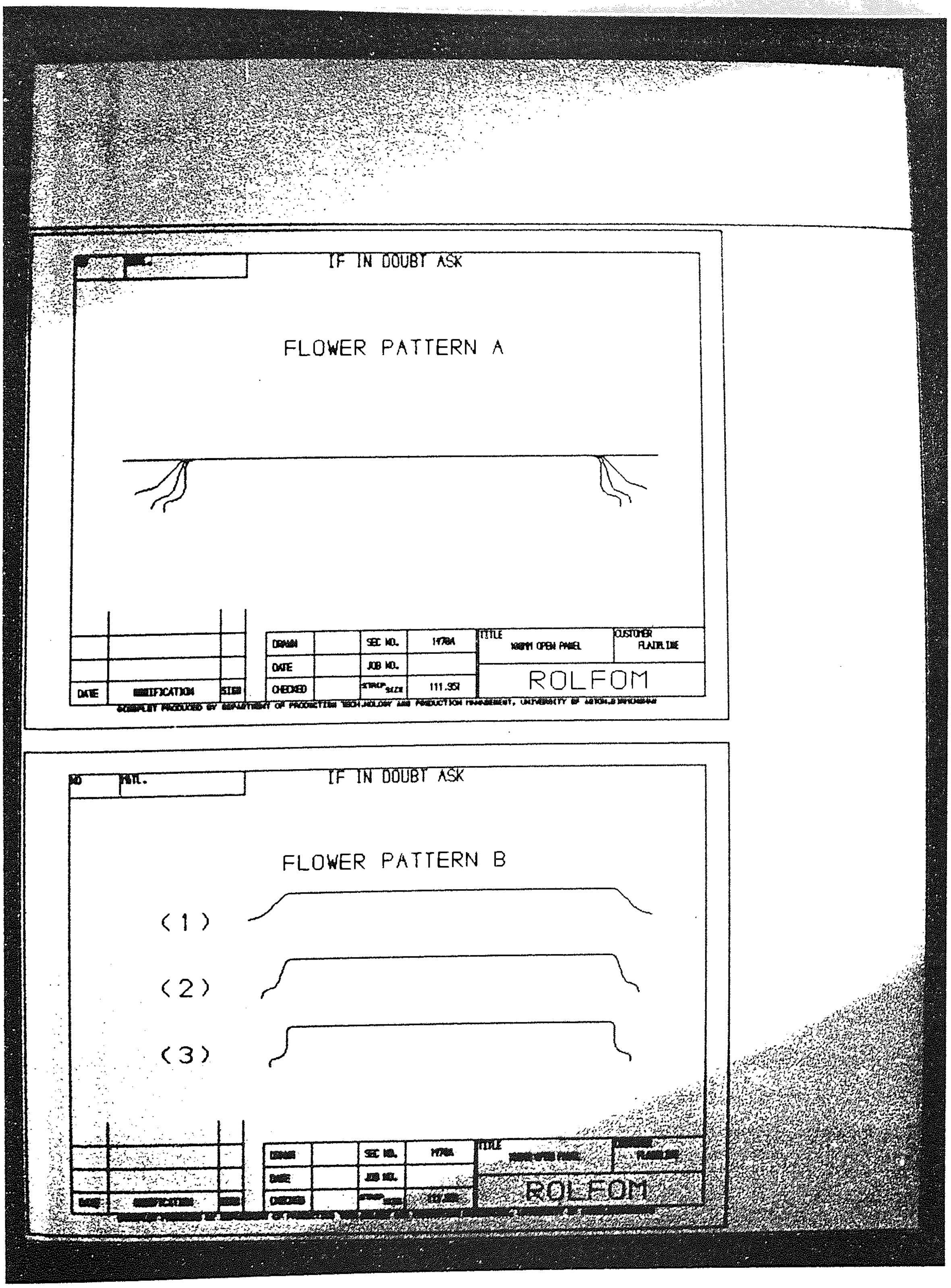


Plate 5.2 PERQ SCREEN DISPLAY OF A FLOWER PATTERN DRAWING

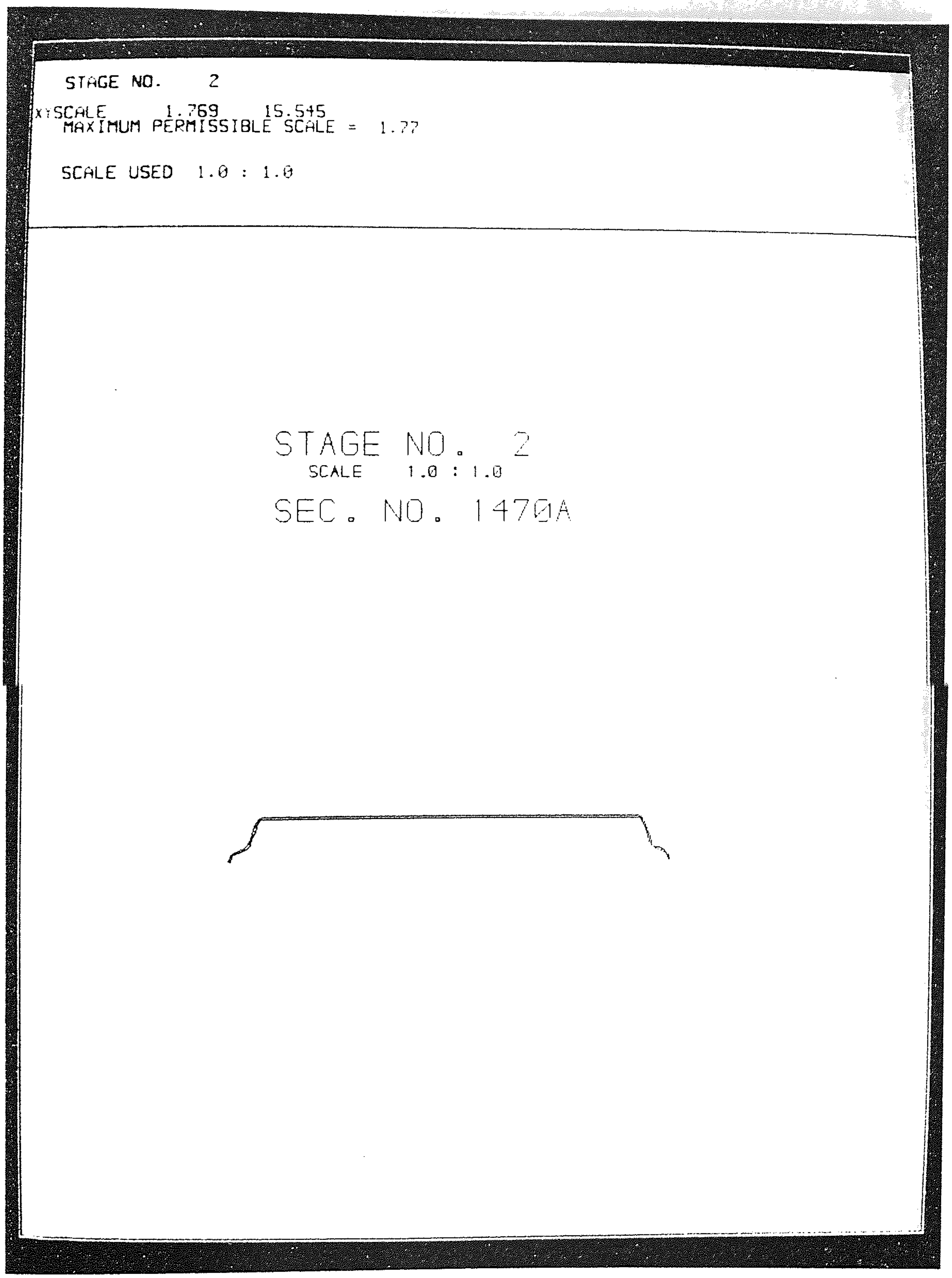


Plate 5.3 PERQ SCREEN DISPLAY OF A TEMPLATE DRAWING

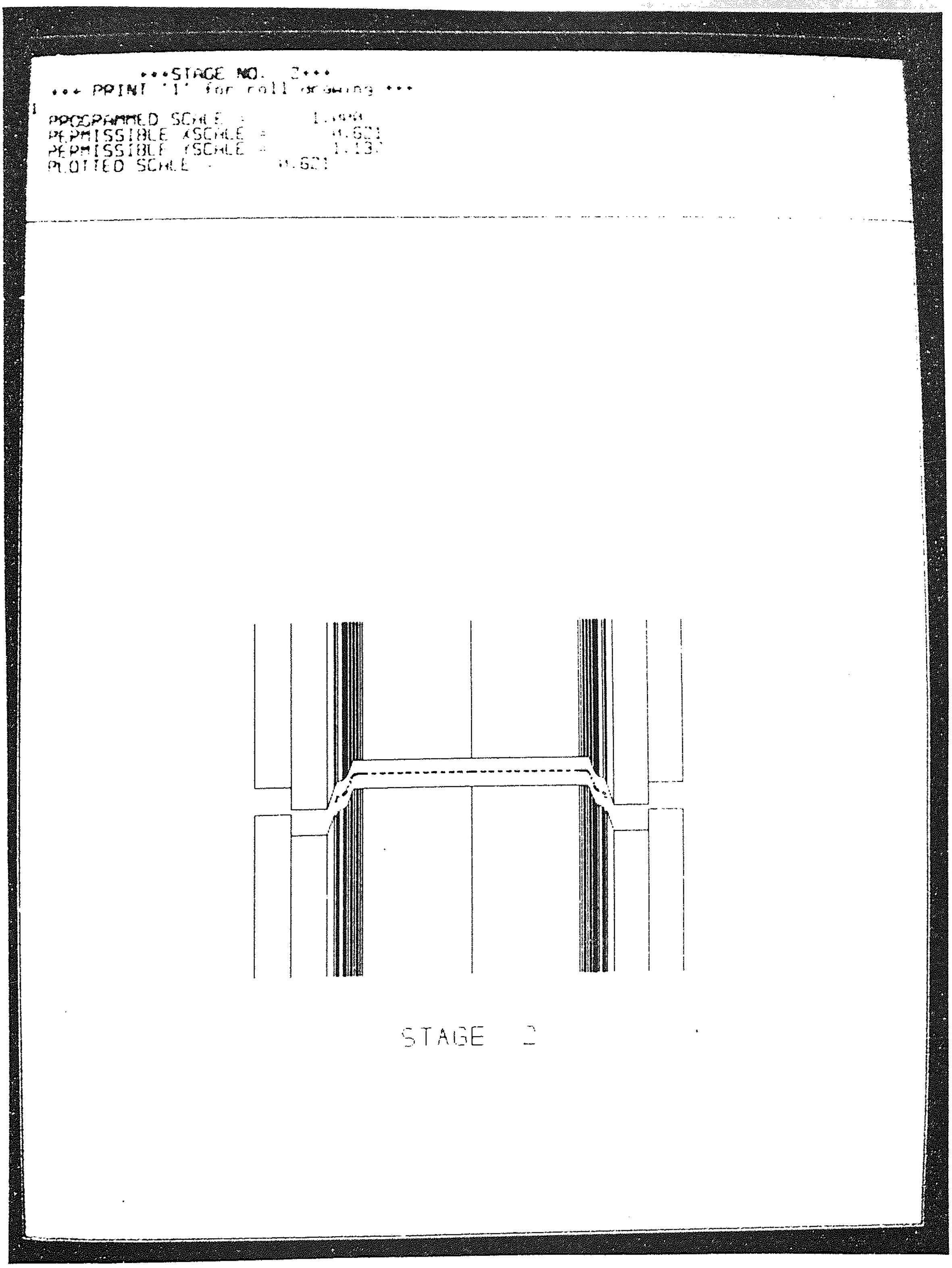


Plate 5.4 PERQ SCREEN DISPLAY OF A ROLL DESIGN DRAWING FOR ONE OF THE FORMING STAGES

CHAPTER 6

CHAPTER 6

EDITING THE ROLL PROFILES

6.1 INTRODUCTION

Although the roll design program can produce roll design drawings and a digitised data file for all rolls in every stage of the roll forming process, minor editing of individual roll in one form or another is sometimes inevitable. For example, for freedom of design, it may be desirable to modify a sharp corner by a blending circular arc or a chamfer, or to perform some final refinements to the roll profile.

The roll editor program, which is a very useful and powerful tool for the roll designer, has therefore been developed for executing random and minor design changes and final adjustments. In order to standardise the procedure in using the CAD/CAM package and to avoid confusion, it has been decided that the roll editor program has to be called upon, despite whether editing is needed, to create an individual digitised roll profile data file for every single roll in all rolling stages.

According to the requirements and general practices of the company, the following functions had been incorporated into the roll editor program:

(1) Inserting a new element, either linear or circular, to the profile.

(2) Replacing an existing element by a new element.

(3) Deleting an existing element.

(4) Modifying a corner by adding a chamfer or blend arc.

As mentioned earlier, all roll profile elements, similar to the elements constituting a thin-section, should either be linear or circular. Thus, the roll editor program has been equipped with facilities to handle linear and circular elements, which should be sufficient for all cases regarding form-roll design.

6.2 DATA INPUT AND OUTPUT

As explained in the previous chapter, the roll design program will generate a data file for each set of rolls. This file contains the digitised roll profile data for all rolls (top roll, bottom roll, left side roll and right side roll) in every rolling stage. Hence, while the roll editor program is being operated, appropriate coding numbers are required to be specified by the user. These codes include the part number of the section for picking up the particular thin-section; the number of the rolling stage for specifying the rolling stage concerned; and the roller number for selecting among the top, bottom, left or right side rolls, as to specify the roll which is desired. Consequently, the system will then trace back to the corresponding set of digitised roll profile data for the specified roll. This set of geometric data will then be decoded and put in the memory storage, while the shape of the roll profile will be displayed on the PERQ screen. The roll designer can then select the editing function in one type or another, with a step by step approach. The software will update the roll data structure as well as the roll profile plot on the display, after each step of editing. Until finally, if the roll designer decides to leave the roll editor program, the roll profile data structure will be dumped into a data file for the edited roll.

6.3 ROLL PROFILE DATA STRUCTURE

The roll profile data structure has been designed in such a way that the leftest contour element is always the first element. Figure 6.1 shows the co-ordinate system for a roll profile drawing. In the memory storage, the roll profile is being represented by a set of linear and circular elements joining each other, from left to right, in ascending order.

Generally, co-ordinates of starting and ending points are required for locating an element. In addition, a straight line in a 2-D co-ordinate system is represented by the general equation $AX + BY + C = 0$, the constants A, B, and C are therefore being stored in the memory storage to represent a linear element. On the other hand, for circular elements, the co-ordinates (F, G) for the centre point together with its radius R will be used for describing its geometry.

The data structure has thus been designed to hold all information about the element identity and status, plus all reference data related to the element. Actually, the data structure comprises of several data substores (see Table 6.1). The substore NEL is an integer array for identifying element numbers from 1 to 100; the substore ITYPE, which is the array for identifying the type of element, whereby ITYPE(n) equals 1 for linear elements, or ITYPE(n) equals 2 for circular elements, and the subscript n is the element number. Besides, the other substore RXA is an array for holding all geometric information about the element. For circular elements, another substore, IDIR, being an array for identifying the direction of the arc, has to be used.

6.4 SOME GEOMETRIC PRINCIPLES AND PROCEDURES APPLYING IN THE ROLL EDITOR PROGRAM

For the roll editor program, just like all other computer graphics software, formulations and algorithms impose significant requirements on the utilisation of geometric principles and procedures. However, as the roll profile comprises of linear and circular elements, several mathematical algorithms and procedures have thus been developed, to deal with lines and circles in analytic geometry, under different situations and constraints. In the roll editor program, thereafter, sub-routines have been developed to incorporate these algorithms.

The purpose of this section is to introduce some of the commonly used geometric algorithms, which have been developed for handling roll profile geometry, and have been inscribed into the roll editor program.

6.4.1 The Basic Line Equation

Throughout this program, as described in Section 6.3, all linear elements are represented by the general equation

$$AX + BY + C = 0 \qquad (6.1)$$

A line can therefore be described by a precise mathematical model if the coefficients A, B, and C are all known.

6.4.2 The Basic Circle Equation

To represent circular elements, the co-ordinates (F, G) of the centre and the radius R should be given. Therefore, the following general circle-defining equation has been used throughout the program:

$$(X - F)^2 + (Y - G)^2 = R^2 \qquad (6.2)$$

6.4.3 Intersection Between Two Linear Elements

As there are various occasions where the intersecting point between two lines has to be determined, a module has then been developed for carrying out the computation. If the equations for the two lines given by

$$A_1 X + B_1 Y + C_1 = 0 \qquad (6.3a)$$

and

$$A_2 X + B_1 Y + C_1 = 0 \qquad (6.3b)$$

if these two lines are not parallel, that means if

$$\frac{A_1}{B_1} \neq \frac{A_2}{B_2},$$

the co-ordinates (x, y) of the intersecting point are given by

$$x = \frac{B_1 C_2 - B_2 C_1}{A_1 B_2 - B_1 A_2} \qquad (6.4)$$

and

$$y = \frac{A_2 C_1 - A_1 C_2}{A_1 B_2 - B_1 A_2} \qquad (6.5)$$

6.4.4 Intersection Between a Line and a Circle

As the roll profile geometry comprises of linear and circular elements, it is necessary, therefore, to consider the case of computing the intersection between a line and a circular arc. Considering the line and the circle are represented by their basic equations as given in equation (6.1) and equation (6.2) respectively, the co-ordinates (x, y) for the intersection point can then be determined by solving between these two equations. The solutions for (x, y) can hence be given by

$$x = \frac{-S_1 \pm \sqrt{U_1}}{2T} \tag{6.6}$$

and

$$y = \frac{-S_2 \pm \sqrt{U_2}}{2T} \tag{6.7}$$

where

$$T = A^2 + B^2 \tag{6.8}$$

$$S_1 = 2AC + 2ABG - 2B^2F \tag{6.9}$$

$$S_2 = 2BC + 2ABF - 2A^2G \tag{6.10}$$

$$U_1 = S_1^2 - 4T\ (B^2(F^2 + G^2 - R^2) + C^2 + 2BCG) \tag{6.11}$$

and

$$U_2 = S_2^2 - 4T\ (A^2(F^2 + G^2 - R^2) + C^2 + 2ACF) \tag{6.12}$$

However, depending on the values of U_1 and U_2 in the above equations, there are three different cases. In the first case, if either U_1 or U_2 is less than zero, then obviously, the line and the circle are not inersected. If on the other hand, both of the values for U_1 and U_2 are equal to zero, that means the line is tangent to the circle and there is only one solution for (x, y) as given by equations (6.6) and (6.7).

Now, the case will be more complicated if the values for U_1 and/or U_2 are larger than zero. That means there are two intersecting points between the line and the circle, and the user, therefore, has to interact with the program to select the appropriate solutions:

(a) If in the first place it happens that the line is a vertical line, U_1 therefore equals zero while U_2 is larger than zero, then there is only one solution for x from equation (6.6) and there are two solutions for y as shown in equation (6.7). Simply, if the desired solution is the intersection point that located above the other one, the plus sign would be chosen for the square root of U_2 in equation (6.7), because that point should have a larger Y-co-ordinate. Conversely, the minus sign would be chosen if the desired solution is specified to be the one located below.

(b) Besides, if U_1 is larger than zero but U_2 equals zero, that is, the line is a horizontal line, and there are two solutions for x from equation (6.6) but there is one solution for y from equation (6.7). Therefore the user has to specify which one of the two intersection points, whether the one on the left or on the right, should be selected. The plus sign would be used in equation (6.6) if the intersection on the right is desired, and the minus sign would be used for the left point.

(c) Lastly, if both U_1 and U_2 are larger than zero, the user is again required to select between the two intersecting points. In this case, both of the X-co-ordinates and Y-co-ordinates for the two intersecting points are not equal. Therefore if the point is chosen to be the one located above, the solution for (x, y) will be given by

$$y = \frac{-S_2 + \sqrt{U_2}}{2T} \tag{6.13}$$

while $$x = \frac{-(By + C)}{A} \tag{6.14}$$

For the intersection that locates below,

$$y = \frac{-S_2 - \sqrt{U_2}}{2T} \tag{6.15}$$

while $$x = \frac{-(By + C)}{A} \tag{6.14}$$

For the intersection that locates on the right,

$$x = \frac{-S_1 + \sqrt{U_1}}{2T} \tag{6.16}$$

while $$y = \frac{-(Ax + C)}{B} \tag{6.17}$$

Finally for the location that locates on the left,

$$x = \frac{-S_1 - \sqrt{U_1}}{2T} \tag{6.18}$$

while $$y = \frac{-(Ax + C)}{B} \tag{6.17}$$

6.4.5 Intersection Between Two Circular Elements

For the case when two circular arcs are joining together, again, there exists three possibilities as there may be two intersecting points; one intersecting point; or the two circles are not intersecting at all. However, the algorithm for this case has been simplified by exploiting the algorithm for obtaining intersection between a line and a circle (see Section 6.4.4).

If the equations for the two circles are given by

$$(X - F_1)^2 + (Y - G_1)^2 = R_1^2 \tag{6.19}$$

and

$$(X - F_2)^2 + (Y - G)_2^2 = R_2^2 \tag{6.20}$$

The equation for the common chord (that is, the line joining the two intersection points) between the two circles can be obtained by subtracting equation (6.20) from equation (6.19), which gives

$$2(F_2-F_1)X + 2(G_2-G_1)Y + (F_1^2+G_1^2-R_1^2-F_2^2-G_2^2+R_2^2) = 0 \tag{6.21}$$

which is actually a general line-defining equation, analogous to equation (6.1), whereas we can let

$$A = 2(F_2 - F_1) \tag{6.22}$$

$$B = 2(G_2 - G_1) \tag{6.23}$$

and

$$C = F_1^2 + G_1^2 - R_1^2 - F_2^2 - G_2^2 + R_2^2 \tag{6.24}$$

Henceforth, the algorithm explained in Section 6.4.4 can be applied to determine the desired intersecting points between the common chord and anyone of the two circles. This method is valid, as the common chord is in fact the line joining the two intersection points between the two circles.

6.4.6 Identifier for a Line Parallel to a Given Line

In the roll editor software, there are various occasions, whereby the equation of a line has to be determined, according to the constraint that the line is parallel to and at a given distance away from a given line. As a matter of fact, there are two lines parallel to, both at the same distance away from, but on opposite sides of a given line. Hence it is necessary to specify the position of the required solution by using an identifier, that is, being 'A', 'B', 'L' or 'R', to specify whether the new line is (A)bove, (B)elow, on the (L)eft or on the (R)ight of the given line. Now, consider a line L_1 is defined by

$$AX + BY + C = 0$$

if a line L_2 is parallel to L_1, and at a distance D away from it, the equation for L_1 will be

$$AX + BY + C \pm D \sqrt{A^2 + B^2} = 0 \qquad (6.25)$$

With regard to the plus or minus signs in the above equation, if the line L_2 is specified to be above L_1, we should choose a sign which is opposite to that of B, as to give a larger value of Y. Or, we can rewrite equation (6.25) in the form

$$AX + BY + C + Jsign * D \sqrt{A^2 + B^2} = 0 \qquad (6.26)$$

Therefore, in this case for L_2 located above L_1, we have

$$Jsign = -B/|B| \qquad (6.27)$$

conversely, if the line L_2 is specified to be below of the given line

L_1, then the sign should be such that it will result in a smaller value of Y,

and $$Jsign = +B/|B| \tag{6.28}$$

on the other hand, if the new line L_2 is specified to be on the left hand side of the given line L_1, we should then choose a sign which is the same as that for A, so as to give a smaller value of X in equation (6.25). Or, as in equation (6.26), we use

$$Jsign = +A/|A| \tag{6.29}$$

Lastly, if the line L_2 is specified to be on the right hand side of L_1, we have to obtain a larger value for the X-co-ordinate in this case, hence, we use

$$Jsign = -A/|A| \tag{6.30}$$

In the roll editor program, the above described procedures have been incorporated into a subroutine called DSIGN, which will generate the appropriate value for the code Jsign automatically, utilising equations (6.27) to (6.30).

6.5 DEFINING NEW ELEMENTS

When the roll editor program is in the Insertion or Replacement mode, it is necessary to input the geometric data for the new element. As the new element can only be either a circular arc or a line, facilities for defining lines and circles have thus been developed and installed in the roll editor program.

Although there are many different ways for prescribing circles and lines in a 2-D co-ordinate system, consideration has been taken on the general roll profile geometry, and the fact that roll designers are not mathematicians or experienced computer programmers, hence only those simple and commonly used methods had been selected for defining new elements. Algorithms and subroutines have thus been established for this purpose.

6.5.1 Defining a Linear Element

There exists many methods for defining a line in the 2-D co-ordinate system. For simplicity, however, three methods have been adopted in the roll editor program for defining linear elements. Therefore, in the roll editor program, a linear element can be defined as:

(1) A line passing through two given points (Fig. 6.2).

(2) A line through a point that makes an angle with the X-axis (Fig. 6.3).

(3) A line through a point and parallel to a given line (Fig. 6.4).

The algorithms for the three methods are explained in the following:

(1) Line Passing Through Two Given Points (Fig. 6.2)

For the first definition, whereby a line is defined by two points, if the co-ordinates for the two given points are (X_1, Y_1) and (X_2, Y_2) respectively, and the general equation of a line as given in equation (6.1), is

$$AX + BY + C = 0 \tag{6.1}$$

By substituting the co-ordinates of the two given points in equation (6.1), the coefficients A, B, and C are determined by solving the equations obtaining

$$A = Y_2 - Y_1 \tag{6.31}$$

$$B = X_1 - X_2 \tag{6.32}$$

$$C = X_2Y_1 - X_1Y_2 \tag{6.33}$$

(2) Line Through a Point that Makes an Angle with the X-Axis (Fig. 6.3)

In this method, the angle θ of the line is given together with a point (X_1, Y_1). The gradient m of the line can therefore be computed by

$$m = \tan(\theta) \tag{6.34}$$

Solving the equation for point slope form of a line, the coefficients can then be obtained. Depending on the value of m, there are two different cases:

Firstly, for $m \neq \infty$,

$$A = -m \tag{6.35}$$

$$B = 1 \tag{6.36}$$

$$c = -Y_1 + mX_1 \tag{6.37}$$

or secondly, for $m = \infty$ (i.e. $\theta = 90^\circ, 270^\circ, -90^\circ, -270^\circ$),

$$A = 1 \tag{6.38}$$

$$B = 0 \tag{6.39}$$

$$C = -X_1 \tag{6.40}$$

(3) Line Through a Point and Parallel to a Given Line (6.4)

In this case, the line is passing through a given point (X_1, Y_1) and parallel to an existing line L_1, where L_1 is described by the form

$$A_1X + B_1Y + C_1 = 0$$

Therefore the coefficients, A and B, for the new line, will be the same as those for the given line L_1, that is,

$$A = A_1 \tag{6.41}$$

$$B = B_1 \tag{6.42}$$

With the given point (X_1, Y_1) and the values of A and B, C can then be calculated by

$$C = -(AX_1 + BY_1) \tag{6.43}$$

6.5.2 Defining a Circular Element

Again, there are various methods to prescribe a circle, in a 2-D co-ordinate system, by giving the specified constraints. By and large, whenever the set of constraints given is sufficient for computing the co-ordinates (F, G) of the centre point and the radius R, the circle definition should then be deemed to be feasible.

In the roll editor program, after examining the general requirements by the roll profiles, four different methods for circle definition have been adopted in this software. Although there are many other different mathematical models for circle definition, these four methods can provide sufficient versatility, to the user, in defining new circular elements. In essence, the four circle definitions are:

(1) Circle with known co-ordinates of the centre and also with known radius.

(2) Circle with known co-ordinates for its centre and passing through a given point.

(3) Translation of an existing circle.

(4) Circle tangent to two existing elements (linear or circular), with known radius.

The mathematical models for the above mentioned four circle-defining algorithms are given as follows:

Case (1) <u>Circle with Known Co-ordinates for its Centre, with Known Radius</u>

Referring back to the basic circle-defining equation as given in equation (6.2), the geometric model for the circle can thus be established with known values for the centre point co-ordinates (F, G) and the radius R. This method is obviously the most straightforward and simplest way to prescribe a circle (Fig. 6.5).

Case (2) Circle Passing Through a Given Point, with Known Co-ordinates for its Centre (Fig. 6.6)

In this case, the co-ordinates (F, G) for the centre point are given, and as the co-ordinates (x, y) of a point on its circumference is given, the radius R on the circle can be computed by the equation

$$R = \sqrt{(F - x)^2 + (G - y)^2} \qquad (6.44)$$

Case (3) Circle which is the Translation of an Existing Circle (Fig. 6.7)

Suppose that there exists a circle with centre point co-ordinates (F', G') and radius R. A new circle, which is obtained by translating the existing circle through a distance (x, y) in the rectangular co-ordinate system, should have the same radius R. The co-ordinates (F, G) for the centre point of the new circle, however, has to be obtained by the equations:

$$F = F' + x \qquad (6.45)$$

and

$$G = G' + y \qquad (6.46)$$

Case (4) Circle Tangent to Two Existing Elements, with Known Radius

In this case, the new circular element is tangential to both of its preceding and succeeding elements. Depending on the type of these two elements, however, there are three different possibilities:

(a) Both of the preceding and succeeding elements are lines - In this case, the newly input circular element is tangential to two known lines, with known radius. This case is depicted in Fig. 6.8. Apparently, there are altogether four possible solutions, depending on the location of the circle with respect to the two lines. Thus, it is required to specify the location of the circle. Again, identification codes, for indicating whether the ideal solution is above; below; on the right; or on the left hand side of the two lines respectively, can be used for this purpose. Thus, suppose the two lines are given by

$$A_1X + B_1Y + C_1 = 0$$

and $$A_2X + B_2Y + C_2 = 0$$

As the circle is of given radius R, we can therefore derive the equations of parallel lines at a distance R that will intersect. This solution will be R units from both original lines and therefore is the centre of the tangential circle. The two equations (parallel to the given lines) become

$$A_1X + B_1Y + C_1' = 0 \qquad (6.47a)$$

and $$A_2X + B_2Y + C_2' = 0 \qquad (6.47b)$$

where $$C_1' = C_1 \pm R\sqrt{A_1^2 + B_1^2} \qquad (6.48)$$

$$C_2' = C_2 \pm R\sqrt{A_2^2 + B_2^2} \qquad (6.49)$$

The plus or minus sign in equations (6.48) and (6.49) would be chosen automatically, by using the algorithm DSIGN as described in Section 6.4.6, according to the identification codes for the location of the circle relative to the two lines respectively. With the known values of C_1' and C_2', therefore, the two equations (6.47a) and (6.47b) may be solved simultaneously, as described in Section 6.4.3, to derive the co-ordinates (F, G) for the circle centre.

(b) Both of the preceding and succeeding elements are circles - Fig. 6.9 depicts this case. As can be seen, there may be more than one circle with known radius, tangent to two given circles. Hence, user intervention is needed, to specify the location of the desired circle with respect to the two given circles respectively, for selecting the particular solution. To begin with, it is necessary to specify whether the new circular element is located inside or outside of the two given circles respectively. This can be done by utilising the appropriate identification code, for example the code 'I' should be used for locating the new circle inside of the given circle, while 'O' represents the outside case. Suppose the two circles are given by

$$(X - F_1)^2 + (Y - G_1)^2 = R_1^2$$

and $$(X - F_2)^2 + (Y - G_2)^2 = R_2^{\ 2}$$

The given radius for the new circle is R. Therefore the centre point for this new circular element is the intersection between two circles, which are concentric to the two given circles respectively, with the radii being $(R_1 \pm R)$ and $(R_2 \pm R)$. The equations for these two circles are

$$(X - F_1)^2 + (Y - G_1)^2 = (R_1 \pm R)^2 \qquad (6.50)$$

and $$(X - F_2)^2 + (Y - G_2)^2 = (R_2 \pm R)^2 \qquad (6.51)$$

The plus or minus signs in equations (6.50) and (6.51) would be chosen, according to whether the new circle is locating outside or inside of the given circles respectively.

In order to obtain the co-ordinates for the centre of the new circular element, it is necessary to determine, and select the desired one from the two possible intersecting points, if any, between the two circles as given by equations (6.50) and (6.51). In fact, the algorithm described in Section 6.4.5 can be used for this computation.

(c) One of the preceding and succeeding elements is linear, while the other one is circular - This case is shown in Fig. 6.10. Suppose we have the equations of the given line and the circle, they are:

$$AX + BY + C = 0$$

and $$(X - F_1)^2 + (Y - G_1)^2 = R_1^{\,2}$$

The given radius of the new circle is R. Therefore the centre of it will be R units from both of the line and the circle given, thus we have

$$AX + BY + C \pm R\sqrt{A^2 + B^2} = 0 \qquad (6.52)$$

and $$(X - F_1)^2 + (Y - G_1)^2 = (R_1 \pm R)^2 \qquad (6.53)$$

In equation (6.52), the plus or minus sign would be determined by the DSIGN algorithm mentioned in Section 6.4.6, depending on the location of the new circle with respect to the given line. For equation (6.53), on the other hand, the plus sign would be used if the new circle is located outside of the given one, while the minus sign would be used in the inside case.

Henceforth, utilising the procedures for obtaining intersection between a line and a circle, as explained in Section 6.4.4, the centre point co-ordinates (F, G) for the new circle can then be determined.

6.6 INSERTING AN ELEMENT

With the roll editor program, the user can choose to insert an element after an existing element. For defining the new element, which can either be a line or a circular arc, the principles and procedures described in Section 6.5 can be applied to obtain the canonical data - that is - the centre co-ordinates (F, G) and radius R for a circle, or the coefficients A, B and C for a line. After establishing the precise mathematical model of the new element, the system will then calculate the starting and ending points of the new element, and hence update the complete data structure of the edited roll profile and draw the new profile on the PERQ screen. The starting point of a new element, which is the ending point of its preceding element, is obtained by calculating the intersecting point between the new element and the preceding element. On the other hand, the ending point of the new element, being the starting point of the succeeding element, can be determined by computing the intersecting point between these two elements. Again, the algorithms developed in Section 6.4 can be applied here for the calculations.

6.7 REPLACING AN ELEMENT

The roll designer can also replace an existing element by a new element, which can either be a line or a circular arc. Similar to the Insert option mentioned in the previous section, the user needs only to define the new element and the system will update the complete digitised roll profile data structure by carrying out the same drill described in the previous section. The original element will be deleted automatically, following a successful replacement by the new element.

6.8 DELETING AN ELEMENT

If an element has to be deleted, its preceding and succeeding elements will then become adjacent elements. The intersecting point between these two adjacent elements is the new ending point for the preceding element, as well as the new starting point for the succeeding element. As in the Insert and Replace options described in the previous two sections, the system will update the data structure accordingly.

6.9 CORNER MODIFICATION

This facility for corner modification is purposely developed for modiyfing the sharp corners in the form-roll profiles. Modifications can either be done by adding a chamfer, or otherwise, a blending arc to the corner concerned. However, according to the geometry of the roll profiles and to avoid cumbersome mathematics, restrictions have been set in the corner modification option. Thus, the corner at the intersecting point with the next roll contour element can be modified, provided that the following conditions can be satisfied, that is:

(1) Both of the elements forming the sharp corner should be linear. In other words, corner modification can only be done at line - line intersection.

(2) The including angle between the two intersecting lines should be less than 135^{o}.

(3) The depth of the chamfer or the blending radius cannot be larger then 60% of anyone of the lengths of the two intersecting elements.

In spite of the above mentioned restrictions, corner modification is in fact a special case of inserting an element, either linear or circular, between two linear elements. Once the corner modification option has been chosen, it is required, however, to specify whether a chamfer or a blending arc is desired. For chamferring, the depth of chamfer C should be specified (Fig. 6.11), while for a blending arc, on the other hand, its blending radius R should be given (Fig. 6.12).

When two given lines L_1 and L_2 are joining together to form a sharp corner, two other lines L_1' and L_2' can be constructed at a distance D away from L_1 and L_2, and parallel to them respectively. In other words, L_1' and L_2' are the offset images of L_1 and L_2 respectively (Fig. 6.13). If the equations for L_1 and L_2 are given by

$$A_1X + B_1Y + C_1 = 0$$

and $$A_2X + B_2Y + C_2 = 0$$

the equations of the offset lines L_1' and L_2' will then be

$$A_1X + B_1Y + C_1 \pm D\sqrt{A_1^2 + B_1^2} = 0 \qquad (6.54)$$

$$A_2X + B_2Y + C_2 \pm D \sqrt{A_2^2 + B_2^2} = 0 \qquad (6.55)$$

The plus or minus sign in equation (6.54) and (6.55) will be determined automatically by the DSIGN algorithm (see Section 6.4.6), according to the relative position between L_1 and L_2.

Suppose L_1 joins two points (X_1, Y_1) and (X_2, Y_2) while L_2 joins (X_2, Y_2) and (X_3, Y_3). Referring to the co-ordinate system for roll profiles (Fig. 6.1), the starting point of an element should be located on the left hand side of the ending point, with the exception for a vertical element, whereby the X-co-ordinates for both of its starting and ending points are equal. Hence there exists four different cases for the relative positions between the points (X_1, Y_1), (X_2, Y_2) and (X_3, Y_3), as depicted in Fig. 6.14. Referring to this figure, the correct locations for L_1' and L_2', relative to L_1 and L_2 respectively, can be deduced in each of the four cases. The positions for L_1' and L_2' will be represented by the appropriate identification codes, $IDEN_1$ and $IDEN_2$ respectively. With the DSIGN algorithm, these tracking symbols $IDEN_1$ and $IDEN_2$, will be interpreted for the determination of the plus or minus signs used in equation (6.54) and (6.55).

6.9.1 Chamferring

As shown in Fig. 6.11, the chamfer is actually a new linear element joining the two points P and Q, where P is the intersecting point between lines L_1 and L_2', and Q is the intersecting point between L_2 and L_1'. The co-ordinates for P and Q, therefore, can be computed by utilising equations (6.4) and (6.5) as described in Section 6.4.3. In fact, point P is the starting point of the new element and Q is its ending point. The coefficients A, B and C, as in equation (6.1), which is the general line-defining equation for this chamfer, can be computed by using equations (6.31) to (6.33), as in Section 6.5.1.

6.9.2 Blending Arc

The centre of the blending arc with a given radius R, as depicted in Fig. 6.12, is the intersecting point between the lines L_1' and L_2'. Hence, equations (6.4) and (6.5) can be used here again, to compute the co-ordinates (F, G) for the centre point. On the other hand, the starting point of the arc is the intersecting point between the line L_1 and a line, which is perpendicular to L_1, passes through (F, G), and the co-ordinates (X_s, Y_s) of the starting point are computed by

$$X_s = \frac{B_1^2F - A_1B_1G - A_1C_1}{A_1^2 + B_1^2} \tag{6.56}$$

and

$$Y_s = \frac{A_1^2G - A_1B_1F - B_1C_1}{A_1^2 + B_1^2} \tag{6.57}$$

Similarly, the co-ordinates (X_e, Y_e) for the ending point are given by

$$X_e = \frac{B_2^2 F - A_2B_2G - A_2C_2}{A_2^2 + B_2^2} \qquad (6.58)$$

and

$$Y_e = \frac{A_2^2 G - A_2B_2F - B_2C_2}{A_2^2 + B_2^2} \qquad (6.59)$$

6.10 APPLYING INTERACTIVE GRAPHICS FOR THE DISPLAY

The operation of the roll editor program can dynamically change the displayed roll profile drawing. In other words, the roll editor program, which utilises the PERQ's interactive graphics facilities, will display on the terminal screen an updated drawing of the edited roll profile, after each editing function. Besides, the user can also choose to enlarge any selected part of the drawing so as to have a clearer view of any particular details of the roll profile.

The PERQ workstation, as illustrated in Section 4.4, is a high resolution raster system. The main output device is the screen. In addition to the keyboard, the main input device for graphic information is the tablet (Plate 4.3). Regarding software programming, GRAFIKS [57] is a package of programming facilities for the building of graphics programs on the ICL PERQ, with particular provision for interactive graphics. The roll editor program, therefore, exploits all these graphics features of the PERQ, can regenerate and redraw the complete roll profile at each change.

6.10.1 Viewport and Text Frame

The PERQ display (Plate 4.2), with a usable area of approximately 275 mm high by 210 mm wide, has its long side vertical. The picture on the display consists of 1024 horizontal lines, and each line contains 768 picture elements, often abbreviated to pixels. For the roll editor program, the PERQ screen has been divided into two segments (Fig. 6.15). The upper part being the viewport for displaying the roll profile while the lower part is the text frame for commands and user messages. The size of the text frame is 768 x 440 pixels, while the size of the viewport is 768 x 584 pixels. The current roll profile drawing will, hence, be automatically scaled for fitting in this viewport. An example of this display on the PERQ screen for this program is shown in Plate 6.1.

6.10.2 Enlarge the Drawing

Sometimes the roll designer may find it necessary, especially after editing, to have a larger and clearer view of a particular part of the roll profile. Therefore, facilities for selecting a window in a viewport have been incorporated in the roll editor program. The window, together with the enclosed picture segment, will be magnified, as large as possible, to match the size of the viewport. This is achieved by utilising the cursor control facilities provided by GRAFIKS.

For example if the user wants to enlarge a portion of the displayed drawing on the PERQ screen (Fig. 6.16a), he can, firstly, enclose the portion by a window. In choosing the window, the user has to fix the cursor position, by moving the puck on the tablet and pressing the button afterwards, to the desired location of one of the corners of the rectangular window (Fig. 6.16b). Thereafter, by moving the puck so as to locate the cursor position to the opposite corner, the user can observe the construction of a rectangle, drawn by the 'rubberbox' technique by GRAFIKS, joining the two opposite corners (Fig. 6.16c). Consequently, by pressing a button on the puck, the window, in conjunction with its enclosed roll profile drawing portion, will be magnified on the screen (Fig. 6.16d). Clipping will then take place automatically to fit the magnified drawing in the viewport. Concerning the scale of magnification, both of the maximum vertical and horizontal scales of magnification will be calculated respectively, and the minimum of these two scales will be chosen for enlarging the window.

6.11 HARDCOPY OUTPUT

As mentioned in Section 6.2, a digitised roll profile data file will be generated after the editing process. This data file, which will be kept in the disk storage on the PERQ computer, will be retrieved for further processing in the later stage of the CAD/CAM system. If the user wants to keep a hardcopy record for this data file, a printer listing, as shown in Fig. 6.17, can be obtained by printing the file with a printer which has been connected to the PERQ system.

On the other hand, the roll profile drawing itself can be drawn by a plotter, provided that the plotter has been interfaced with the PERQ. In this research, a HP7221C plotter (Plate 4.5) has been used for this purpose. A set of graphic routines, or the software plotter driver, has been developed by the author and implemented in the PERQ system for generating graphics commands for the HP7221C plotter. This plotter driver, called 'HP7221', which is a set of FORTRAN subroutines, is listed in Appendix 1. In Appendix 2, the outline of the 'HP7721' graphic routines and their functions are given. A plotter drawing for a roll profile is shown in Fig. 6.18.

6.12 THE PROGRAM

The roll editor software consists of three distinctive units, namely:

(1) THE ROLL EDITOR PROGRAM (ROEDIT)

(2) THE SCREEN DISPLAY MODULE (PLOTGR)

(3) THE PLOTTER DRAWING SUBROUTINE (ROLDWG)

The hierarchy of the program structure is shown in Chart 6.1, with the main program ROEDIT residing at the first level of execution, controlling the sequence of execution of the subroutines at lower levels. A brief account of the nature of the subroutines used at each level is given in Table 6.2.

6.12.1 The Screen Display Module (PLOTGR)

This module has been designed to draw the roll profile on the PERQ screen. The hierarchy of the subroutines in the PLOTGR module is as shown in Chart 6.2, and a brief account of the nature of each subroutine is given in Table 6.3.

6.12.2 The Plotter Drawing Subroutine (ROLDWG)

This subroutine has been developed to call upon the 'HP7221' plotter driver facilities (Appendix 2) to plot the final edited roll profile by the HP7221C plotter. The drawing can either be plotted on A3 or A4 size paper.

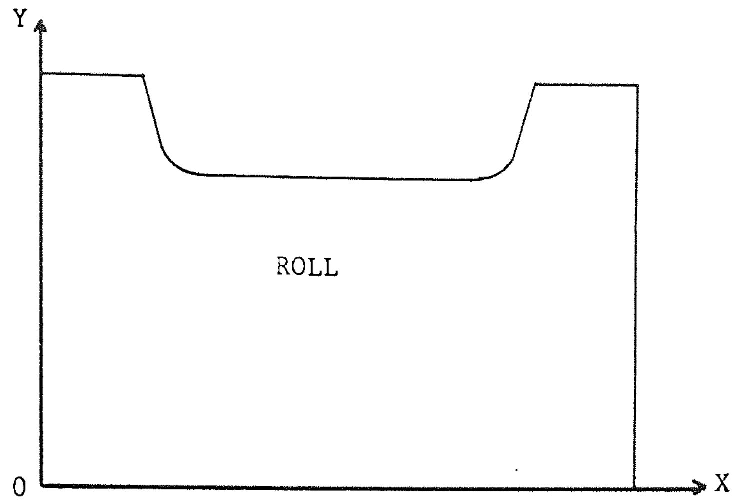


Fig. 6.1 COORDINATE SYSTEM FOR ROLL PROFILE DRAWING

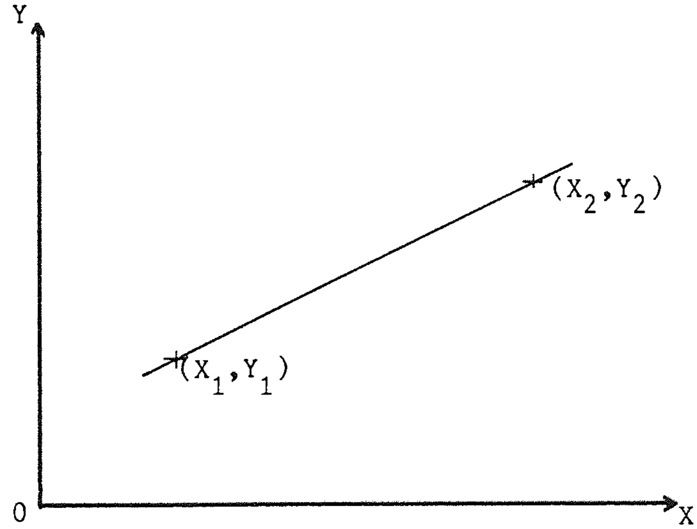


Fig. 6.2 LINE THROUGH TWO POINTS

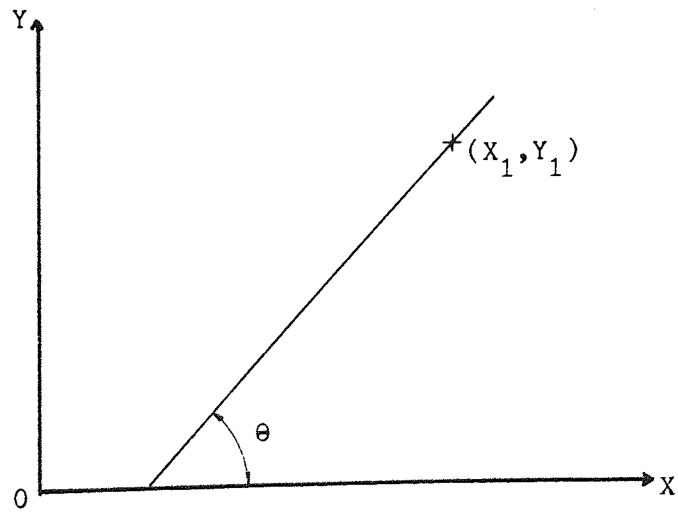


Fig. 6.3 LINE DEFINED BY A POINT AND ANGLE

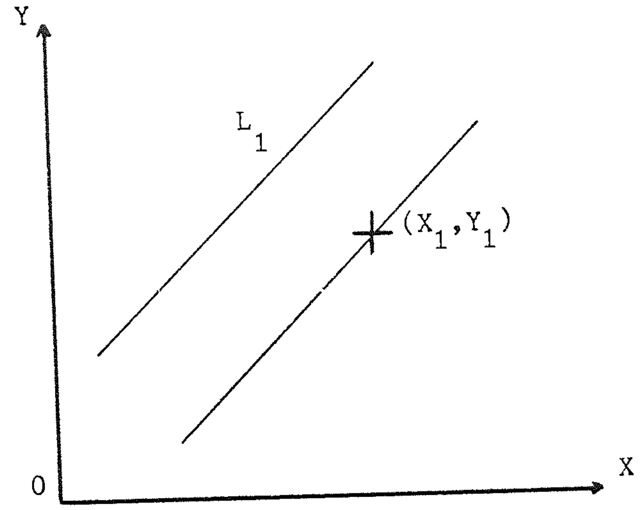


Fig. 6.4 LINE DEFINED THROUGH A POINT AND PARALLEL TO A LINE

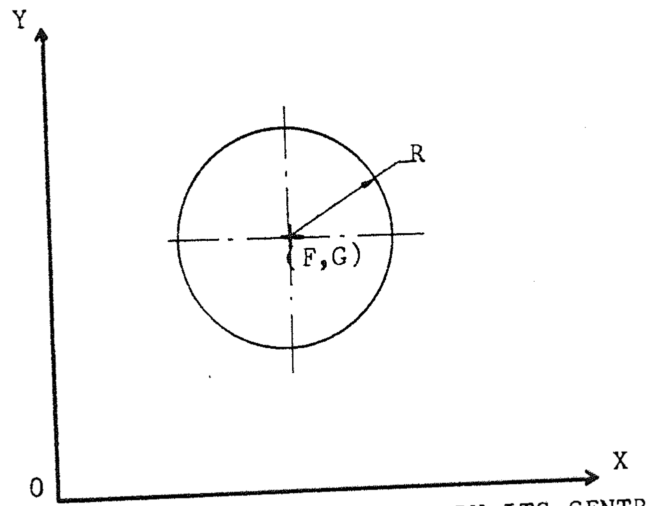


Fig. 6.5 CIRCLE DEFINED BY ITS CENTRE & RADIUS

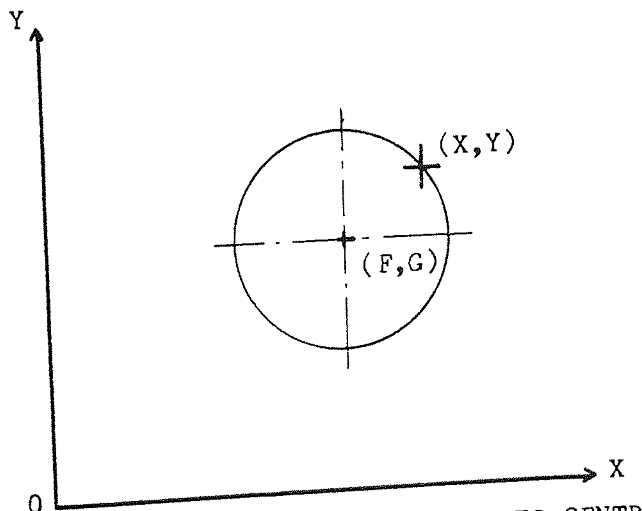


Fig. 6.6 CIRCLE DEFINED BY ITS CENTRE AND THROUGH A POINT

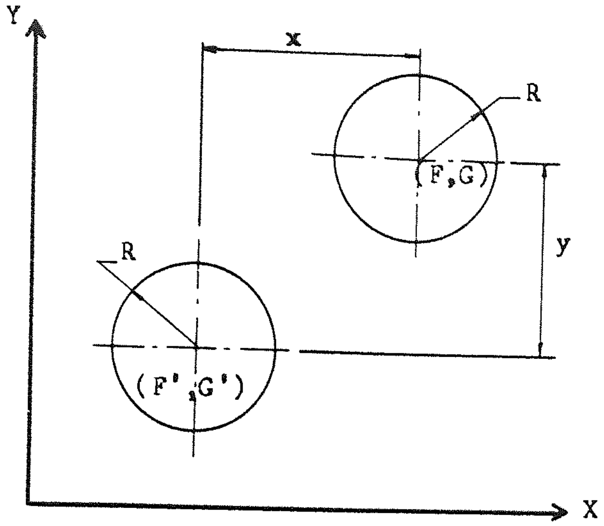


Fig. 6.7 CIRCLE DEFINED AS A GIVEN CIRCLE TRANSLATED

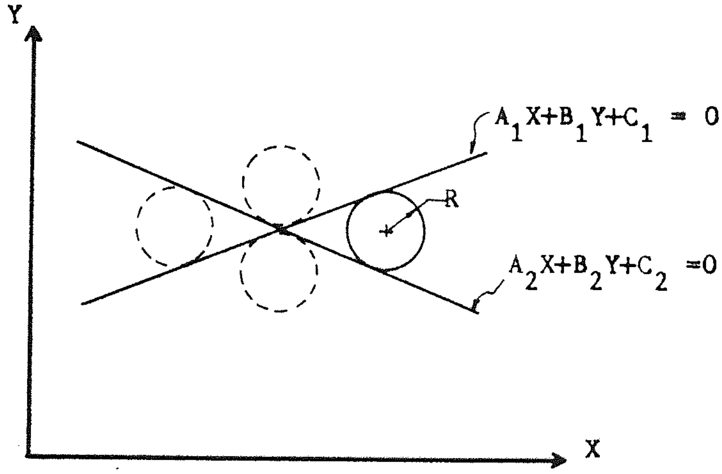


Fig. 6.8 CIRCLE DEFINED TANGENT TO TWO LINES

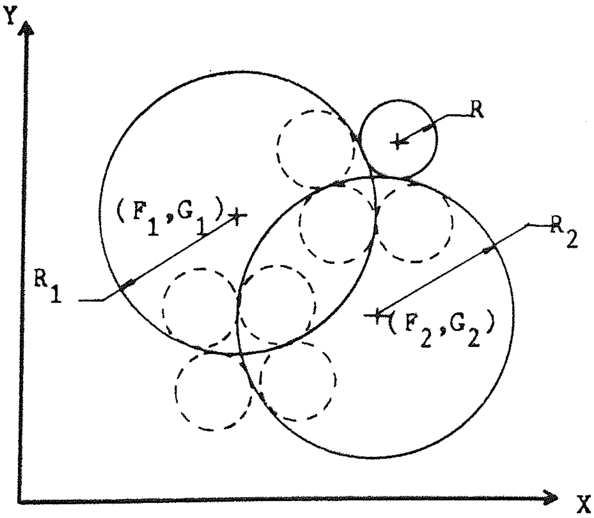


Fig. 6.9 CIRCLE DEFINED TANGENT TO TWO CIRCLES

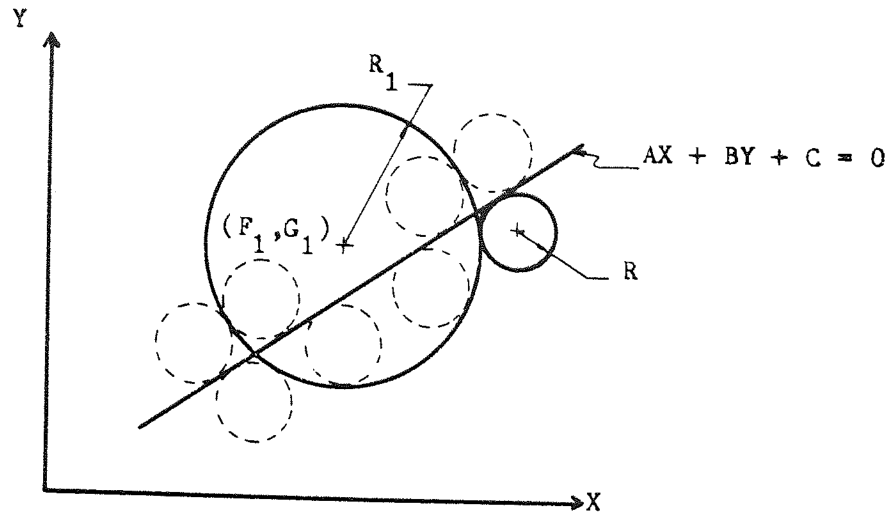


Fig. 6.10 CIRCLE DEFINED THROUGH A LINE AND CIRCLE

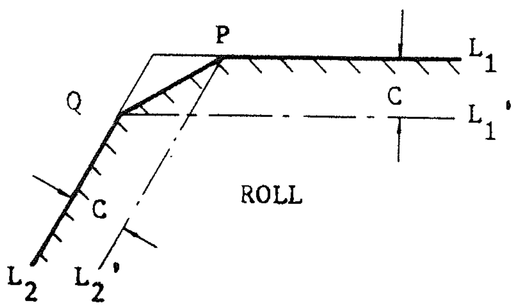


Fig.6.11 CHAMFER

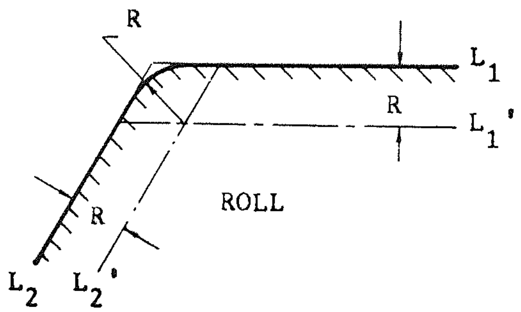


Fig. 6.12 BLENDING ARC

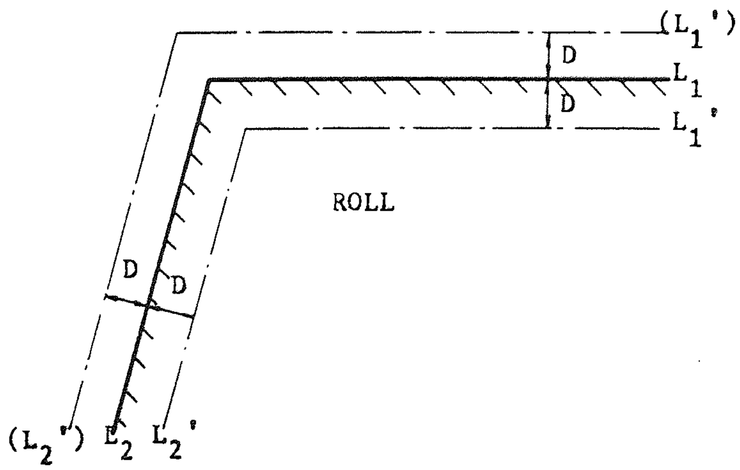
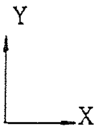
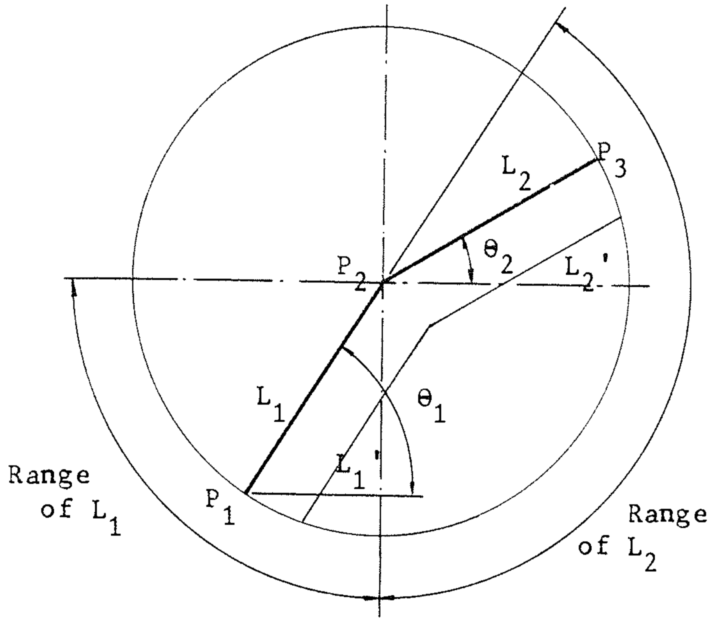


Fig. 6.13 OFFSET IMAGES OF GIVEN LINES



$P_1(x_1, y_1); P_2(x_2,y_2); P_3(x_3,y_3)$



<u>Case 1</u>

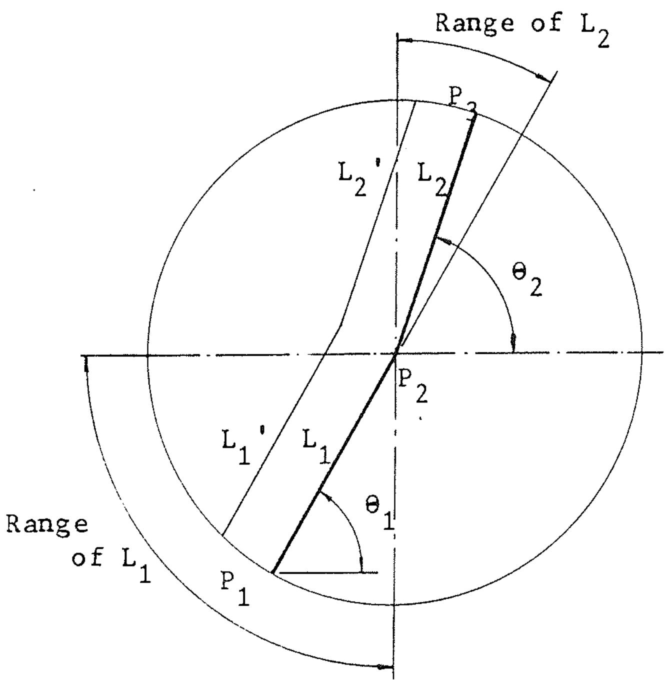
$y_2 \geqslant y_1$ and

$(y_3 \leqslant y_2$ or

$(y_3 > y_2$ & $\theta_1 > \theta_2))$

$IDEN_1$ = 'R' or
= 'B' if $y_2=y_1$

$IDEN_2$ = 'B' or
= 'L' if $x_3=x_2$



<u>Case 2</u>

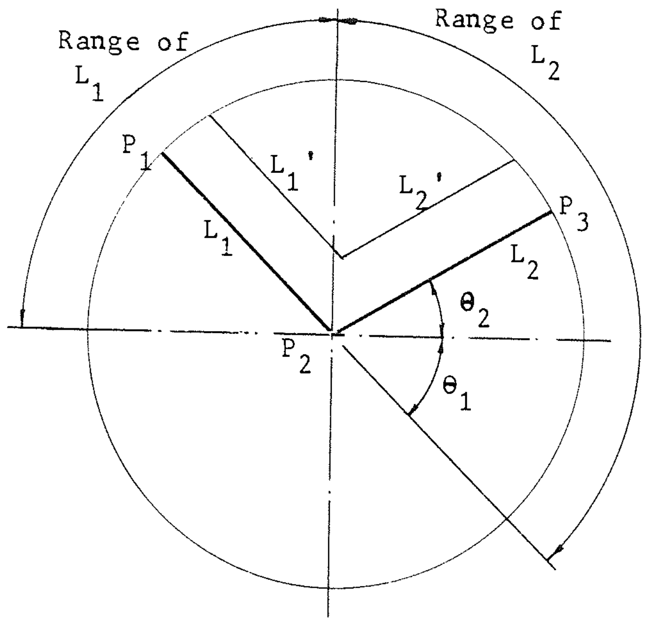
$y_2 \geqslant y_1$ and

$(x_3 > x_2$ & $\theta_2 > \theta_1)$

$IDEN_1$ = 'A' or
= 'L' if $x_2=x_1$

$IDEN_2$ = 'L'

Fig. 6.14 DETERMINATION OF THE RELATIVE LOCATIONS FOR OFFSET PATHS OF THE TWO LINES AT A CORNER (PART 1)



<u>Case 3</u>

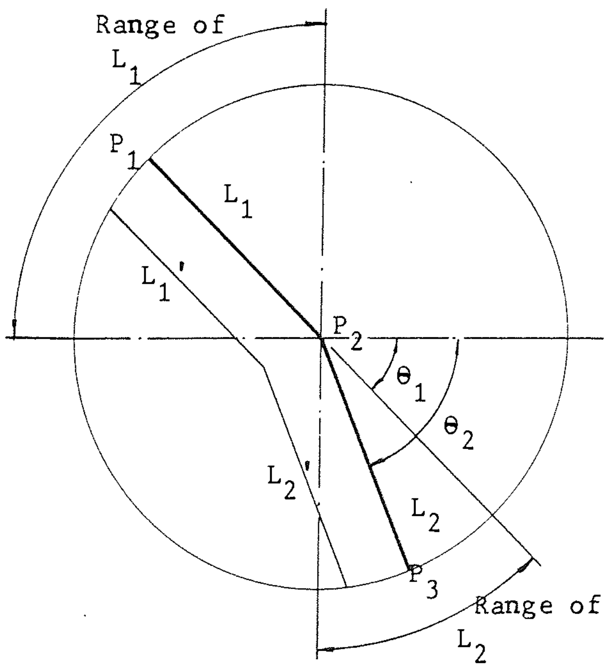
$y_2 < y_1$ and

$(y_3 \geqslant y_2$ or

$(y_3 < y_2$ & $|\theta_2| < |\theta_1|$))

$IDEN_1$ = 'R'

$IDEN_2$ = 'A' or
= 'L' if $x_3 = x_2$



<u>Case 4</u>

$y_2 < y_1$ and

$(y_3 < y_2$ & $|\theta_2| > |\theta_1|$))

$IDEN_1$ = 'L'

$IDEN_2$ = 'L'

Notes : $IDEN_1$ = 'R' for L_1' on the Right of L_1 ,or
= 'L' for L_1' on the Left of L_1 ,or
= 'A' for L_1' Above L_1 , or
= 'B' for L_1' Below L_1

(The same definitions apply to $IDEN_2$)

Fig. 6.14 (PART 2)

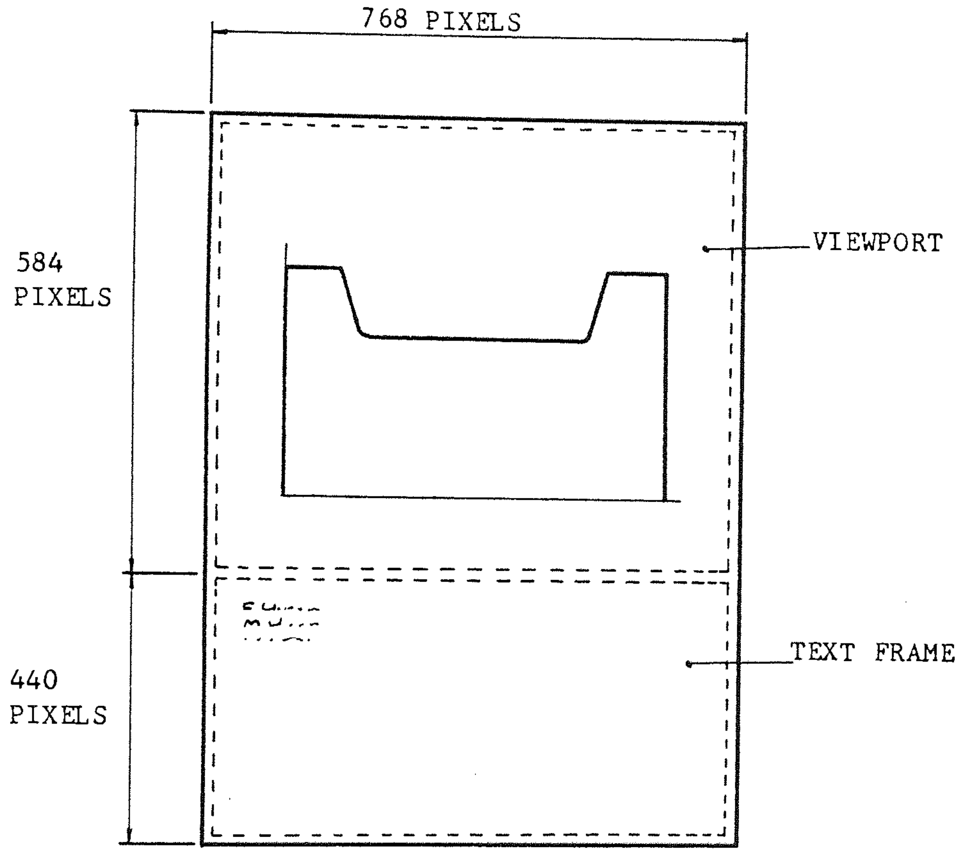


Fig. 6.15 VIEWPORT AND TEXT FRAME

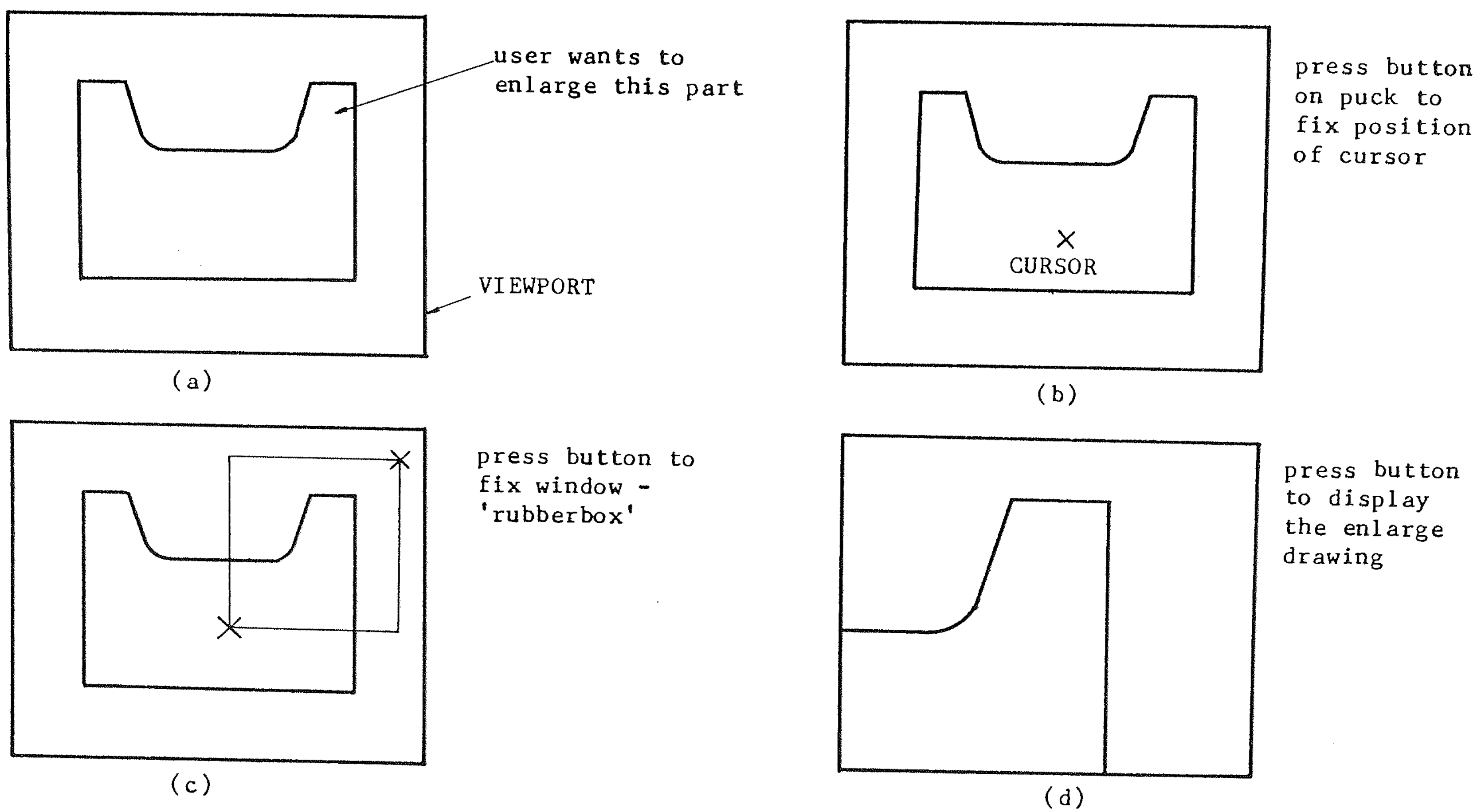


Fig. 6.16 ENLARGE THE DRAWING ON THE PERQ SCREEN

E1470b-2T

1	1	0.000	0.000	0.000	63.668	1.000	0.000	0.000	0
2	1	0.000	63.668	2.000	65.668	-1.000	1.000	-63.668	0
3	1	2.000	65.668	13.000	65.668	0.000	1.000	-65.668	0
4	1	13.000	65.668	13.000	71.668	1.000	0.000	-13.000	0
5	2	12.999	71.668	15.000	73.668	15.000	71.668	2.000	-1
6	1	15.000	73.668	23.899	73.668	0.000	1.000	-73.668	0
7	2	23.899	73.668	26.718	71.693	23.899	70.668	3.000	-1
8	1	26.718	71.694	29.232	64.788	2.748	1.000	-145.111	0
9	2	29.232	64.788	30.048	63.972	30.515	65.256	1.366	1
10	1	30.048	63.972	31.615	63.296	0.432	1.000	-76.954	0
11	2	31.615	63.296	32.099	62.987	31.161	62.049	1.326	-1
12	1	32.099	62.987	32.529	62.713	0.637	1.000	-83.441	0
13	2	32.529	62.713	33.084	61.841	30.840	61.025	2.388	-1
14	1	33.084	61.841	33.494	60.713	2.745	1.000	-152.641	0
15	1	33.494	60.713	34.119	58.999	2.744	1.000	-152.621	0
16	1	34.119	58.999	34.786	57.167	2.748	1.000	-152.762	0
17	2	34.786	57.167	35.682	56.188	36.415	57.760	1.734	1
18	1	35.682	56.188	35.924	56.075	0.467	1.000	-72.850	0
19	2	35.924	56.075	36.770	55.888	36.769	57.889	2.001	1
20	1	36.770	55.888	122.533	55.888	0.000	1.000	-55.888	0
21	2	122.533	55.888	123.379	56.075	122.533	57.889	2.001	1
22	1	123.379	56.075	123.621	56.188	-0.467	1.000	1.538	0
23	2	123.621	56.188	124.517	57.167	122.888	57.760	1.734	1
24	1	124.517	57.167	125.808	60.714	-2.745	1.000	284.648	0
25	1	125.808	60.714	126.219	61.841	-2.746	1.000	284.801	0
26	2	126.219	61.841	126.774	62.713	128.463	61.025	2.388	-1
27	1	126.774	62.713	127.204	62.987	-0.637	1.000	18.070	0
28	2	127.204	62.987	127.688	63.296	128.141	62.049	1.326	-1
29	1	127.688	63.296	129.254	63.972	-0.431	1.000	-8.258	0
30	2	129.254	63.972	130.071	64.788	128.787	65.256	1.366	1
31	1	130.071	64.788	132.587	71.694	-2.746	1.000	292.383	0
32	2	132.586	71.695	135.406	73.668	135.406	70.668	3.000	-1
33	1	135.406	73.668	143.303	73.668	0.000	1.000	-73.668	0
34	2	143.303	73.668	146.303	70.668	143.303	70.668	3.000	-1
35	1	146.303	70.668	146.302	65.668	1.000	0.000	-146.303	0
36	1	146.301	65.668	157.303	65.668	0.000	1.000	-65.668	0
37	1	157.303	65.668	159.303	63.668	1.000	1.000	-222.971	0
38	1	159.303	63.668	159.303	0.000	1.000	0.000	-159.303	0
0	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0

Fig. 6.17 PRINTER LISTING OF A ROLL PROFILE DATA FILE ((a) TOP ROLL)

Fig. 6.18 A ROLL DRAWING (a) TOP ROLL

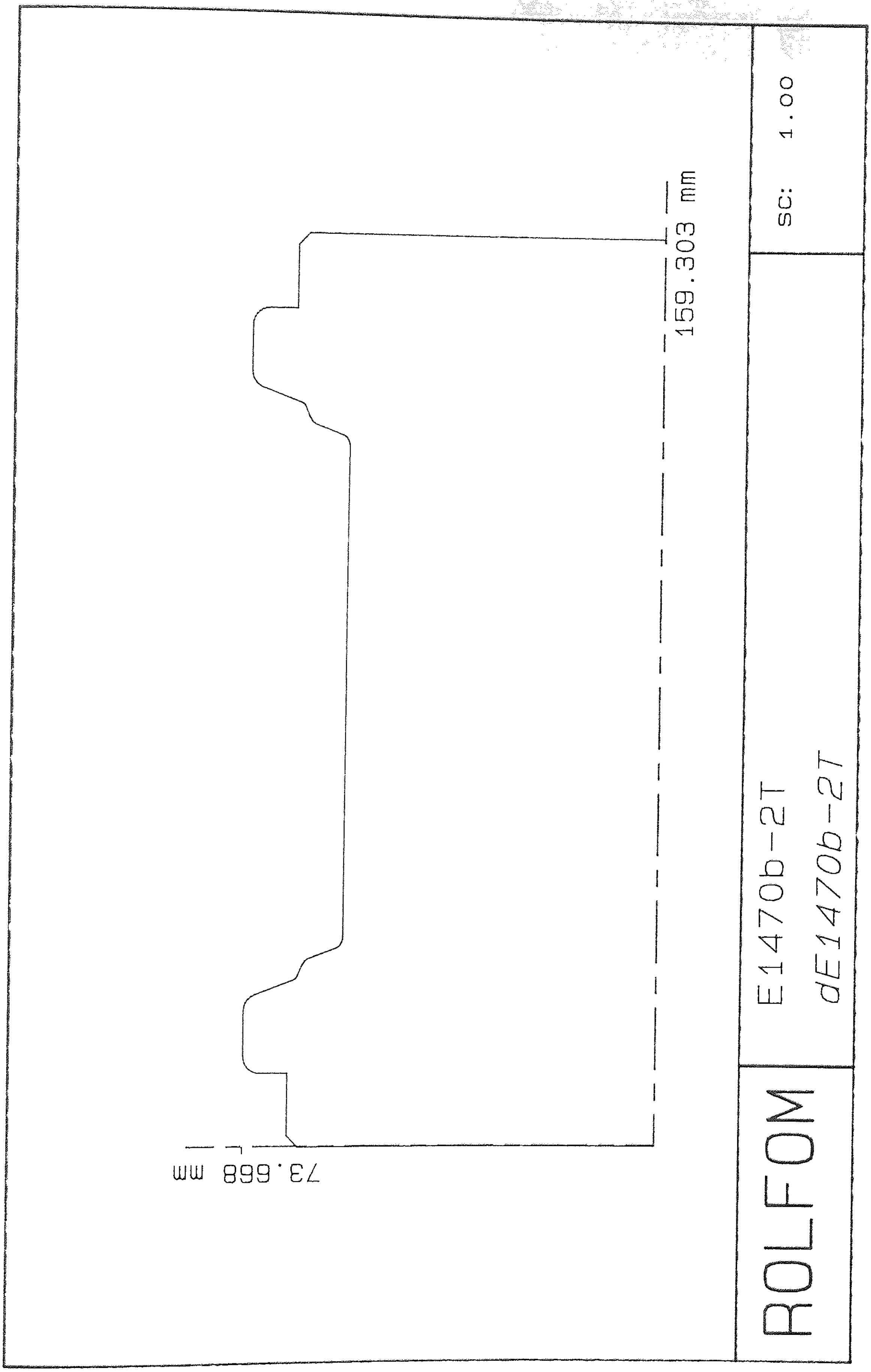
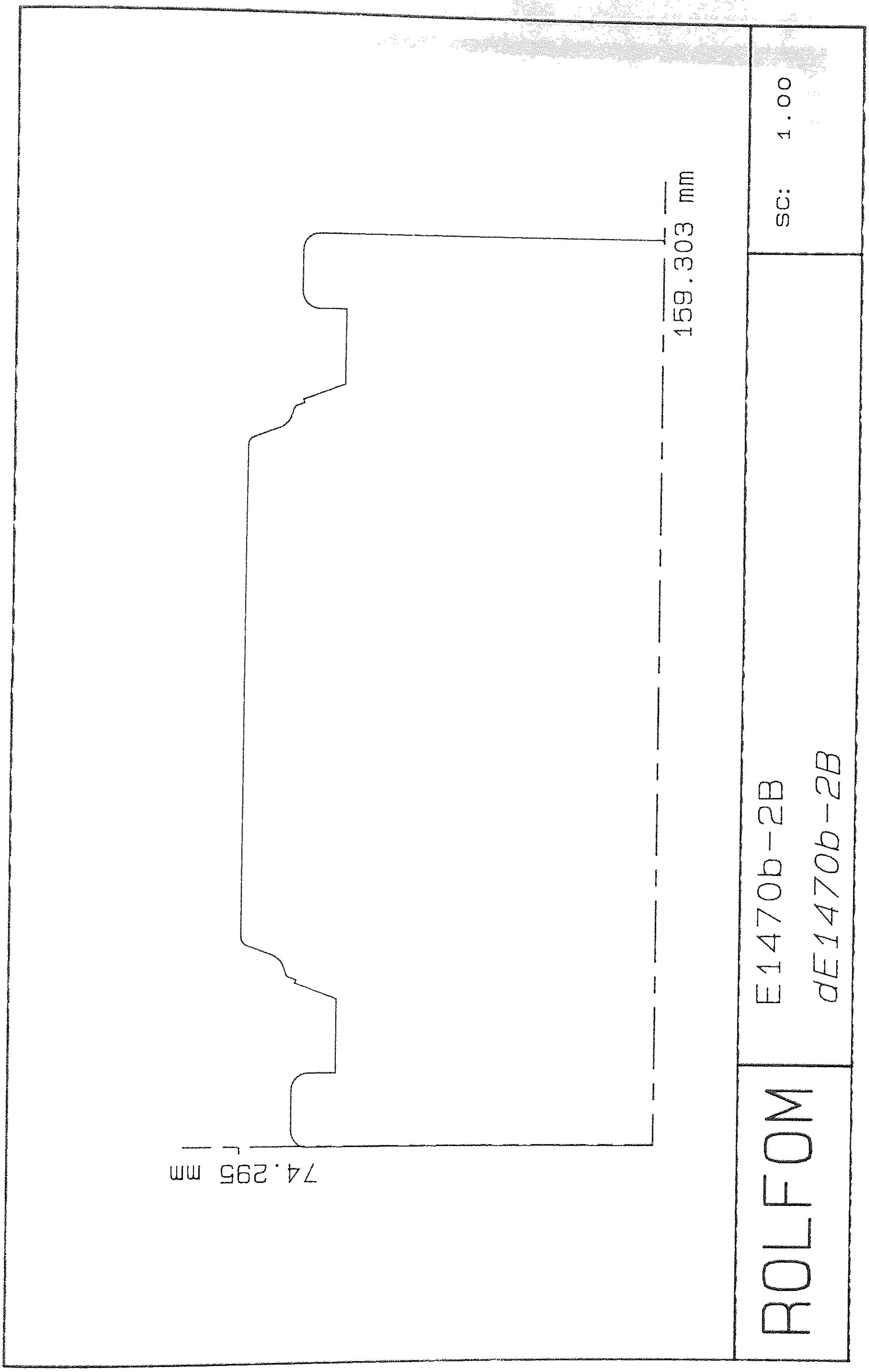


Fig. 6.18 A ROLL DRAWING (b) BOTTOM ROLL



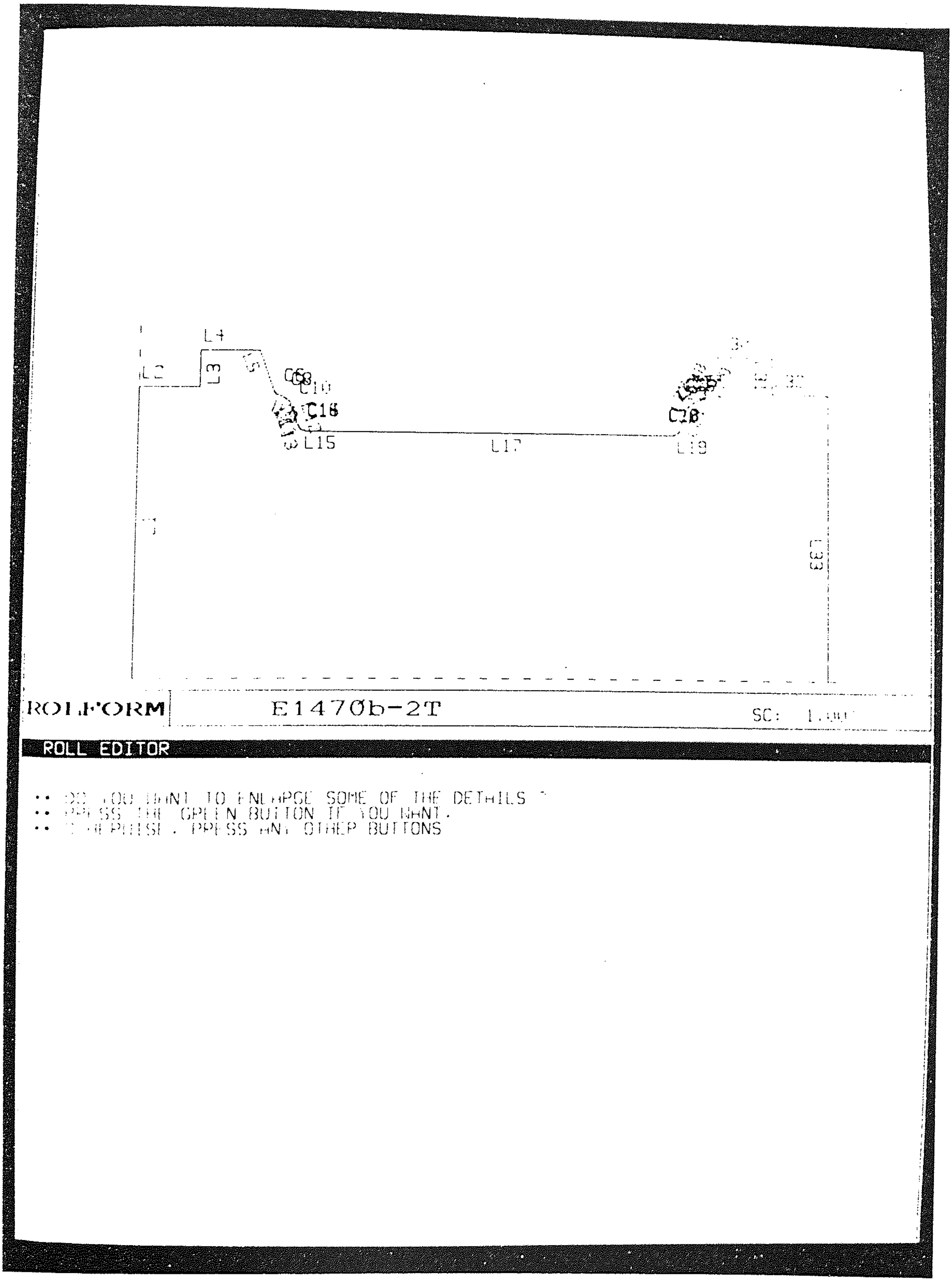


Plate 6.1 PERQ SCREEN DISPLAY OF A ROLL PROFILE DRAWING GENERATED BY THE ROLL EDITOR PROGRAM

Table 6.1 ROLL PROFILE DATA STRUCTURE

DATA SUBSTORE	SUBSCRIPTS	DESCRIPTION AND VALUES
NEL(n) (Integer)	n	Element sequence number, varies from 1 to 100
ITYPE(n) (Integer)	n	Element type, 1 for linear, and 2 for circular
RXA(n,5) (Real)	n,1	X-coordinate of starting point
	n,2	Y-coordinate of starting point
	n,3	The coefficient 'A' for linear element, or X-coordinate for centre point of circular element
	n,4	The coefficient 'B' for linear element, or Y-coordinate for centre point of circular element
	n,5	The coefficient 'C' for linear element, or radius 'R' for circular element
IDIR(n) (Integer)	n	Direction of arc, 1 for CCW, and -1 for CW arc

Notes :- (1) n is the element number from 1 to 100

(2) equation of linear element is $AX + BY + C = 0$

(3) all dimensions are in mm

Table 6.2 SUBROUTINES OF THE ROEDIT PROGRAM (PART 1)

SUBROUTINE	LEVEL	FUNCTION
ROEDIT	1	Calls upon subordinate subroutines to decode roll geometry data from the output of the Roll Design Program, to select the editing functions, to display the roll profile on the PERQ screen, and to generate the file containing the driving codes for drawing the roll profile with the HP7221c plotter
RINPUT	2	Decodes the roll contour data from the output of the Roll Design Program
PLOTGR	2	(see the PLOTGR module)
WINDOW	2	Calls up the GRAFIKS routines to set up a window by the rubberbox action, and to enlarge the window on the viewport
INSERT	2	Inserts a new element after a specified element, and updates the data structure
REPLAC	2	Replaces an existing element by a new element, and updates the data structure
CORNER	2	Modifies a sharp corner by inserting a chamfer or a blending arc, and updates the data structure
DELETE	2	Deletes a current element and updates the data structure
ROLDWG	2	Calls up facilities to plot the edited roll profile drawing by the HP7221c plotter
RINIT1	3	Initializes the roller data file
RINIT2	3	Initializes the current roll data structure
FRAME	3	Plots the viewport frame on the PERQ screen
ROPLOT	3	(see the PLOTGR module)

Table 6.2 (PART 2)

SUBROUTINE	LEVEL	FUNCTION
DEFEL	3	Calls up appropriate subroutines to define a new element, either linear or circular
DLN	4	Defines a linear element
DCIR	4	Defines a circular element
LNLN1	4	Determines the intersection point between two linear elements
LNCIR1	4	Determines the intersection point between a linear element and a circular element
CACIN	4	Determines the intersection point between two circular elements
DSIGN	5	Determines the identification sign for the relative position between a new line and a a given line

Table 6.3 SUBROUTINES OF THE PLOTGR MODULE

SUBROUTINE	LEVEL	FUNCTION
PLOTGR	1	Calls upon subordinate subroutines to set up the viewport and plot the roll profile on the PERQ screen
ROPLOT	2	Plots the roll profile in the viewport
CLIPCR	3	Plots a circular arc with starting and ending points and given centre point, at the the given direction
LNNO	3	Outputs the element number for linear element
ARCNO	4	Outputs the element number for circular element

Chart 6.1 HIERARCHY OF THE ROEDIT PROGRAM (PART 1)

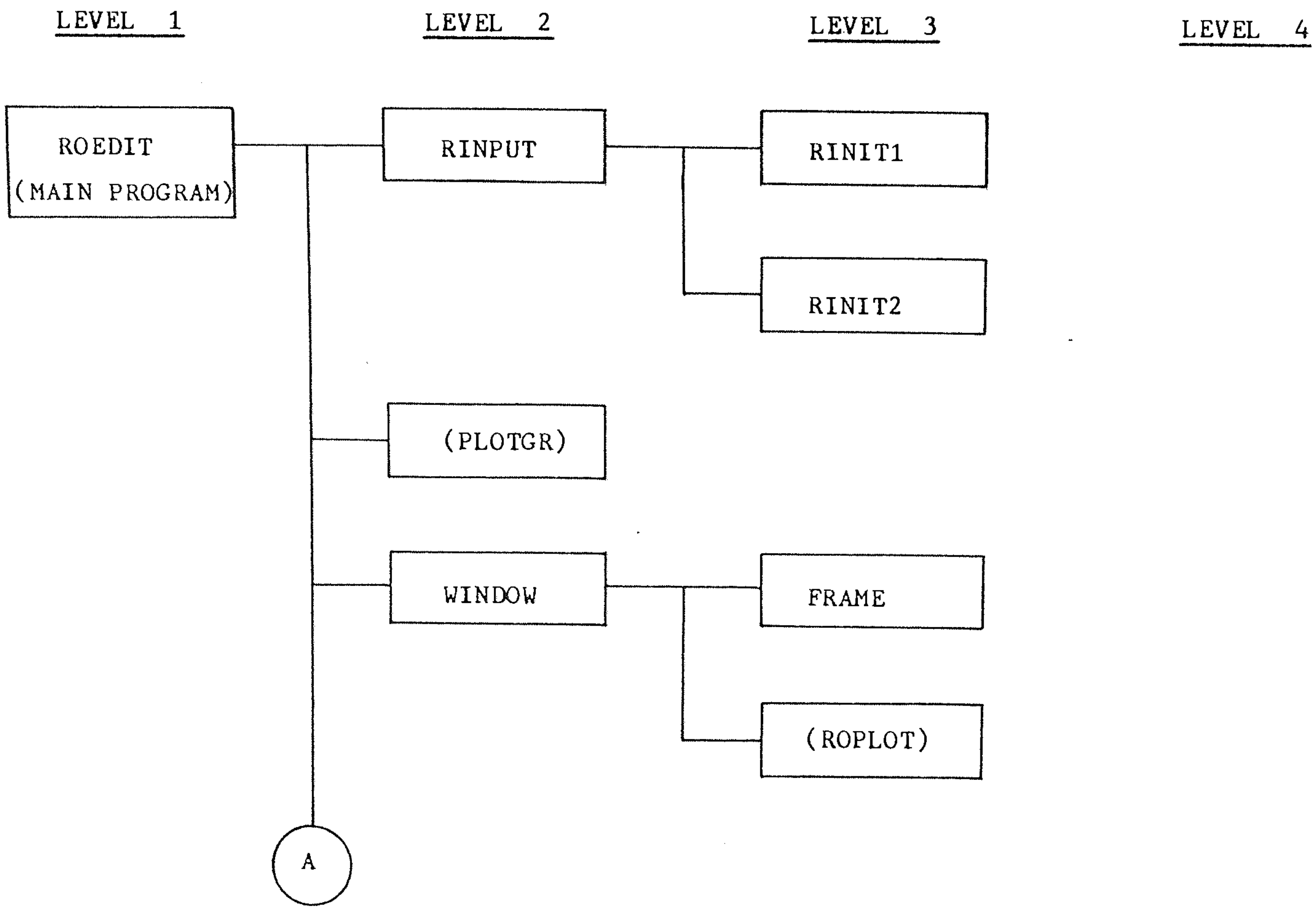


Chart 6.1 (PART 2)

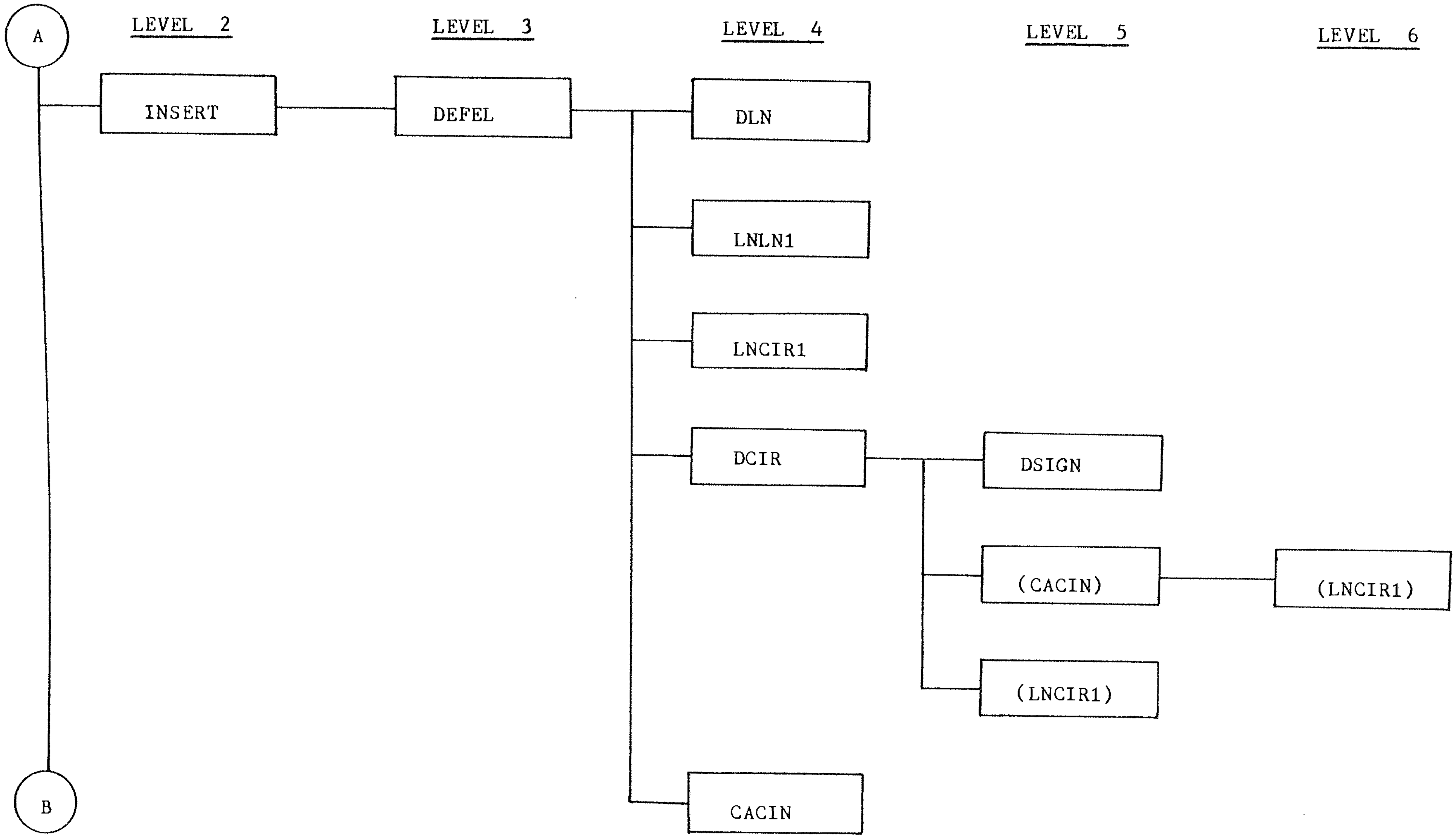


Chart 6.1 (PART 3)

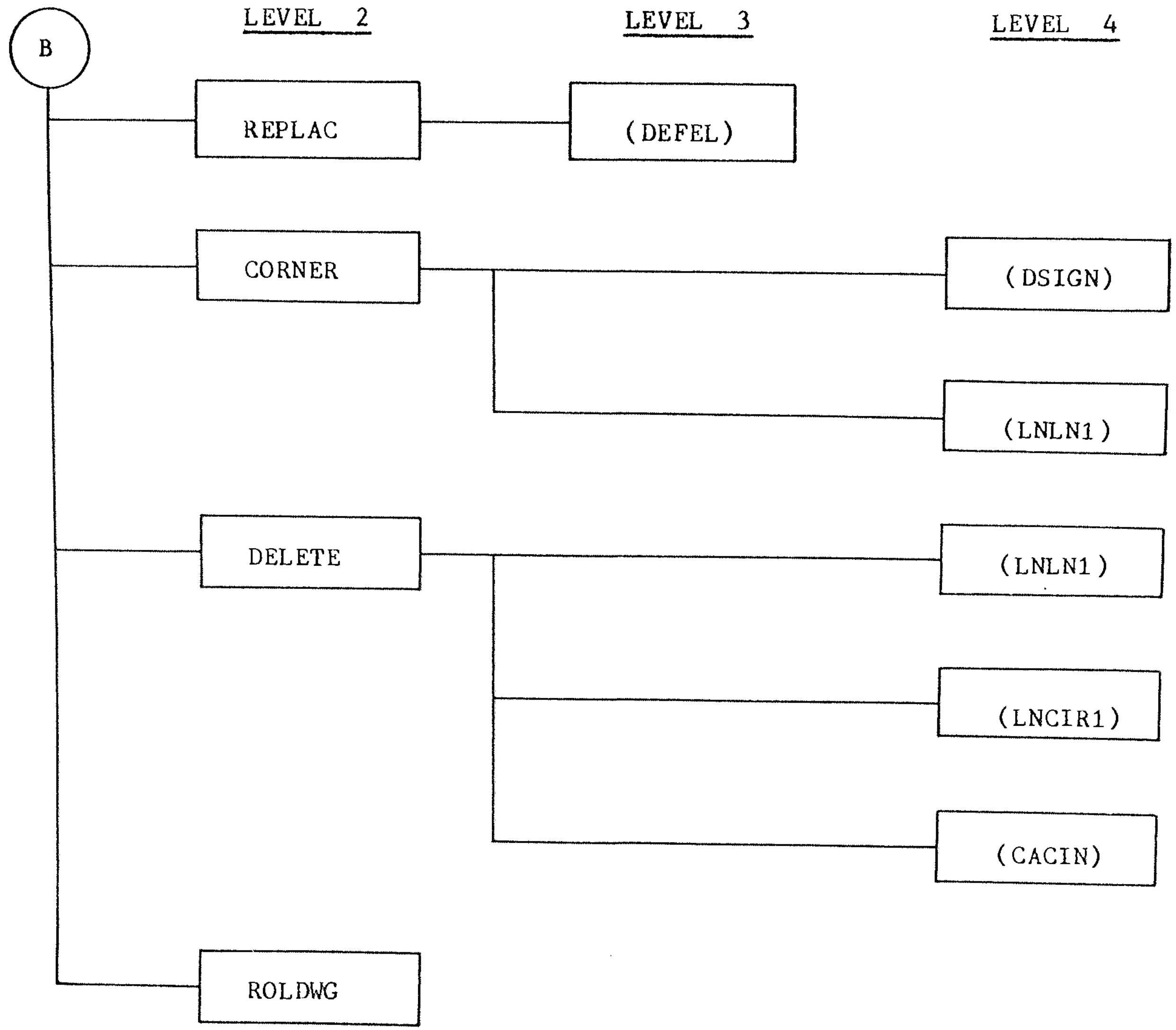
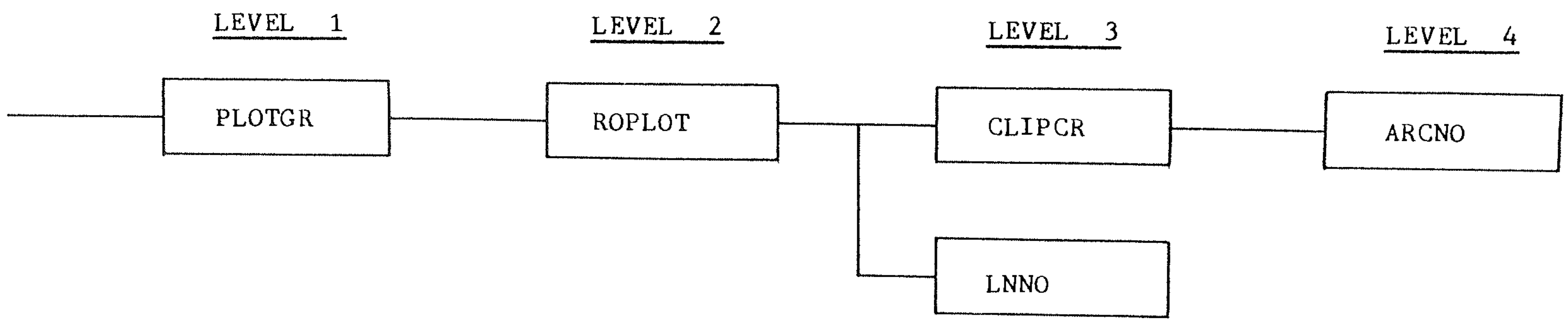


Chart 6.2 HIERARCHY OF THE PLOTGR MODULE



CHAPTER 7

CHAPTER 7

PRODUCTION OF FORM-ROLLS BY CNC

7.1 INTRODUCTION

As mentioned before, for the traditional method of turning form-rolls, it is necessary to make by hand full scale wire-templates. These templates can then be used as a guide to the machinist, as well as a measuring instrument, during the form-roll turning process. This conventional manual method for producing form-rolls, not surprisingly, relies greatly upon the skill of the operator. In particular, for large and complex roll profiles, working on a conventional lathe is very time-consuming and demanding high operator skill. It was thus one of the objectives of this research to enhance form-roll production through the implementation of NC turning.

7.2 NC FOR ROLL MANUFACTURE

Due to the fact that different sets of rolls bearing different roll profiles, the turning of form-rolls is actually a one-off job. Initially it may not be economically justified to apply NC turning as it requires a great deal of time to prepare a manual part-program and then to prove the punched control tape. In fact, sometimes it is even quicker to cut on the conventional lathe than to prepare the manual program for an NC lathe. Indeed, if it is required to prepare the NC part-program manually, with the only available information being the roll design drawing, it will be a very tedious job as it is required to digitise the complicated roll profile. It is, therefore, not deemed to be economically justified to simply adopt NC turning for producing form-rolls, while lacking any facility - such as computer aids - to facilitate the NC part-program preparation process.

In this research, however, NC turning had been implemented, according to the development programme that the complete roll design and manufacture process would eventually be computerised and integrated together. To begin with, a Fanuc controlled Mori-Seki CNC lathe (Plate 4.6) has been acquired and installed in the sponsored company. Investigations had then been carried out for establishing the standard procedures for NC form-roll turning, that is, to decide on the work holding method and select cutting tools on the one hand, and the NC tape preparation on the other.

7.3 TOOLING AND WORKPIECE SETUP

The accuracy of NC machining can easily be affected by the tooling and/or workpiece setup. Hence it has been decided that, in order to obtain an accurate roll profile, the turning process should be done in such a way as to produce a complete roll profile in one single workpiece setup on the lathe. In order to achieve this, a set of mandrels, to cope with the range of different sizes of rolls, has been purposely designed and produced for holding workpieces during NC turning. In the actual NC turning process, therefore, blanks should be pre-bored and accurately faced to length. The diameters of the bore should be in accordance with the diameters of the spindles in the workstations of the cold roll forming machine. The blank is then held on the mandrel and supported by the tailstock. It can, thus, be possible to cut the roll from both ends, which means to allow turning both towards and away from the headstock. Fig. 7.1 depicts the design of a mandrel. A typical roll being held by a mandrel in the Fanuc controlled Mori Seki CNC lathe is shown in Plate 7.1.

In essence, turning of form-rolls is basically divided into two sections, roughing out and finish profiling. A set of carbide tip turning tools has then been chosen as standard cutting tools for the form-roll turning process. This set of standard cutting tools has been deployed into permanent positions in the 12-station turret of the CNC lathe (Plate 7.2). The setting up position (Fig. 7.2) of each individual cutting tool, with reference to the origin of the roll profile, can therefore be kept as a standard. Left-hand and right-hand turning tools are usually employed for the roughing out (area clearance) of the roll profile. For some intricate concave profiles, however, ordinary left-hand and right-hand cutting cannot be applied, a parting-off tool has to be used to cut away concave areas by grooving. A 4.1 mm wide and a 6 mm wide carbide-tipped parting tool have been set in the turret for this purpose. On the other hand, both left-hand and right-hand fine turning tools, together with a button tool have been equipped for the finishing cuts in the roll turning process. Table 7.1 gives the list of the cutting tools set in the turret.

7.4 PROGRAMMING THE MORI-FANUC CNC LATHE

Unlike conventional machining methods, NC machining demands a greater effort in preparing a more detail process plan. In NC programming, the part programmer should determine tooling requirements, the sequence of cutting operations, the feed and speed technology and also the slide way movements required to produce the finished part. Fig. 7.3 shows the conventional tape preparation process.

The part-programming process starts with extracting co-ordinate information from the component drawing and written out in a process layout sheet with cutting conditions (e.g. rpm, cutting depth, feed, etc.), to regulate the operation of the machine tool. With regard to form-rolls, it is a very tedious job to digitise the co-ordinate information from a roll design drawing, mainly due to the complexity of the roll profile. This drawback has been remedied by the implementation of computer aids in this research. As mentioned in Chapter 6, it is possible to obtain from the roll editor program, a roll profile data file containing the co-ordinates for the computerised points on the roll profile drawing. An NC part-program can then be prepared, with reference to this roll profile data file for the roll geometry description. Fig. 7.4 illustrates the procedure for programming the CNC lathe for turning form-rolls.

7.4.1 Co-ordinate System

In an NC machine tool part-program, the desired movements of the machine tool slides should be given in accordance with the co-ordinate system of the particular NC machine tool. Generally, NC lathes have two slideways, and the direction in which the carriage moves is the Z-axis, whereas the X-axis is the direction in which the cross slide moves. The convention for the co-ordinate system of the Mori-Seki CNC lathe is given in Fig. 7.5.

7.4.2 Program Pattern

In the NC part-program, details of machining operations have to be set down in sequence. Each line of the program should be numbered in sequence, details of the operation have to be stated and the X and Z co-ordinates should be given. This basic information should then be supplemented by code numbers respresenting the preparatory functions; feed rate; spindle speed; required tool; and miscellaneous functions. Nevertheless, the format or pattern for an NC part-program may vary according to the system used by the manufacturer of the NC equipments. Details for the part-program format for the Fanuc controlled Mori Seki lathe can be referred to the programming manual [58] and hence are not mentioned here. However, the program pattern and a list of G-codes (for preparatory functions) and M-codes (for miscellaneous functions) are given in Appendix 3.

7.5 APPLICATION OF COMPUTER-AIDED PART-PROGRAMMING SYSTEMS

Despite the help from the roll design drawings and the digital roll design data file output from the roll editor program in connection with the CAD roll design software, it is still a very time-consuming task to manually prepare an NC control tape for the intricate roll profile. In particular, the part-programmer has to be familiar with the program format, such as the G-codes and M-codes, for the machine control tape program.

Although it is the objective of the complete CAD/CAM development programme to establish a special-purpose NC lathe programming system to link the CAD software with the CNC lathe for manufacturing. In the interim, however, before the implementation of the special-purpose NC lathe programming system, a general purpose turning processor had been used. The general purpose turning processor adopted is FAPT [59] which was supplied by Fanuc with the Fanuc System P microcomputer (Plate 7.3). FAPT is an APT like part-programming system, whereby simple means are available for defining the part geometry, machining processes and specifying tool requirements. From the FAPT part-program, the computer can then generate, with the Fanuc system processor, the machine tool control tape program for cutting the required component. With the FAPT post-processor, the control tape program will be generated according to the program format for the specified NC machine tool, for which in this case is the Mori-Fanuc CNC lathe. Unlike manual programming, the part-programmer is not required to specify the G-codes and M-codes, which will be determined by the post-processor automatically in the program. During the preparation of a FAPT part-program for machining a form-roll, the roll profile data output from the editor program has to be used, together with the roll design drawing, to obtain information for the geometric data statements of the FAPT program. In fact, a FORTRAN program POSTFANUC had been written for converting the roll profile data file into the geometric data input format for the FAPT system. The listing of POSTFANUC is given in Appendix 4.

Although the APT like FAPT part-programming system can be quite easily handled by the part-programmer, it is yet lacking in the integrity and simplicity provided by the tailored special-purpose processor and post-processor system, which will be discussed in detail in the next chapter. Fig. 7.6 outlines the procedure for preparing an NC tape for machining form-rolls with the assistance of the FAPT system.

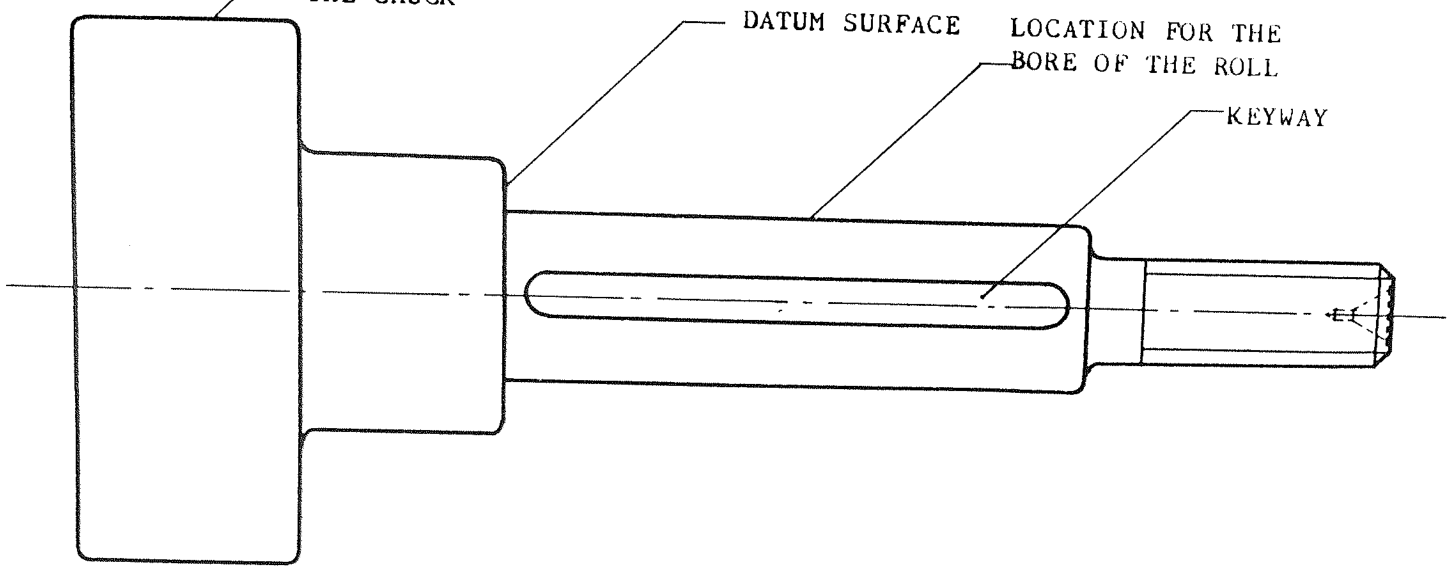


Fig. 7.1 DESIGN OF THE MANDREL

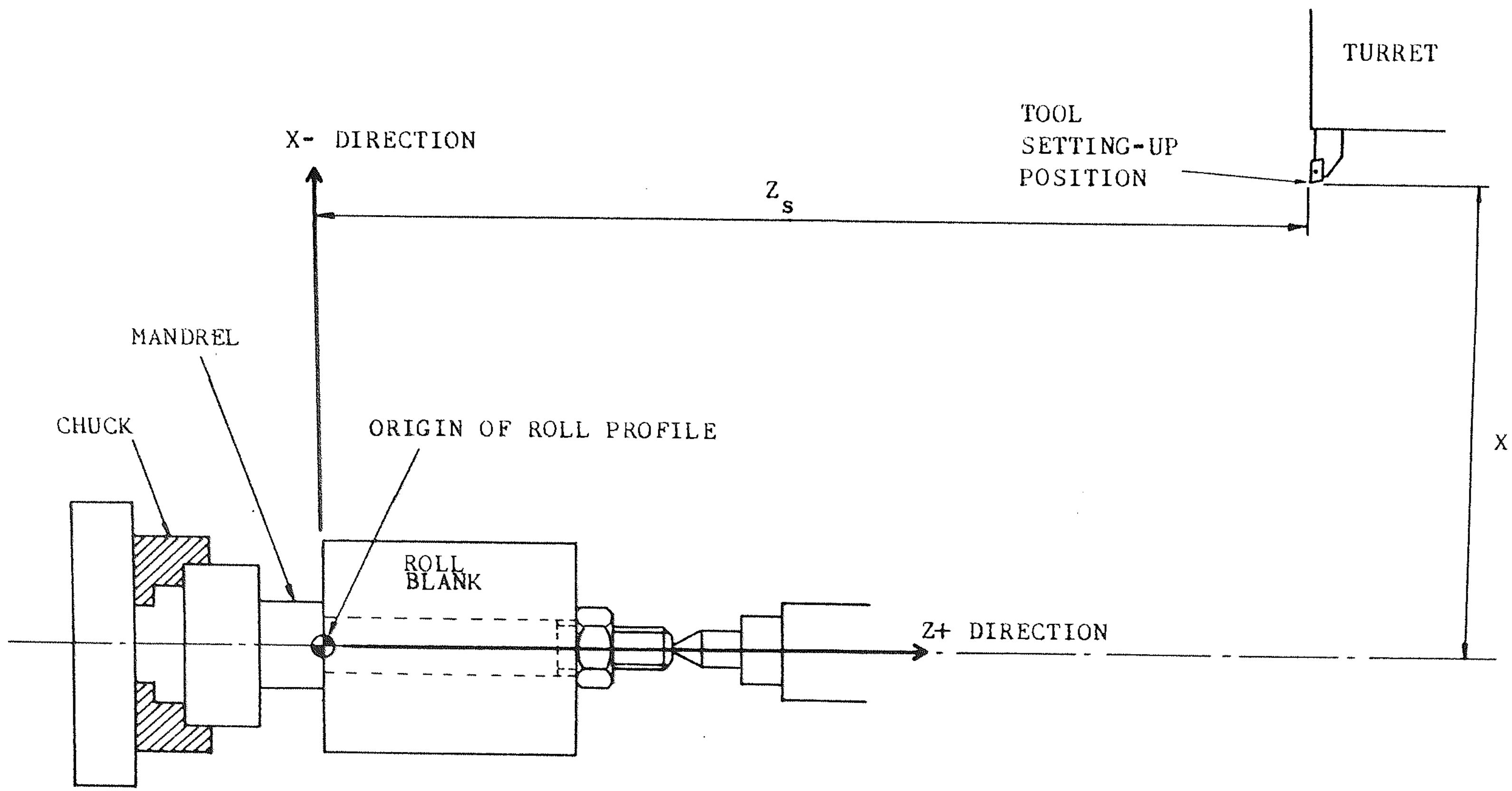


Fig. 7.2 SETTING UP POSITION OF CUTTING TOOL

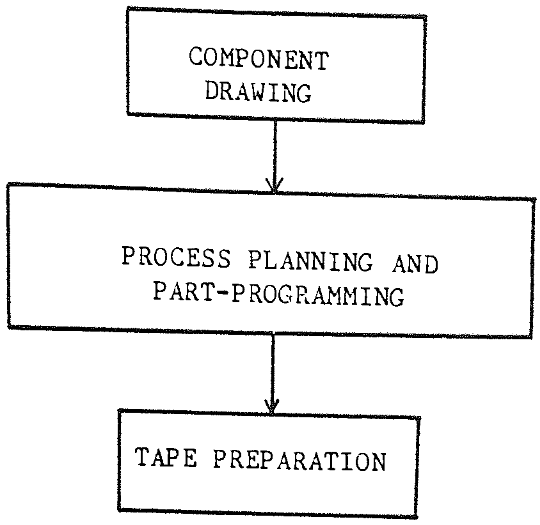


Fig. 7.3 CONVENTIONAL TAPE PREPARATION PROCESS

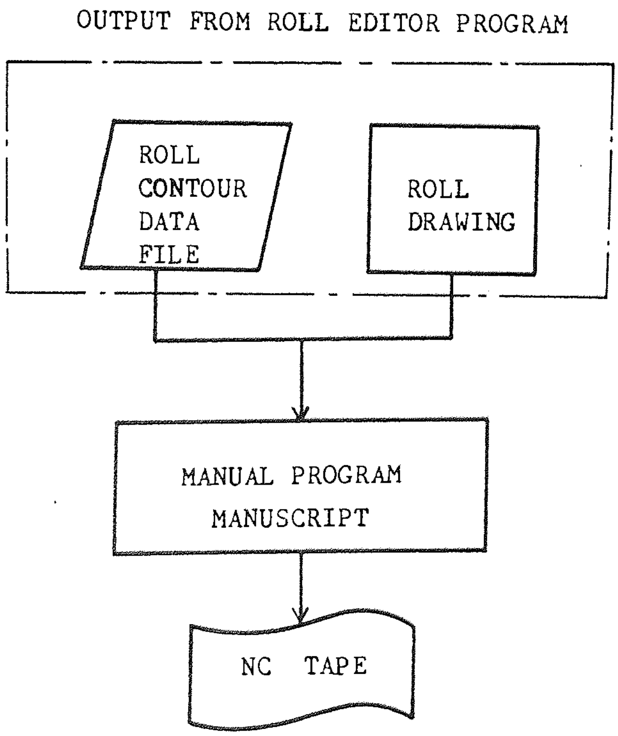


Fig. 7.4 PROCEDURE FOR MANUALLY PROGRAMMING THE CNC LATHE FOR TURNING FORM-ROLLS

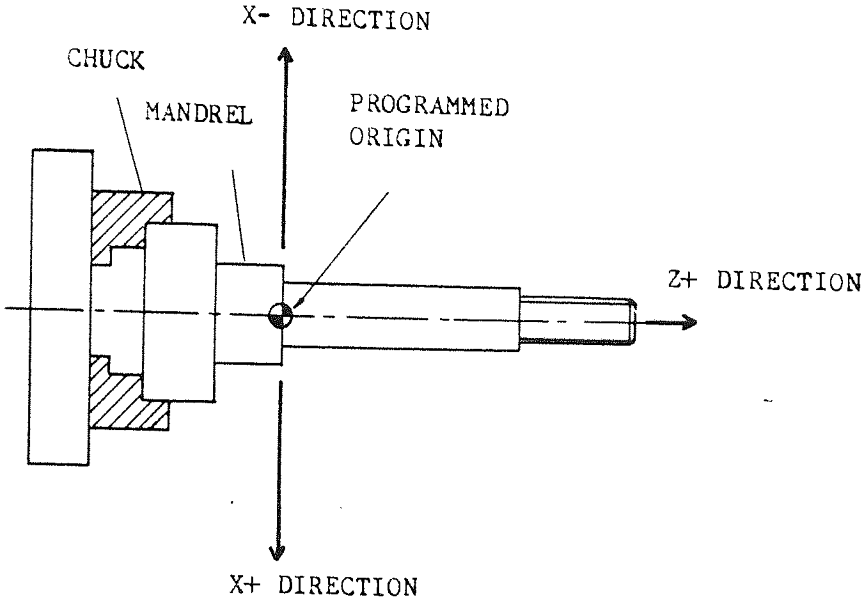


Fig. 7.5 COORDINATE SYSTEM FOR THE CNC LATHE

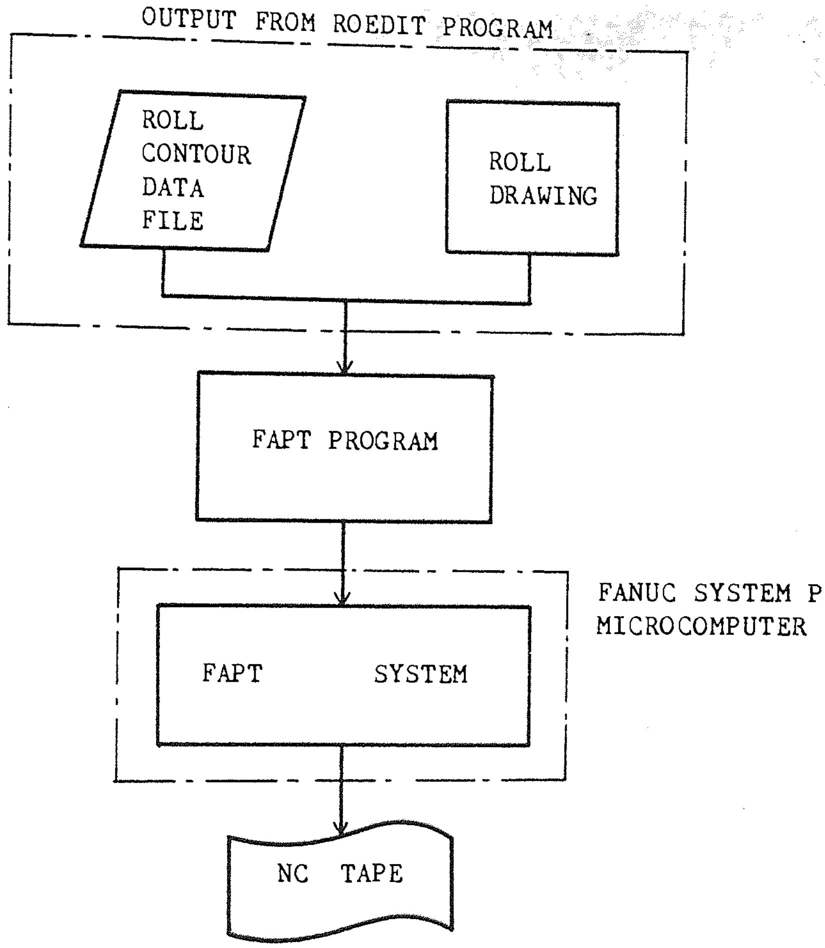


Fig. 7.6 PROCEDURE FOR PREPARING NC TAPE WITH FAPT SYSTEM

Plate 7.1 WORKPIECE SETUP USING A MANDREL

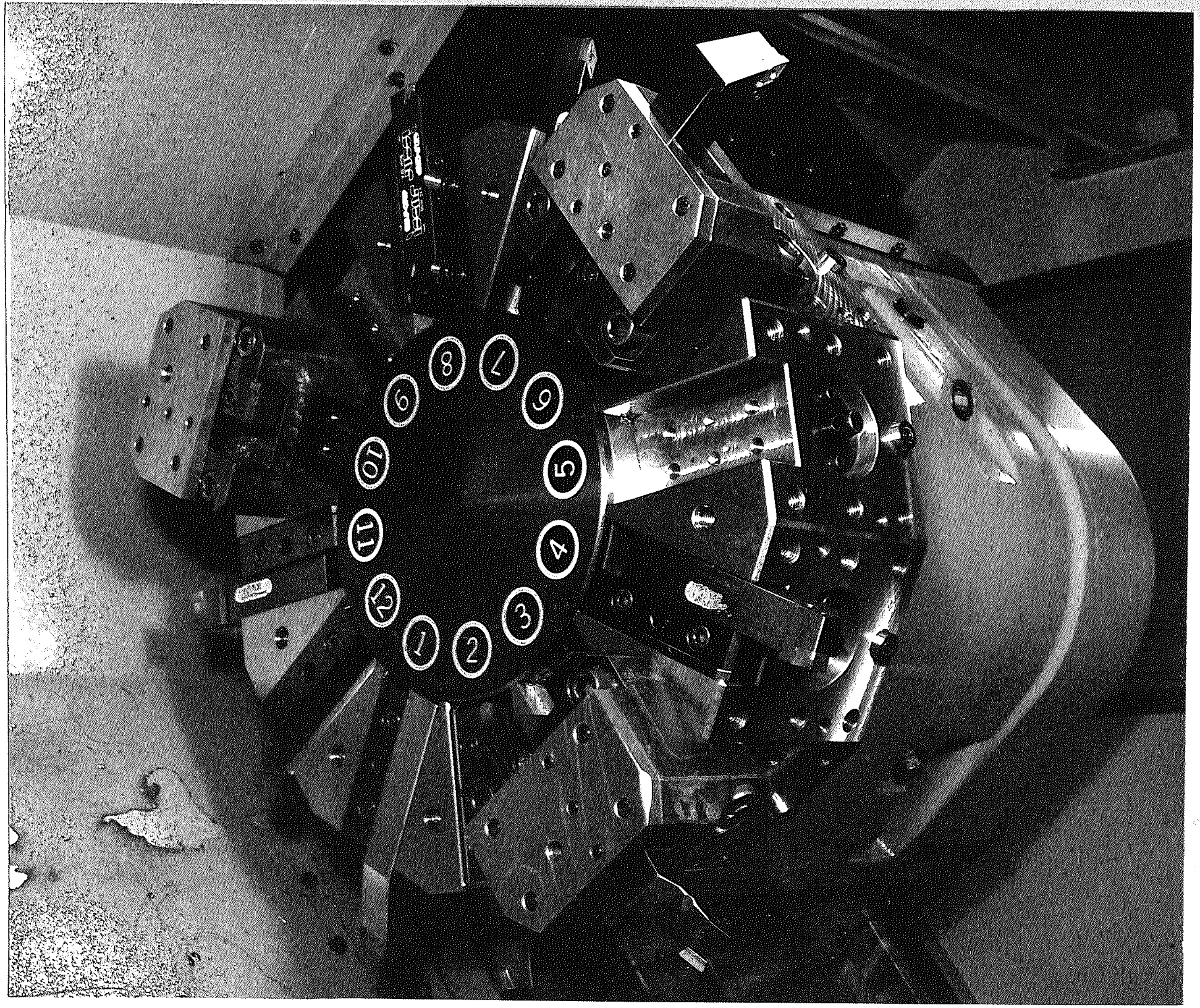


Plate 7.2 TOOLING SETUP IN THE TURRET OF THE CNC LATHE

Plate 7.3 FANUC SYSTEM P MICROCOMPUTER FOR FAPT PROGRAMMING

Table 7.1 CUTTING TOOL LIST OF THE CNC LATHE

TURRET POSITION	TOOL
1	5 mm Button Tool for Finishing
2	55^{o} Romboid RH Roughing Tool
3	35^{o} Diamond LH Finishing Tool
4	Not In Use
5	6 mm RH Parting-off Tool
6	55^{o} Romboid LH Roughing Tool
7	35^{o} Diamond RH Roughing Tool
8	4.1 mm RH Parting-off Tool
9	Not In Use
10	35^{o} Diamond LH Roughing Tool
11	35^{o} Diamond RH Finishing Tool
12	70^{o} Copying Tool

CHAPTER 8

CHAPTER 8

THE SPECIAL-PURPOSE ROLL MACHINING SOFTWARE AND POST-PROCESSOR

8.1 INTRODUCTION

The efficiency and economic use of NC machine tools depends mainly on the preparation of the NC control tapes. It has been discussed in the previous chapter that it had been found impractical, though not impossible, to manually prepare NC control programs and tapes for machining form-rolls, which are generally complicated in shape, but small in quantity - literally one-off. The introduction of the FAPT programming systems, like APT, EXAPT and COMPACT II, requires the part-programmer to conform to its strict program syntax. Therefore, the FAPT system was not considered as a permanent solution for integrating the CAD and NC systems in this research.

In order to complete and integrate the CAD/CAM development programme concerned in this research, a special-purpose NC lathe part-programming system has been designed and developed to link the CAD software with the CNC lathe for manufacturing. The special-purpose NC lathe programming system in this case, is a combination of two computer programs, which were written in FORTRAN and have been implemented in the PERQ system. The first program, CUTPLOT, is a processor which considers the geometry of the rolls to be turned and interprets the cutting instructions, consequently, the cutter location information - known as CLDATA, will be generated. A plot of the tool-paths will be displayed on the PERQ screen, or a hardcopy drawing can also be produced by the HP7221c plotter, so that the part-programmer can verify the operating sequence and can also check for any possible tool-material collision. The second program, MFPOST, on the other hand, is the special-purpose post-processor which has been written to convert the CLDATA file to a control tape for the FANUC controlled Mori-Seki CNC lathe. This chapter is to give a description of the CUTPLOT software, while the MFPOST post-processor will also be discussed.

8.2 GENERAL ASPECTS OF THE CUTPLOT SOFTWARE

The CUTPLOT software, like the other NC programming systems, requires two sets of data input. The first being the geometric description of the component to be machined, while the second set of input data required is the technological data for the machining process. However, the CUTPLOT software is designed basically to simplify the part-programmer's task, and unlike FAPT or the other APT like part-programming systems, the operation of the CUTPLOT software does not require the part-programmer to use the cumbersome alphanumeric syntax and vocabulary for data input. On the contrary, this CUTPLOT software only requires, through an interactive dialogue via the PERQ screen and the keyboard, some simple numerical codes for data input. In fact, the aim of the CUTPLOT software is to generate automatically, with minimum user intervention, the CLDATA file for turning form-rolls.

8.3 THE TOOL LIBRARY

In Section 7.3, it has been mentioned that a set of carbide-tipped turning tools had been selected for the form-roll turning purpose, and the list of these tools has been given in Table 7.1. In fact, this list of turning tools, together with the tool definitions, has been stored in the CUTPLOT software as a tool library module TLIB. The part-programmer, hence, is only required to specify the number of the turret position, and the corresponding tool definitions will be retrieved automatically from TLIB. Of course, an option has also been provided for choosing to use a turning tool which is not in the tool library TLIB. The part-programmer, in that case, has to input the proposed turret position and the tool definitions accordingly.

In the tool library TLIB, the geometry of a tool is defined by its tip nose radius r and the virtual tip nose circle centre co-ordinates (z, x) in the tool co-ordinate system (Fig. 8.1). For a parting-off tool, the tool nose width w is also specified (Fig. 8.2). In addition, the distances (z_s, x_s) in both Z and X directions from the component origin to the pre-set starting point (Fig. 8.3) of each individual cutting tool is also encoded. Table 8.1 gives a complete set of data stored in the tool library TLIB.

8.4 GEOMETRIC DATA

As mentioned earlier, the CUTPLOT software has been designed to capture the geometric profile data from the output data file generated by the computer-aided roll design process. In fact, a roll profile data file had been generated for each individual roll by the roll editor program. Hence, during the operation of the CUTPLOT software for the preparation of an NC part-program for machining a form-roll, the part-programmer has to specify the part number of the thin-section, together with the stage number and the roller number, the particular roll profile data file will then be retrieved by the system and the geometric data can thus be decoded. The canonical data of the geometric elements of the roll will be stored in the memory storage. Again, the roll profile comprises of either linear or circular geometric elements, and as many as 100 elements can be stored in the memory. The design of the geometric data store for this CUTPLOT software is in fact the same as that for the roll editor program, which has already been explained in detail in Section 6.3, and hence is not described here.

8.4.1 Determination of Blank Size

According to the dimensions of the roll geometry profile, the size of the blank will be determined automatically by the CUTPLOT software. In the first place, the width of the blank is the maximum Z-dimension of the roll, while the diameter of the blank will be selected from a range of standard blank diameters. The standard blank diameters used by the company are 115, 130, 145, 160, 175, 190 and 210 mm. The software will first find out the largest finished roll diameter, then a 10 mm clearance will be added to obtain the minimum blank diameter required, from that, the nearest larger blank diameter from the standard set will be selected. However, if the final finished roll diameter plus 10 mm clearance is greater than 210 mm, the user (part-programmer) will be requested to provide the blank diameter manually.

8.4.2 Machining Area Definition

It is common in the turning process to divide the component into different manufacturing segments (or machining areas), and different cutting tools can then be used to machine the different segments, either for roughing out or finish profiling, with different machining conditions such as spindle speed and feedrate. In operating the CUTPLOT software, the machining area should be defined before commanding each of the chosen machining cycles. Concerning this application, the machining area is a series of contour elements on the roll profile segment to be machined and the shape of the material from which the part is to be removed. As the geometric data for the roll has already been kept in the memory storage, a machining area in the CUTPLOT system needs only to be defined by the designated element numbers of the starting and ending elements, and these element numbers should be obtained from the corresponding roll contour data file. The radius of the blank, and the vertical lines at the two profile terminal points form the boundary for the machining area (Fig. 8.4).

The direction of cut for a machining area will be determined automatically by the relative locations between the starting and the ending elements. If the starting element is on the right hand side of the ending element, cutting for that area should be from right to left. Conversely, cutting will be considered as from left to right if the starting element locates on the left hand side of the ending element. For example, referring to the roll profile as shown in Fig. 8.4, a machining area can be defined by designating element No. 2 as the starting element and element No. 10 as the ending element. Therefore the machining area will be enclosed by the roll profile areas from element No. 2 to 10, while the maximum blank diameter will be the blank contour. The cutting direction will be automatically assigned to be from left to right. On the other hand, if element No. 10 is chosen as the starting element while element No. 2 to be the ending element, the cutting direction in this case will be considered as from right to left.

8.5 CUTTER OFFSET

Turning form-rolls is actually a kind of contouring job, without any exception from other contour turning works on an NC lathe, the tool tip nose radius will affect the actual profile of the finished product. As tool bits usually have a nose radius, which changes the point of contact between the tool and the workpiece, that is, the command point is not the actual cutting point. Therefore, although cutting can be performed as commanded by outside diameter, inside diameter and face cutting programs, excess or insufficient cutting will occur in chamferring, tapering and circular cutting, as shown in Fig. 8.5. A solution to this problem is to take into consideration of the tool nose radius r, and the program should be prepared by commanding the cutter path for the centre point of the nose radius circle of the tool tip. In other words, the tool centre moves on a parallel contour, being offset from the original one by an amount equal to the tip nose radius r - the tool compensation.

Furthermore, with only the exception of the final finishing cut, finishing allowance or stock of material should be left by the machining cycle for finishing touch-up. Thus, as shown in Fig. 8.6, the finishing tolerance t and the tool nose radius r must be added together to obtain the total amount of offset OFF, that is:

$$OFF = t + r \tag{8.1}$$

In the CUTPLOT software, the subroutine OFFSET has been designed and developed to obtain the offset contour of the machining area in every machining cycle. Firstly, the finishing tolerance t has to be specified by the part-programmer, and the tool tip nose radius r for the chosen cutting tool will be picked up by the subroutine directly from the tool library TLIB in the memory storage. The geometry of the offset contour, which is at a distance OFF away from the original machining area, can then be obtained, as shown in Fig. 8.7. Therefore, the tool path of each machining cycle will be governed by the offset machining area, which is the check surface for the cutting stroke in the roughing, grooving and pocketing cycles, and, on the other hand, is the drive surface for all profiling cuts.

In order to determine the complete set of geometric data for the offset machining area contour, the OFFSET algorithm will first obtain the offset tool path for each of the elements of the original contour segment. Then these offset paths can be assembled together to form the continuous offset contour profile segment. The geometric data of the offset contour profile segment will be stored in the data structure which is similar to the one used for the original roll profile (see Table 6.1).

Just like the algorithms in the roll editor program, the OFFSET algorithm, for determining the offset contour of a given profile, has to utilize the geometric principles and procedures dealing with straight lines and circles. Thus several subordinate modules have been designed and developed for the complete OFFSET algorithm. However, the main approach of the OFFSET algorithm is, in essence, to determine the direction of cutting in the first place, then the offset tool paths for all contour elements can be determined, and finally the geometry of the offset contour can be established by joining all these offset tool paths together.

8.5.1 Sign Convention for Cutting Direction

In Section 8.4, it has been mentioned that the cutting direction will be determined automatically according to the relative position between the starting and the ending elements of the machining area. As a matter of fact, the position of the cutting tool, in relative to the roll contour, depends on the direction of cuts. When the cutting direction is from right to left, the cutter should be on the right hand side of the contour, and this is denoted in the OFFSET algorithm by introducing an indicator flag IK, which equals to -1 in this case (Fig. 8.8a). Conversely, as shown in Fig. 8.8b, IK equals to +1 if the tool is on the left hand side of the contour in the case when cutting starts from left to right.

8.5.2 Location of Cutter Relative to Circular Elements

In addition to the sign convention for IK for the cutter position, it is necessary to introduce another indicator JK for circular elements. It is determined that JK equals 1 if the cutter locates outside of the circle which contains the circular element, while JK equals -1 if the cutter locates inside. Depending on the direction of the circular arc (IDIR) and the direction of cut (IK), there are altogether four different cases which are shown in Fig. 8.9. A module SIGN has been developed for this purpose.

8.5.3 Offset Path of a Linear Element

After obtaining the magnitude of offset 'OFF' and the relative position of the cutting tool, being denoted by IK, it is then possible to translate a point on a linear element to its offset counterpart. As a linear element in the data storage is represented by its starting and ending points, the offset translation of these two end points can then be sufficient enough to represent the offset path of the linear element. Thus, a module STL has been established for computing the offset of given points on straight lines.

Assuming the starting point P and the ending point Q of the linear element are having co-ordinates (z_1, x_1) and (z_2, x_2) respectively. The direction of this linear element will be determined by the relative position of the two points P and Q, and there are altogether eight different cases to be considered, as depicted in Fig. 8.10. In order to obtain the co-ordinates for P' and Q', which are the offset for P and Q respectively, each of the eight cases should be considered separately, as shown in Fig. 8.11. However, the required calculations are as follows:

Case 1 $z_2 > z_1$ and $x_2 > x_1$

$$z_1' = z_1 - (IK * OFF * \sin(A)) \quad (8.2)$$

$$x_1' = x_1 + (IK * OFF * \cos(A)) \quad (8.3)$$

$$z_2' = z_2 - (IK * OFF * \sin(A)) \quad (8.4)$$

$$x_2' = x_2 + (IK * OFF * \cos(A)) \quad (8.5)$$

Case 2 $z_2 < z_1$ and $x_2 > x_1$

$$z_1' = z_1 - (IK * OFF * \sin(A)) \quad (8.6)$$

$$x_1' = x_1 - (IK * OFF * \cos(A)) \quad (8.7)$$

$$z_2' = z_2 - (IK * OFF * \sin(A)) \quad (8.8)$$

$$x_2' = x_2 - (IK * OFF * \cos(A)) \quad (8.9)$$

Case 3 $z_2 < z_1$ and $x_2 < x_1$

$$z_1' = z_1 + (IK * OFF * \sin (A)) \quad (8.10)$$

$$x_1' = x_1 - (IK * OFF * \cos (A)) \quad (8.11)$$

$$z_2' = z_2 + (IK * OFF * \sin (A)) \quad (8.12)$$

$$x_2' = x_2 - (IK * OFF * \cos (A)) \quad (8.13)$$

Case 4 $z_2 > z_1$ and $x_2 < x_1$

$$z_1' = z_1 + (IK * OFF * \sin (A)) \quad (8.14)$$

$$x_1' = x_1 + (IK * OFF * \cos (A)) \quad (8.15)$$

$$z_2' = z_2 + (IK * OFF * \sin (A)) \quad (8.16)$$

$$x_2' = x_2 + (IK * OFF * \cos (A)) \quad (8.17)$$

Case 5 $z_2 = z_1$ and $x_2 > x_1$

$$z_1' = z_1 - (IK * OFF) \quad (8.18)$$

$$x_1' = x_1 \quad (8.19)$$

$$z_2' = z_2 - (IK * OFF) \quad (8.20)$$

$$x_2' = x_2 \quad (8.21)$$

Case 6 $z_2 < z_1$ and $x_2 = x_1$

$$z_1' = z_1 \quad (8.22)$$

$$x_1' = x_1 - (IK * OFF) \quad (8.23)$$

$$z_2' = z_2 \quad (8.24)$$

$$x_2' = x_2 - (IK * OFF) \quad (8.25)$$

Case 7 $\quad z_2 = z_1$ and $x_2 < x_1$

$$z_1' = z_1 + (IK * OFF) \quad (8.26)$$

$$x_1' = x_1 \quad (8.27)$$

$$z_2' = z_2 + (IK * OFF) \quad (8.28)$$

$$x_2' = x_2 \quad (8.29)$$

Case 8 $\quad z_2 > z_1$ and $x_2 = x_1$

$$z_1' = z_1 \quad (8.30)$$

$$x_1' = x_1 + (IK * OFF) \quad (8.31)$$

$$z_2' = z_2 \quad (8.32)$$

$$x_2' = x_2 + (IK * OFF) \quad (8.33)$$

8.5.4 Offsetting a Circular Element

In order to obtain the points of cutter compensation for circular arcs, a different approach should be used. A subroutine NML has been established for this purpose. The calculation is based on the equal intercept theorem along the normals to the circle at the points of the contours (Fig. 8.12). For a circular arc PQ, the co-ordinates of the offset points P' and Q' are given by:

$$z_1' = F + ((R + JK * OFF)/R) * (z_1 - F) \qquad (8.34)$$

$$x_1' = G + ((R + JK * OFF)/R) * (x_1 - G) \qquad (8.35)$$

$$z_2' = F + ((R + JK * OFF)/R) * (z_2 - F) \qquad (8.36)$$

$$x_2' = G + ((R + JK * OFF)/R) * (x_2 - G) \qquad (8.37)$$

where R is the radius of the arc,

(F,G) is the centre of the arc, and

JK equals 1 if the tool locates outside of the arc, and equals -1 if the tool is inside.

8.5.5 The Change Point Between Two Offset Tool Paths

The problem of tool offset becomes more complex when the profile changes. Consider the profile illustrated in Fig. 8.13a, the offset tool path for the profile PQR can be established, utilizing the STL algorithm, into two offset paths P'Q' and Q"R'. There are two methods by which the cutter movement can be accomplished from Q' to Q", by extending the line P'Q' to the point of intersection with an extension of the line Q"R', as shown in Fig. 8.13b; or by introducing circular interpolation between Q' and Q" (Fig. 8.13c). The latter method, however, cannot be applied in concave profiles as depicted in Fig. 8.14. Hence, it has been decided that for concave profiles, the change point of the two offset tool paths is their point of intersection, as depicted in Fig. 8.15. On the other hand, for the case of a convex profile, a circular arc will be added to blend the two offset tool paths as shown in Fig. 8.16.

First considering the case for the convex profile PQR as shown in Fig. 8.16. The start point of that blending arc will be the end point Q' of the first offset path, while the end point of the blending arc is the start point Q" of the second offset path, and the original changing point Q of the profile is the centre of that arc whose radius equals to the amount of offset OFF. The direction of this blending arc will be such that for the tool on the right hand side of the profile, the arc should always be counter-clockwise, while conversely, the arc will be clockwise if the tool is on the left hand side of the profile (Fig. 8.17).

Going back to the cases of the concave profile as illustrated in Fig. 8.15. For a combination of linear and circular contours, there are four different situations of intersection, namely,

(1) straight line to straight line,

(2) straight line to circular arc,

(3) circular arc to straight line, and

(4) circular arc to circular arc.

Despite all the different combinations, the same problem is to obtain the point of intersection of the offset paths. Thus, it is necessary to apply either the STL or NML algorithm, depending on the type of the elements concerned, to obtain the co-ordinates for the terminal points of the two offset tool paths respectively. For a linear element, the two terminal points are sufficient enough to represent a straight line. Whereas for a circular path, the centre of the offset arc is the same centre for the original arc, and the radius of the arc is obtained by increasing or decreasing the origianal radius R by the offset amount OFF, depending on whether the path locates outward or inward of the original circular profile element. Thereafter, the working principles in the Roll Editor program for obtaining the intersecting point between two elements, as explained in Section 6.4, can be applied here for the same purpose. Nonetheless, a subroutine LNLN has been developed to tackle the first case when the two offset paths are linear; for the second and third cases, where a linear path joins a circular path or vice versa, a subroutine LNCIR has to be used; and finally, subroutine CIRCIR has been developed for computing the intersection between two circular paths. The procedures and principles for the algorithms of these subroutines are the same as their analogies in the Roll Editor software, and hence are not described again.

8.6 MACHINING CYCLES

The machining of form-rolls mainly utilizes external turning operations. The whole turning process is basically divided into two sections, roughing out (area clearance) and finish profiling.

With the CUTPLOT software, it is possible to choose from four special-purpose turning cycles, to do the roughing out or finishing cut. These machining cycles can automatically generate the entire sequence of motions necessary to rough or finish the part, and make all the decisions for tool placements at the beginning and end of each cut and throughout the entire machining sequence. The four machining cycles available are roughing, grooving, pocketing, and finishing. The roughing cycle is mainly designed for the area clearance of convex profiles; for concave profiles, area clearance can be done by applying either the grooving cycle or the pocketing cycle; and finally, the finishing cycle is designed for the finish profiling.

The application of a machining cycle refers to a designated machining area, which should be defined by the element numbers for the starting and ending elements. The choice of the machining cycle is denoted by a numerical code ICYCLE, for which the user needs only to input ICYCLE equals 1 for the roughing cycle; 2 for grooving; 3 for finishing and 4 for pocketing. The choice of a machining cycle has to be accompanied by the choice of cutting tool, the depth of cut, the cutting speed and the feed rate. However, for the grooving and finishing cycles, the depth of cut should be zero.

For each machining cycle, the offset contour will be developed for the designated machining area segment. The tool placements for the complete machining sequence will then be generated in accordance with the specified machining cycle. The positions of the tool, referring to the virtual tool tip (Fig. 8.1), will be put into a cutter location data (CLDATA) file. Eventually, after finishing the CUTPLOT program, this CLDATA file can then be converted to an NC machine control program by a post-processor. Details of the post-processor will be gone through later; while each of the four different machining cycles are described separately in the following.

8.6.1 Roughing Cycle

The module ROUGH is designed and developed to generate the cutter locations for the roughing cycle. This routine defines the basic turning operation of machining repeatedly parallel to the Z-axis, by doing so, the blank material of the machining area segment can then be removed to the specified contour segment with the desired material allowance left for finish profiling (Fig. 8.18). The features of the roughing cycle are as follows:

(1) Cutting is done parallel to the Z-axis, while the machining direction is governed by the direction of the machining area.

(2) The first tool movement of the complete roughing out cycle is to position the tool to the machining start point, being done at rapid traverse from a previous stop point, which is usually the pre-set tool changing position (Fig. 8.18). This machining start point is set to be at a clearance of 2 mm away from the start line.

(3) The actual machining is done as a repetition of a three-stroke cutting cycle, as shown in Fig. 8.19. The first stroke is the advancing stroke, for which the tool advances in cutting feed from its previous position N to the cutting start point P, so as to achieve the specified depth of cut d. The second stroke is the cutting stroke, the tool cuts parallel to the Z-axis from the start point P to the point Q which is at a distance t (the finishing tolerance) away from the contour. Finally, the returning stroke takes place when the tool returns at rapid traverse from point Q to a point R which is at a clearance value c away from the point P. At present, c has been set to be fixed at 2 mm.

(4) The end point of a cutting stroke is determined by computing the point of intersection between the cutting line and the offset profile element. For a linear profile element, as shown in Fig. 8.20a, the co-ordinates (z_q, x_q) of the point Q are given by

$$x_q = x_p \tag{8.38}$$

$$z_q = z_L + (x_L - x_q)*(x_L - x_S)/(z_L - z_S) \tag{8.39}$$

for a circular element, as shown in Fig. 8.20b,

$$x_q = x_p \tag{8.40}$$

$$z_q = F \pm \sqrt{R^2 - (G - x_p)^2} \tag{8.41}$$

where the application of the + or - sign in equation (8.41) depends on whether the tool locates outside or inside of the circular arc.

(5) All undercuts or concave contour segments will be ignored to be uncut, as shown in Fig. 8.21.

(6) After all the possible Z-directional cuts, there will be a final cut parallel to the contour of the specified contour segment, results in a offset contour with the desired material allowance left for finish profiling. Again, all undercuts and concave contours will be skipped automatically.

(7) Finally, the tool will return rapidly to its pre-set tool changing position.

8.6.2 Pocketing Cycle

For concave roll contour profiles, the roughing cycle cannot be applied for area clearance. As shown in Fig. 8.22, area ABCD is a simple concave contour. A method to remove the material is to choose a turning tool of larger clearance angle so that this tool can be directed to 'dig in' and remove the material.

If for example, the angle ANG between the line AB and the Z-axis is smaller than the clearance angle CL of the tool, as depicted in Fig. 8.23. It is then possible to choose this line AB as a cut-in line. The tool can then dig in the material along this line with an advancement equals to the depth of cut d, and a cutting stroke can be carried out as that in the roughing cycle.

In some cases, however, due to the large angle ANG, neither the line AB nor CD can be used as the cut-in line. Thus, the area ABCD can be subdivided into two or more segments, and specific cut-in lines must be input for each segment. Referring to the area ABCD in Fig. 8.24, a line L can be introduced as the cut-in line. A left hand tool can then be used to clear the area RSCD, and thereafter, area RSBA can be machined by a right hand tool consequently. In order to define the cut-in line L, the co-cordinates (z_p, x_p) of a point P and the angle ANG between the line L and the Z-axis should be specified. The cut-in line can only be valid if it intersects with the starting element of the machining area, otherwise, the user will be requested to define the cut-in line again.

The features of the pocketing cycle are more or less the same as those features of the roughing cycle. The main difference being the introduction of the cut-in line in the pocketing cycle. As a result, a four-stroke cutting cycle has to be employed, as depicted in Fig. 8.25. Considering the current cutting cycle EFGH, the first tool movement is the digging-in stroke, for which the tool digs in the material along the cut-in line with an amount equals to the depth of cut d. That is, the tool moves at the cutting feed rate from point E to the cutting start point F. The second stroke is the cutting stroke, the tool cuts parallel to the Z-axis from the start point F to the end point G, which is at a distance t (the finishing tolerance) away from the contour. The third movement takes place by returning the tool in rapid traverse to a point H which is at a vertical clearance c away from the point F. Finally, the tool will move back to the cutting start point F at the feed rate, to start another cycle.

In addition to the above mentioned cut-in line and the cutting tool movements, all the facilities available in the roughing cycle are also applicable in the pocketing cycle. Thus, the pocketing cycle is in fact a special roughing cycle for machining concave profile segments.

8.6.3 Grooving Cycle

Another way for removing material in concave areas is by utilizing a parting-off tool to carry out the grooving operation in the diameter direction, as shown in Fig. 8.26. The grooving cycle option is designed for this purpose, features of the grooving cycle are as follows:

(1) Cutting is done parallel to the X-axis.

(2) The first tool movement of the complete grooving cycle is to position the tool to the machining start point, at rapid traverse from a previous stop point (Fig. 8.26). The machining start point is set at a clearance value of 3 mm above the maximum radius of the blank.

(3) The actual machining is done as a repetition of a three-stroke cutting cycle, as shown in Fig. 8.27. The first stroke is the actual grooving stroke parallel to the X-axis, from the

cutting start point E to the end point F. The second stroke is to rapidly return the tool to the cutting start point E. Then the tool moves to a new cutting start point G, where the length EG equals to half of the tool width of the grooving tool.

(4) For the ending point of the cutting stroke, the Z-co-ordinate of the end point is the same as that for the cutting start point, as cutting is parallel to the X-axis. However, for the X-co-ordinate, the largest radius of the offset profile segment within the width of the grooving tool will be selected, as shown in Fig. 8.28.

(5) The tool will finally return to its pre-set tool changing position.

8.6.4 Finishing Cycle

After roughing out has been completed, finish profiling can be specified by calling up the finishing cycle. It is also possible to do medium finishing before the final profiling by specifying a desired material allowance left for finishing, as shown in Fig. 8.29. Final finishing can be achieved by simply assigning the material allowance to be zero.

If desired, it is also possible, just like the pocketing cycle, to designate the machining start point from its previous stop position. Cutting will then be done along the offset contour until reaching the ending point of the machining area. The tool will then rapidly retreat to its rest position.

8.7 OUTPUT OF THE CUTPLOT SOFTWARE

During the execution of the CUTPLOT software, the tool placements generated for the machining process will be dumped into a cutter location data file CLDATA. The file will be kept in the hard disc storage of the PERQ system, and hence can be retrieved by the post-processor program later on, to generate the NC machine control tape program for the actual machining process. On the other hand, a plot of the tool paths will be displayed on the PERQ screen, as shown in Plate 8.1. As a matter of fact, this tool path plot is a graphical representation of the CLDATA file. Therefore, the user can follow the path of the tool on the screen, and see if the correct form of roll is being produced in the correct manner. In addition, as the drawing routines for the HP7221c plotter have been incorporated in the CUTPLOT software as well, a hardcopy of the cutter path plot can be produced by the HP7221c plotter. An example of the plotter drawing of the tool path is shown in Fig. 8.30.

8.8 PROGRAM STRUCTURE OF THE CUTPLOT SOFTWARE

The CUTPLOT software consists of three distinctive units, namely;

(1) THE ROLL MACHINING PROGRAM (CUTPLOT)

(2) THE TOOL OFFSET MODULE (OFFSET)

(3) THE PLOTTER DRAWING SUBROUTINE (PLOTHP)

The hierarchy of the program structure is shown in Chart 8.1, with the main program CUTPLOT residing at the first level of execution, controlling the sequence of execution of the subroutines at lower levels. A brief account of the nature of the subroutines used at each level is given in Table 8.2.

8.8.1 The Tool Offset Module (OFFSET)

This module has been designed to determine the offset path for the designated manufacturing contour segment. The hierarchy of the subroutines in the OFFSET module is as shown in Chart 8.2, and a brief account of the nature of each subroutine is given in Table 8.3.

8.8.2 The Plotter Drawing Subroutine (PLOTHP)

This subroutine has been developed to call upon the 'HP7221' plotter driver facilities (Appendix 2) to plot the tool path drawing by the HP7221c plotter. The full size drawing has to be done on A3 size paper, but it is also possible for the user to specify the scale of the drawing.

8.9 THE POST-PROCESSOR

The function of a post-processor program is to convert the cutting location data (CLDATA) information to the exact requirements of the particular machine and control system on which the component will be cut [60]. Although the NC technology has now been well developed, yet there is not a commonly used standard for the NC machine tool control tape program format. In fact, different control systems require different programming codes and different control tape requirements. Post-processors are therefore necessary, for converting the CLDATA to an NC control tape that suits the requirement of the particular NC control system.

A post-processor program MFPOST, thus, has been designed and developed to process the CLDATA information, and generate an NC control tape program in accordance with the exact requirements of the Fanuc controlled Mori-Seki CNC lathe. As described before, the operation of the CUTPLOT processor generates a CLDATA file. The MFPOST post-processor will retrieve this CLDATA file from the disc memory storage, and then further process the CLDATA information into a control tape program file, with appropriate formats and syntax for the Mori-Fanuc CNC system. In comparison with the CUTPLOT processor, the MFPOST post-processor is a relatively small FORTRAN program, which can be considered to be mainly data processing with a good deal of syntax and sequence checking of the CLDATA.

8.9.1 Output of the Post-Processor

During the processing of the MFPOST program, the NC control program and the display of the tool path are displayed on the PERQ screen, as shown in Plate 8.2. In this case, the display of the tool path is the actual interpretation of the NC machine control program; while in the case of the CUTPLOT software as described in Section 8.7, the tool path plot was based on the information from the CLDATA file. This approach, therefore, can provide a check on the syntax of the NC tape program. The NC control tape program will be kept in the hard disc storage as a permanent file under the file name specified by the user. Therefore, the user can dump this file into a paper tape, which can then be used on the CNC lathe for the machining process. An example of the NC control tape program, which was generated by the MFPOST program, is listed in Fig. 8.31.

8.9.2 Program Structure of the MFPOST Post-Processor

The MFPOST software consists of two units, namely:

(1) THE POST-PROCESSOR PROGRAM (MFPOST)

(2) THE TOOL PATH PLOT MODULE (DETAPE)

The hierarchy of the MFPOST program is given in Chart 8.3, while a brief account of the nature of the subroutines used at each level is given in Table 8.4. Besides, the hierarchy of the DETAPE module is given in Chart 8.4, and Table 8.5 gives a brief account of the subroutines in the module.

8.10 OPTIONAL TAPE CHECKING

Sometimes the user may want to re-check the information contained in the NC control tape program file before the actual machining process. Thus, a tape checking program has been prepared to fulfil this requirement. This tape checking program CHKTAP is in fact developed from the DETAPE module in the MFPOST program. The operation of the CHKTAP program will produce a plot of the tool path on the PERQ screen, which is similar to the display from the MFPOST post-processor, and hence acts as a means for visually inspecting the NC tape program. The program structure of the CHKTAP program is illustrated in Chart 8.5, while Table 8.6 gives a brief account of the subroutines. Nevertheless, the operation of the CHKTAP program is only optional, is literally a quality control tool and will not affect the normal procedure of the CAD/CAM system.

8.11 PREPARING AN NC TAPE WITH THE SPECIAL-PURPOSE PROCESSOR AND POST-PROCESSOR

In conclusion, according to the computer generated geometric data of the form-roll, the user can use the CUTPLOT processor, with a minimum input of technological data, to generate a CLDATA for the complete machining process. A typical listing of data input for the CUTPLOT software is shown in Fig. 8.32. The MFPOST post-processor will then operate subsequently, retrieving data from the CLDATA file, and then generate an NC control tape program in accordance with the programming format for the Mori-Fanuc control system. Although the

CUTPLOT processor and the MFPOST post-processor are two separate FORTRAN programs, there is, however, no breakdown between the operation of these two programs. Fig. 8.33 outlines the procedures for preparing an NC tape for machining form-rolls with the CUTPLOT processor and MFPOST post-processor. A set of form-rolls made on the CNC lathe is shown in Plate 8.3.

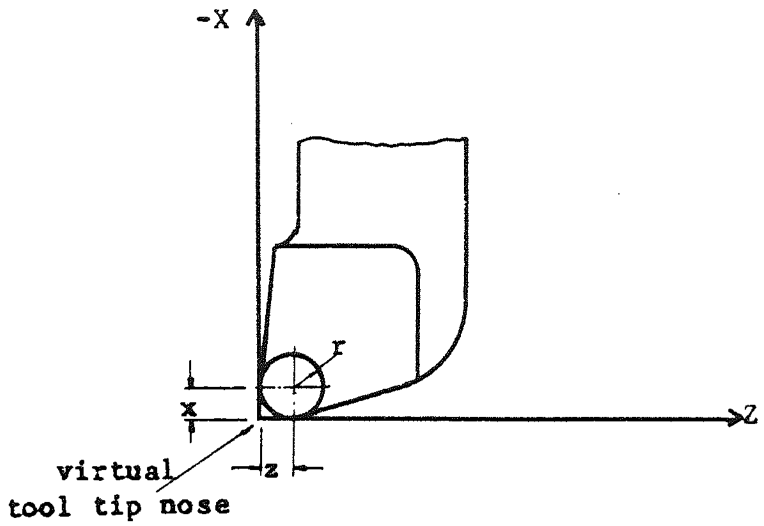


Fig. 8.1 TOOL COORDINATE SYSTEM

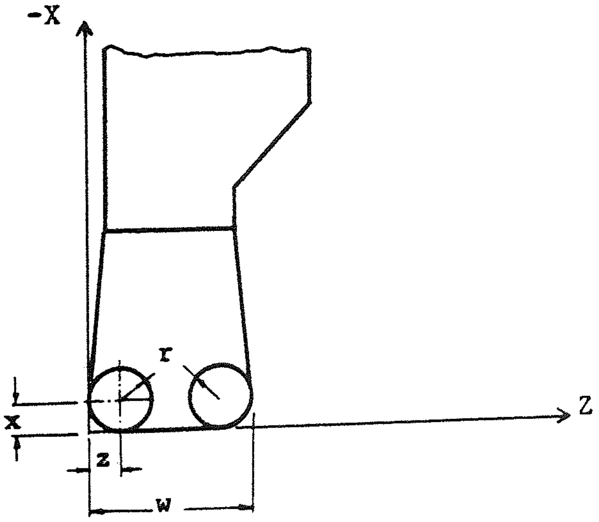


Fig. 8.2 TOOL COORDINATE SYSTEM FOR PARTING-OFF TOOL

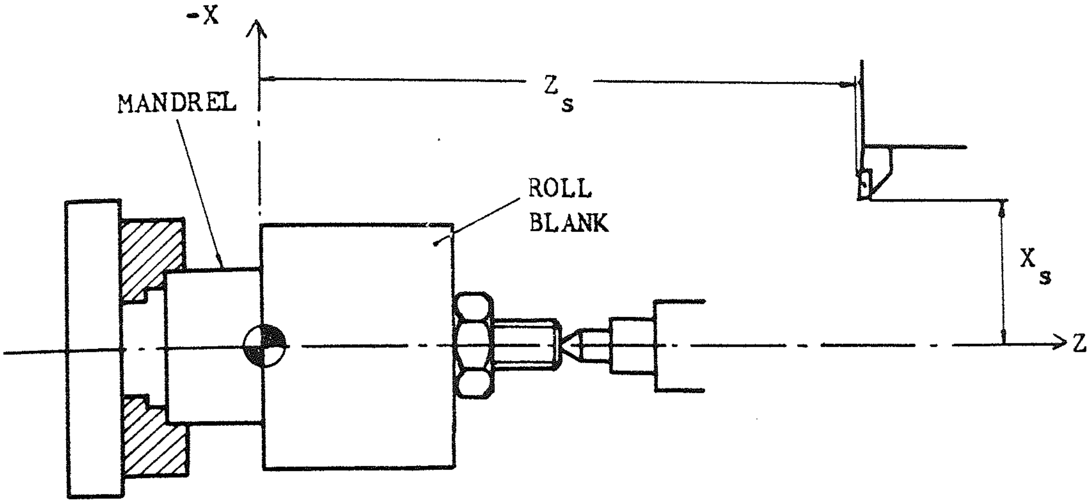


Fig. 8.3 PRE-SET STARTING POSITION

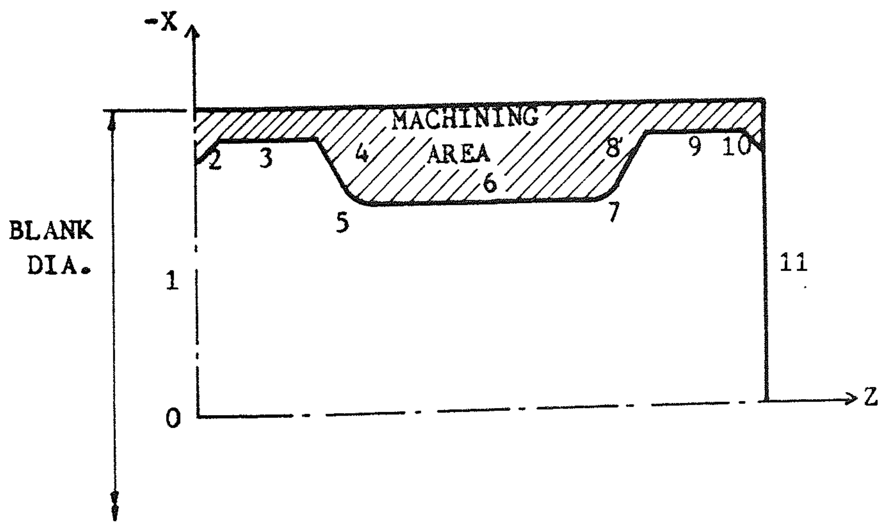


Fig. 8.4 MACHINING AREA

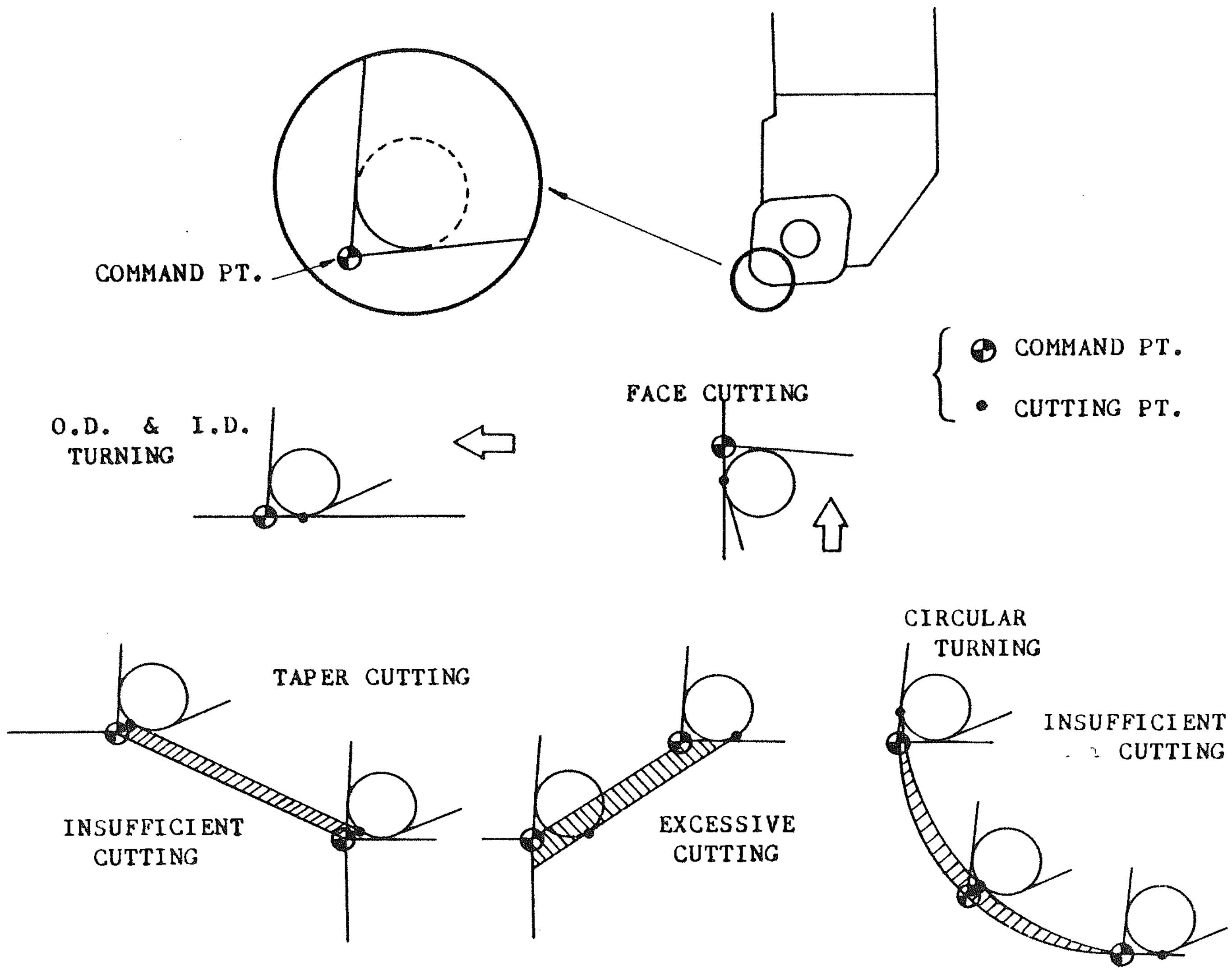


Fig. 8.5 RADIUS COMPENSATION OF TURNING PROCESS

Offset Amount OFF = t + r

r = tip nose radius

t = finishing tolerance

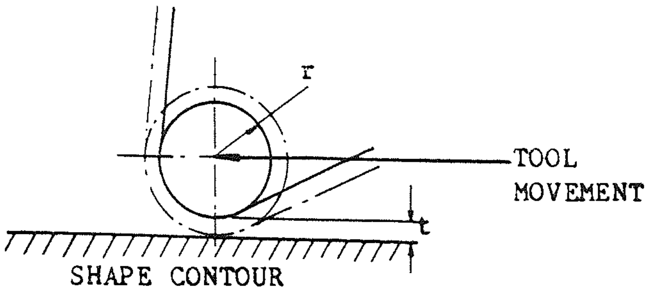


Fig. 8.6 TOTAL CUTTER OFFSET

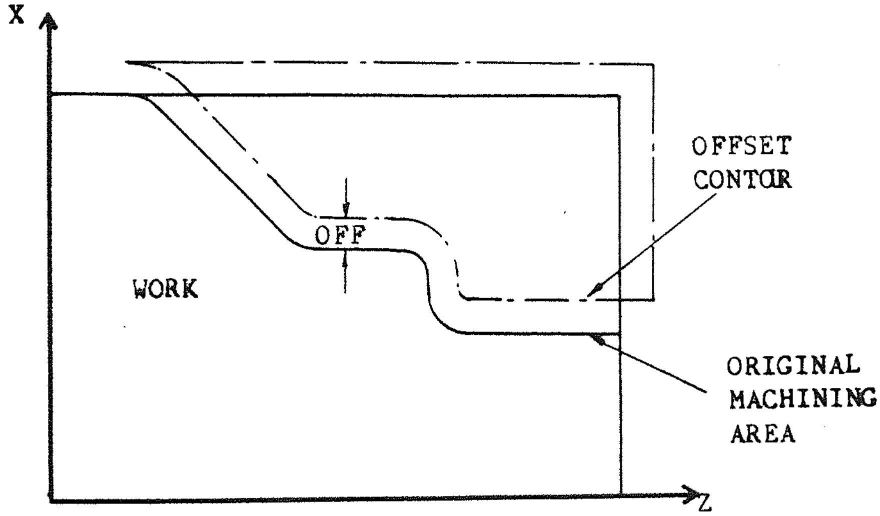


Fig. 8.7 OFFSET CONTOUR FOR A MACHINING AREA

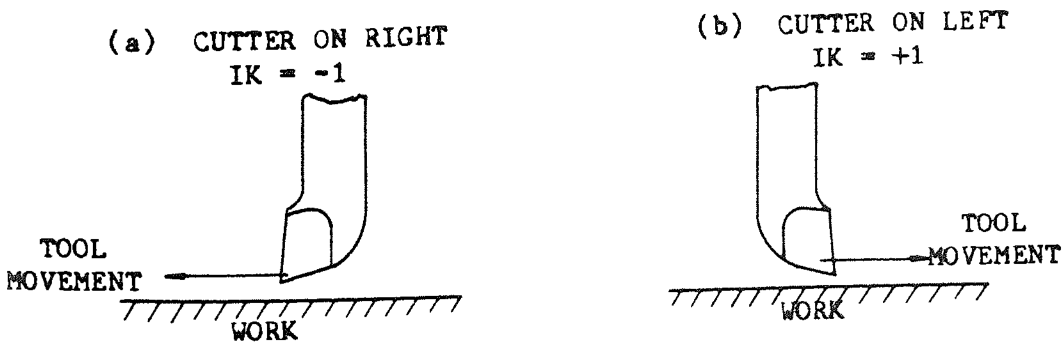


Fig. 8.8 SIGN CONVENTION FOR CUTTER POSITION

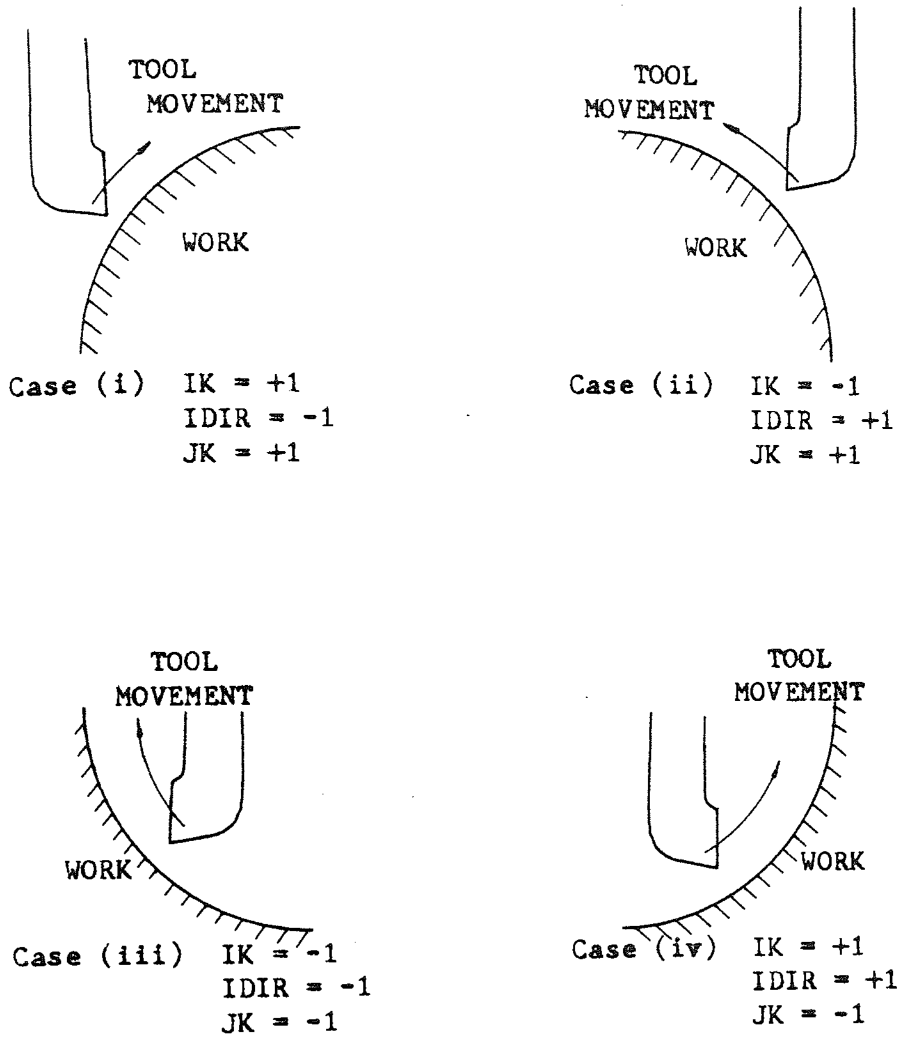


Fig. 8.9 POSITION OF CUTTER RELATIVE TO CIRCULAR ELEMENT

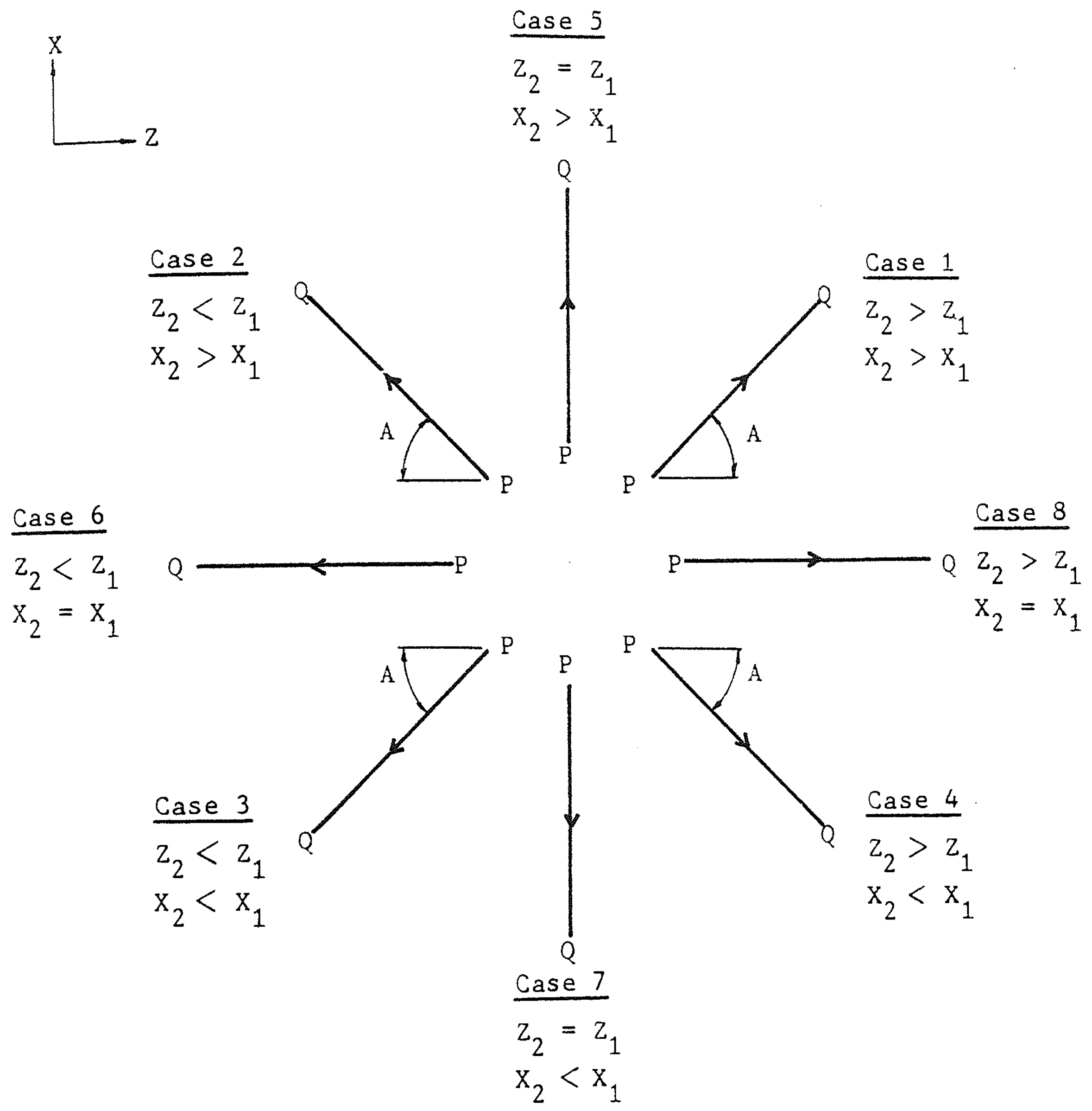
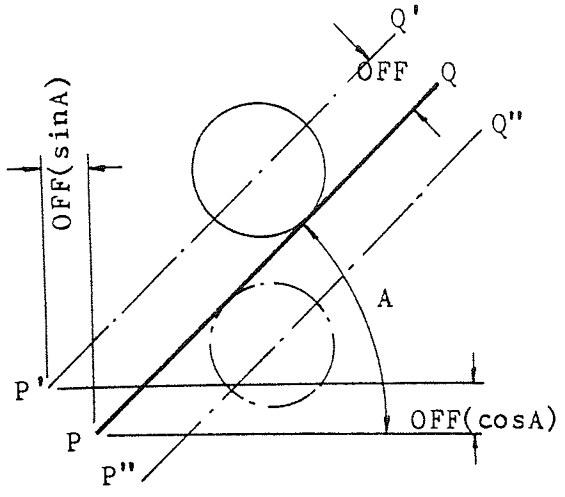
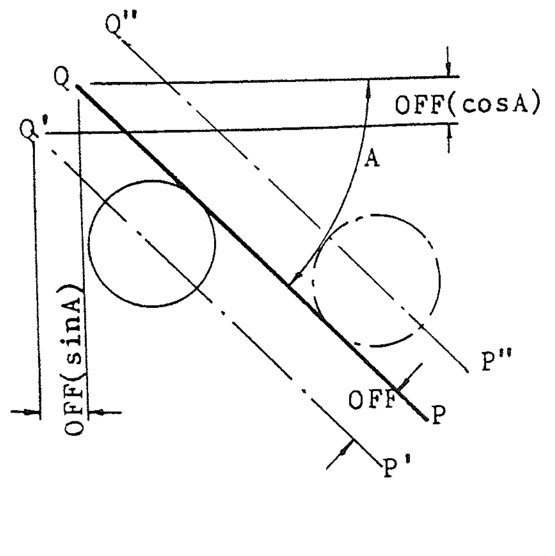


Fig. 8.10 ORIENTATIONS OF STRAIGHT LINE



Case 1

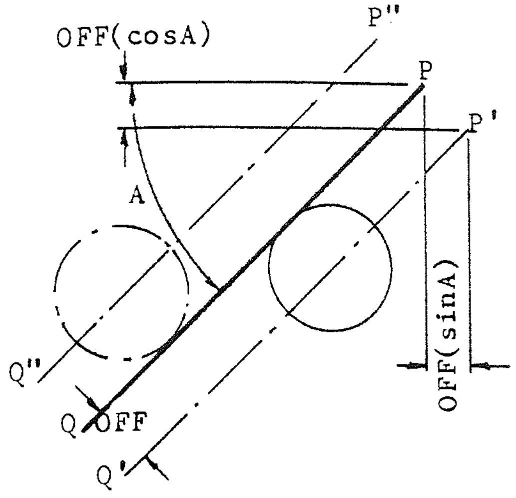
$Z_2 > Z_1$
$X_2 > X_1$



Case 2

$Z_2 < Z_1$
$X_2 > X_1$

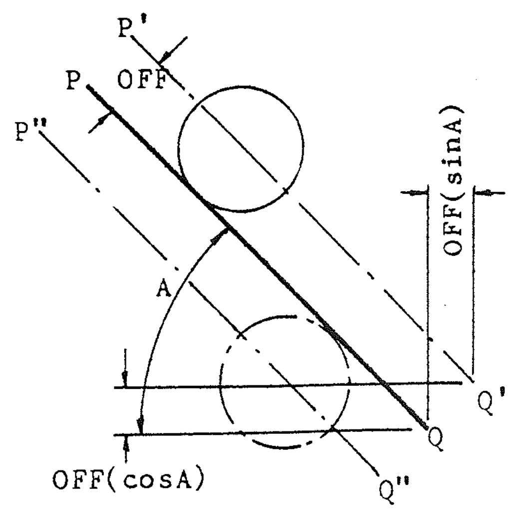
Fig. 8.11 OFFSET OF POINTS ON STRAIGHT LINE (PART 1)



<u>Case 3</u>

$Z_2 < Z_1$

$X_2 < X_1$

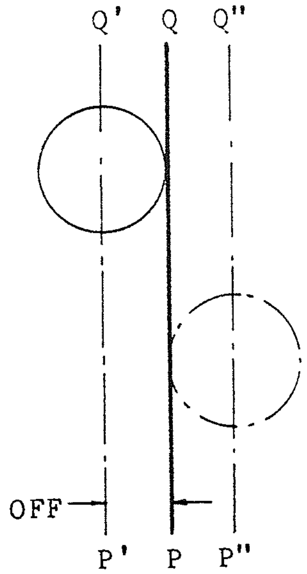


<u>Case 4</u>

$Z_2 > Z_1$

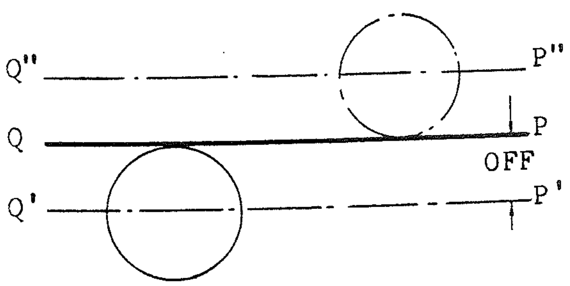
$X_2 < X_1$

Fig. 8.11 (PART 2)



Case 5

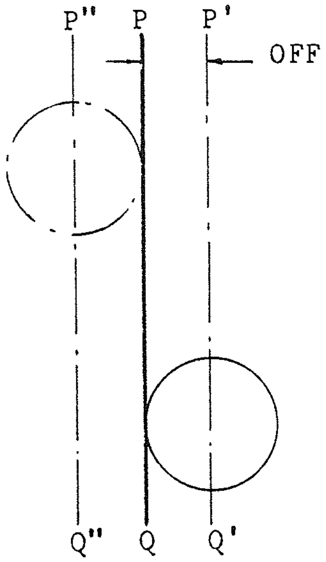
$Z_2 = Z_1$
$X_2 > X_1$



Case 6

$Z_2 < Z_1$
$X_2 = X_1$

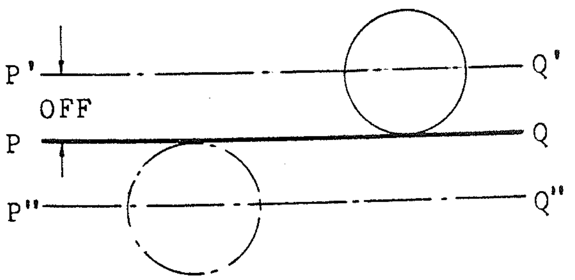
Fig. 8.11 (PART 3)



<u>Case 7</u>

$Z_2 = Z_1$

$X_2 < X_1$

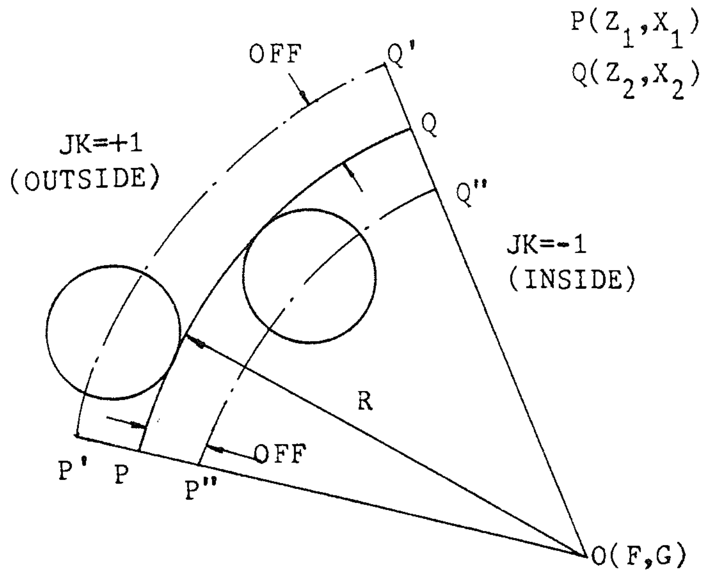


<u>Case 8</u>

$Z_2 > Z_1$

$X_2 = X_1$

Fig. 8.11 (PART 4)



THE EQUAL INTERCEPT THEOREM

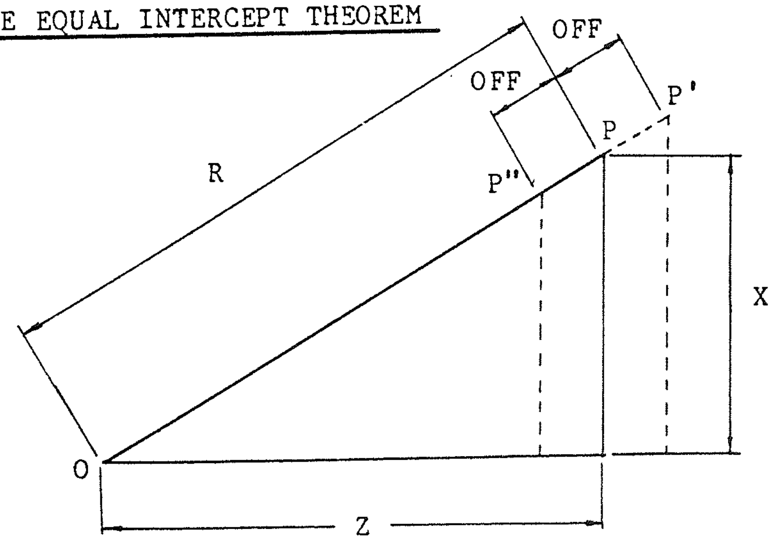


Fig. 8.12 OFFSET OF POINTS ON CIRCULAR ARC

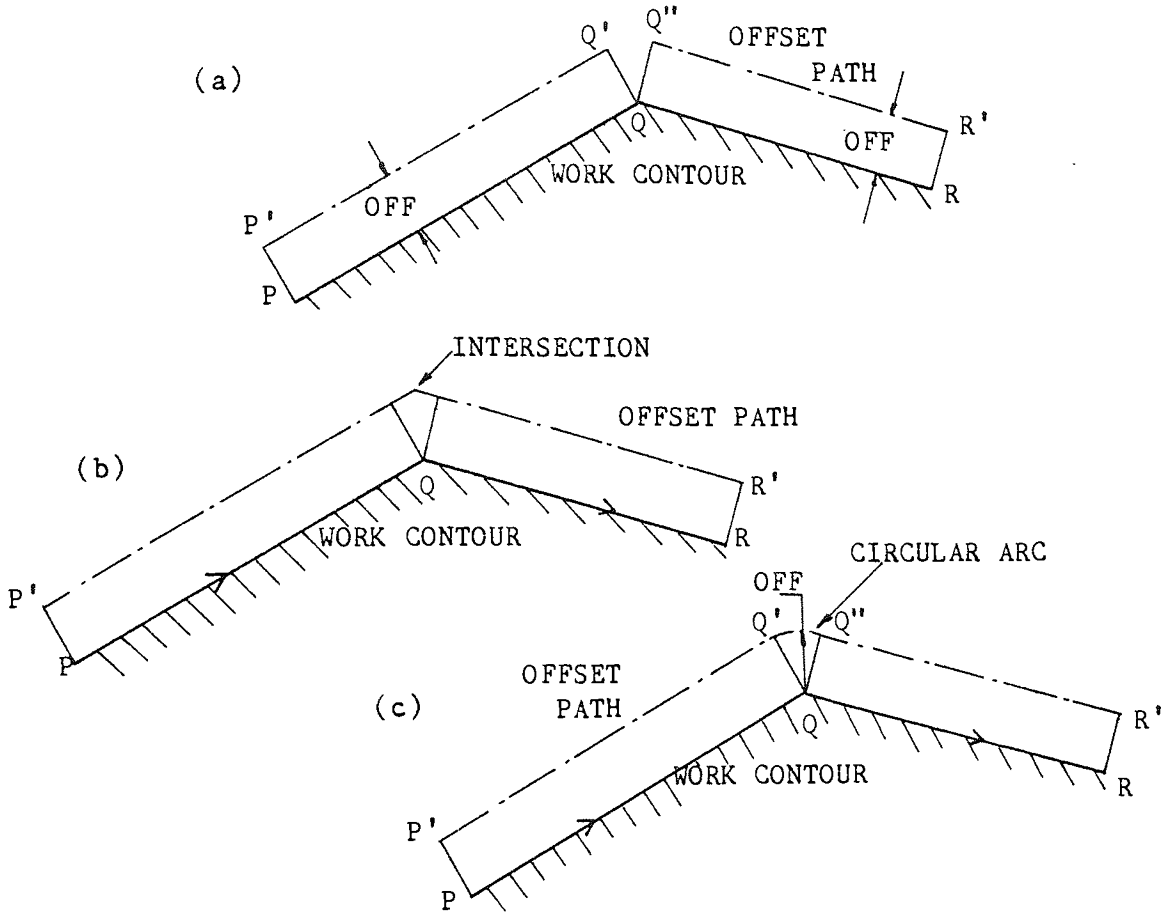


Fig. 8.13 TOOL OFFSET AT CHANGE POINT

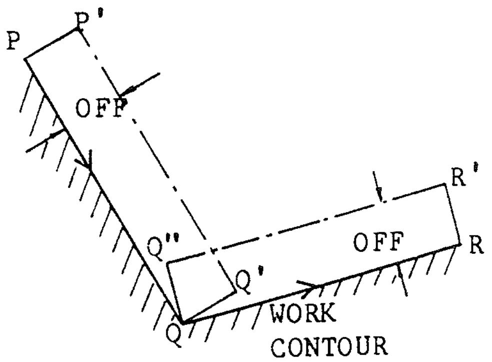


Fig. 8.14 OFFSET PATHS FOR CONCAVE PROFILE

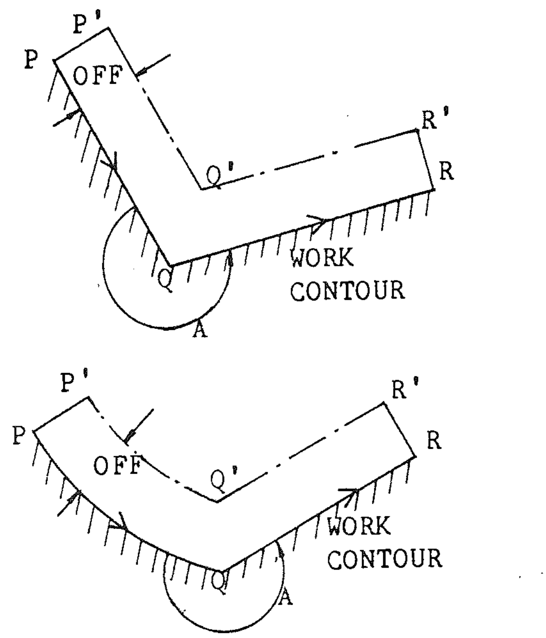


Fig. 8.15 CHANGE POINT FOR OFFSET PATHS FOR CONCAVE PROFILES (A >180^{o})

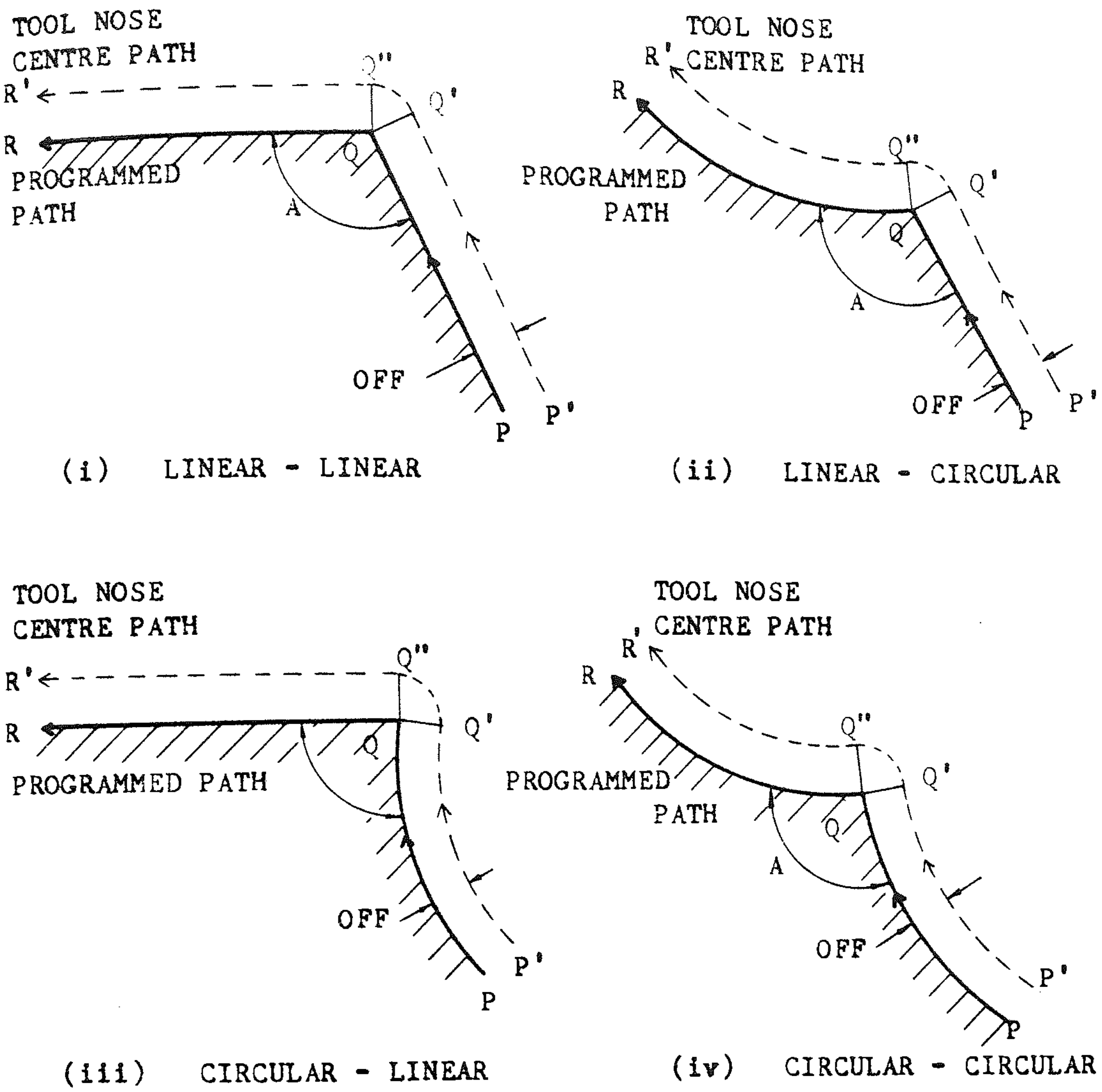


Fig. 8.16 BLENDING OFFSET TOOL PATHS FOR CONVEX PROFILES ($A < 180^{\circ}$)

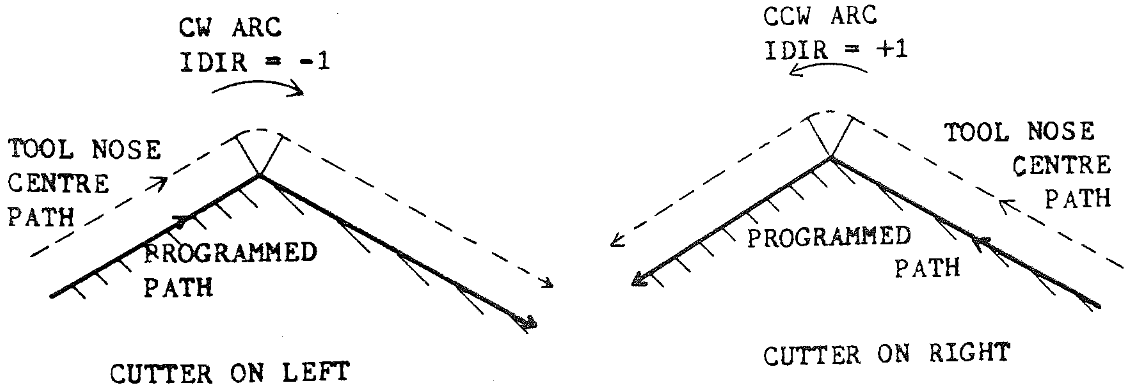


Fig. 8.17 DIRECTION OF THE BLENDING ARC

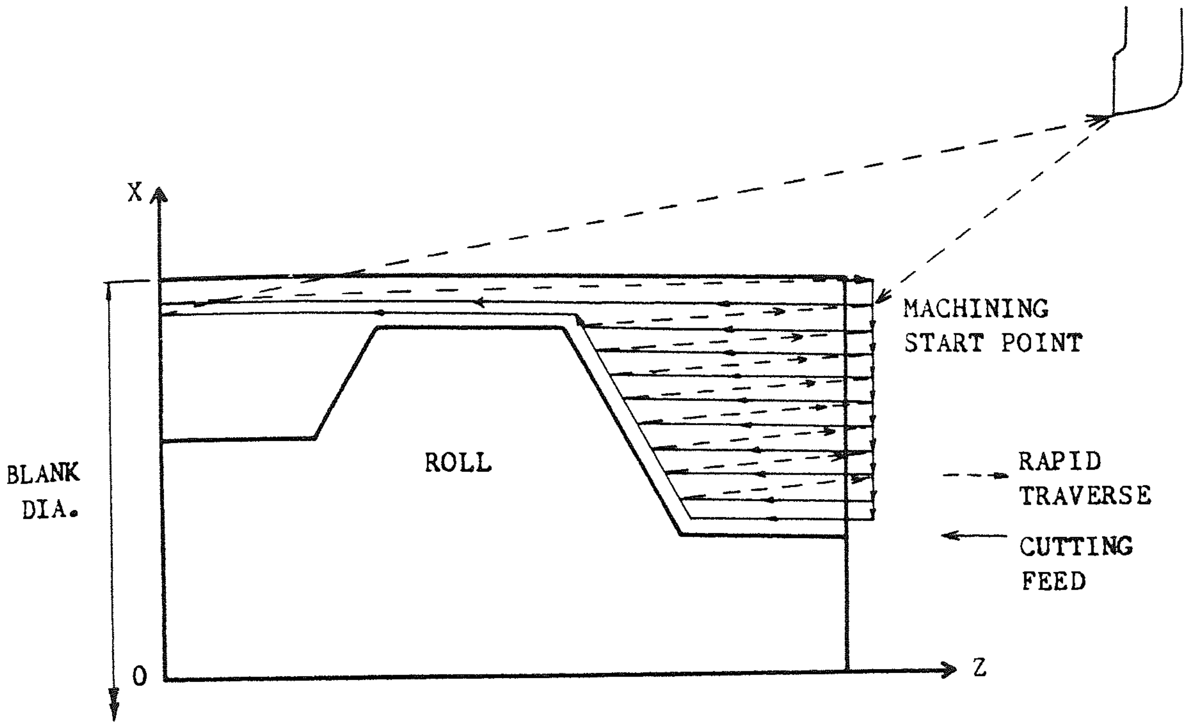
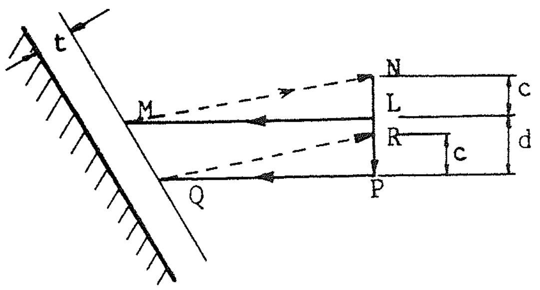
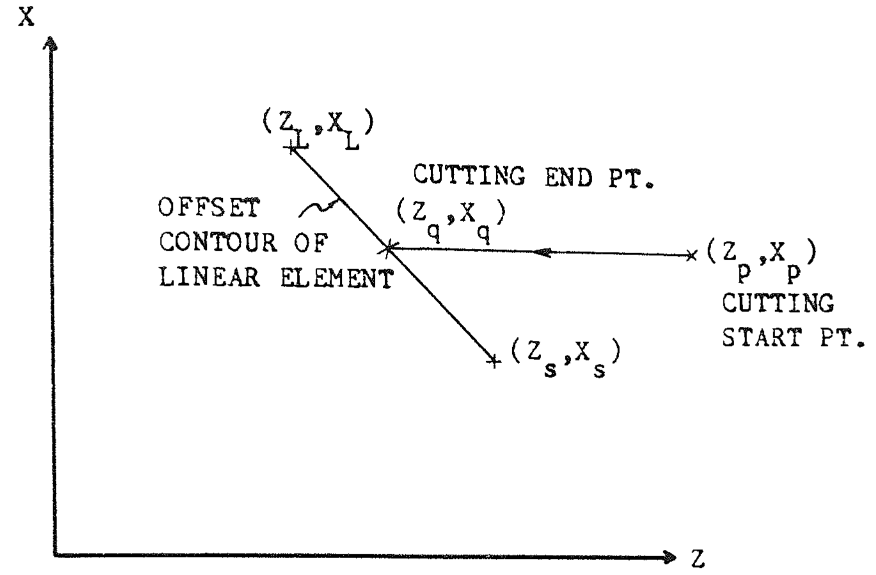


Fig. 8.18 ROUGHING CYCLE

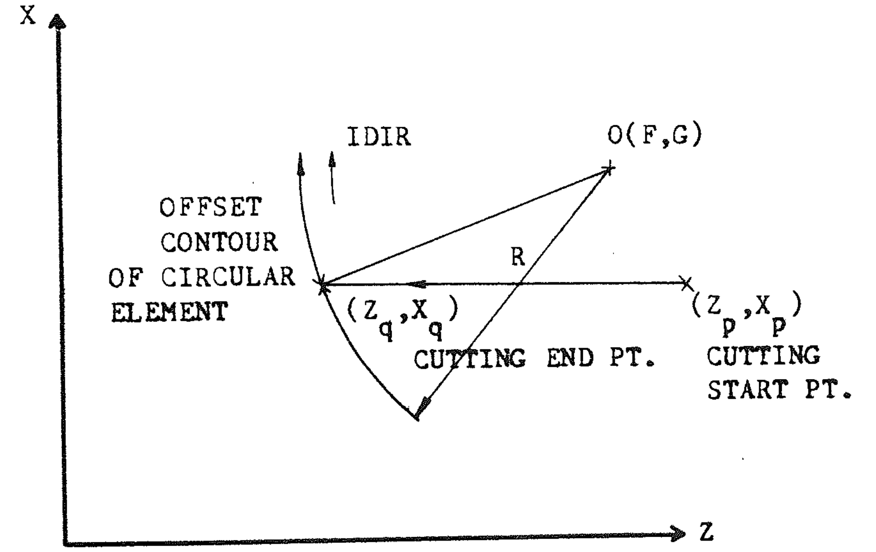


c = clearance value
d = depth of cut
t = finishing tolerance

Fig. 8.19 3-STROKE CUTTING CYCLE FOR ROUGHING



(a) LINEAR ELEMENT



(b) CIRCULAR ELEMENT

Fig. 8.20 END POINT OF A CUTTING STROKE

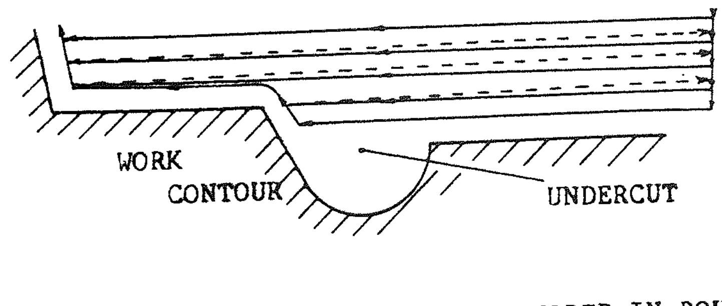


Fig. 8.21 UNDERCUT WILL BE IGNORED IN ROUGHING

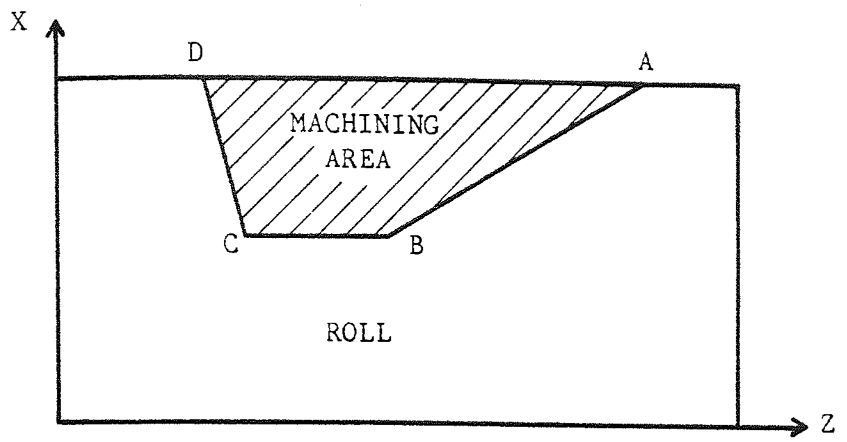


Fig. 8.22 A SIMPLE CONCAVE MACHINING AREA (ABCD)

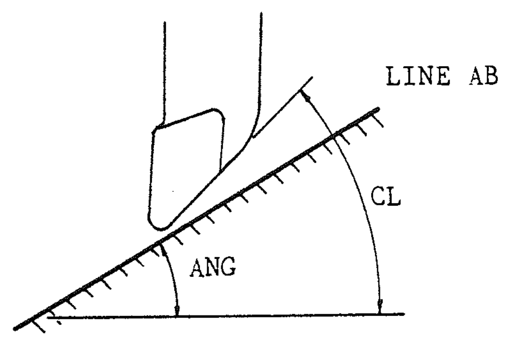


Fig. 8.23 CLEARANCE ANGLE (CL) OF THE TOOL FOR POCKETING CYCLE

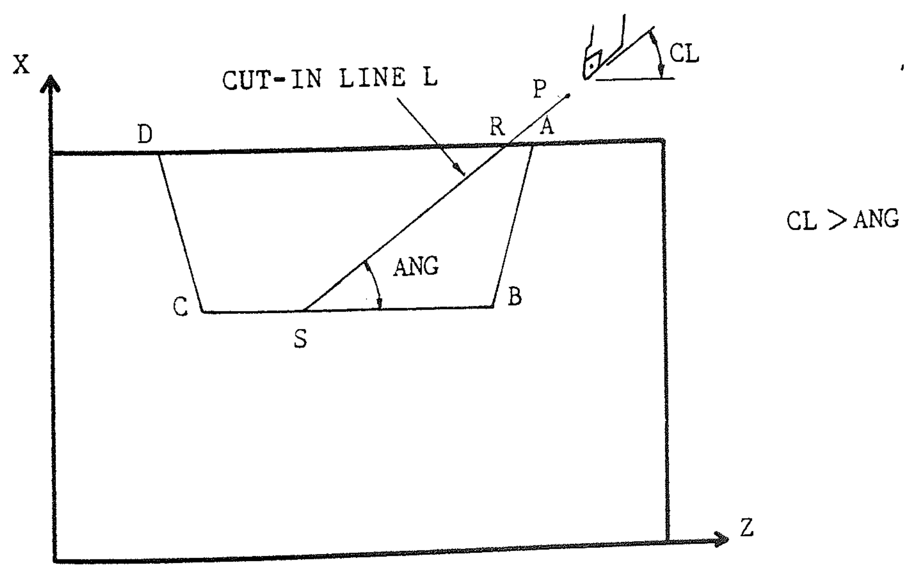


Fig. 8.24 CUT-IN LINE FOR POCKETING CYCLE

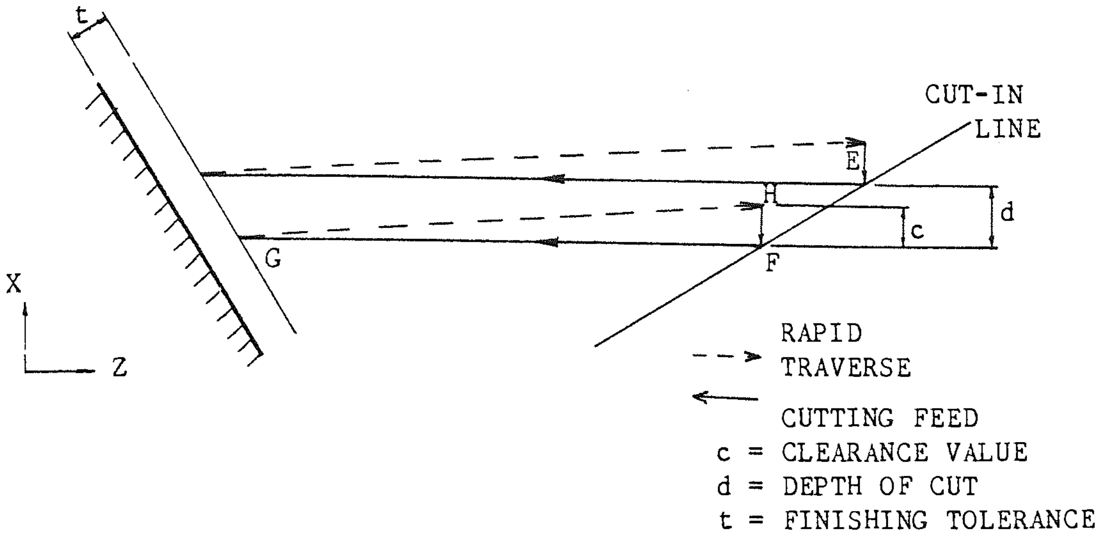


Fig. 8.25 4-STROKE CUTTING CYCLE FOR POCKETING

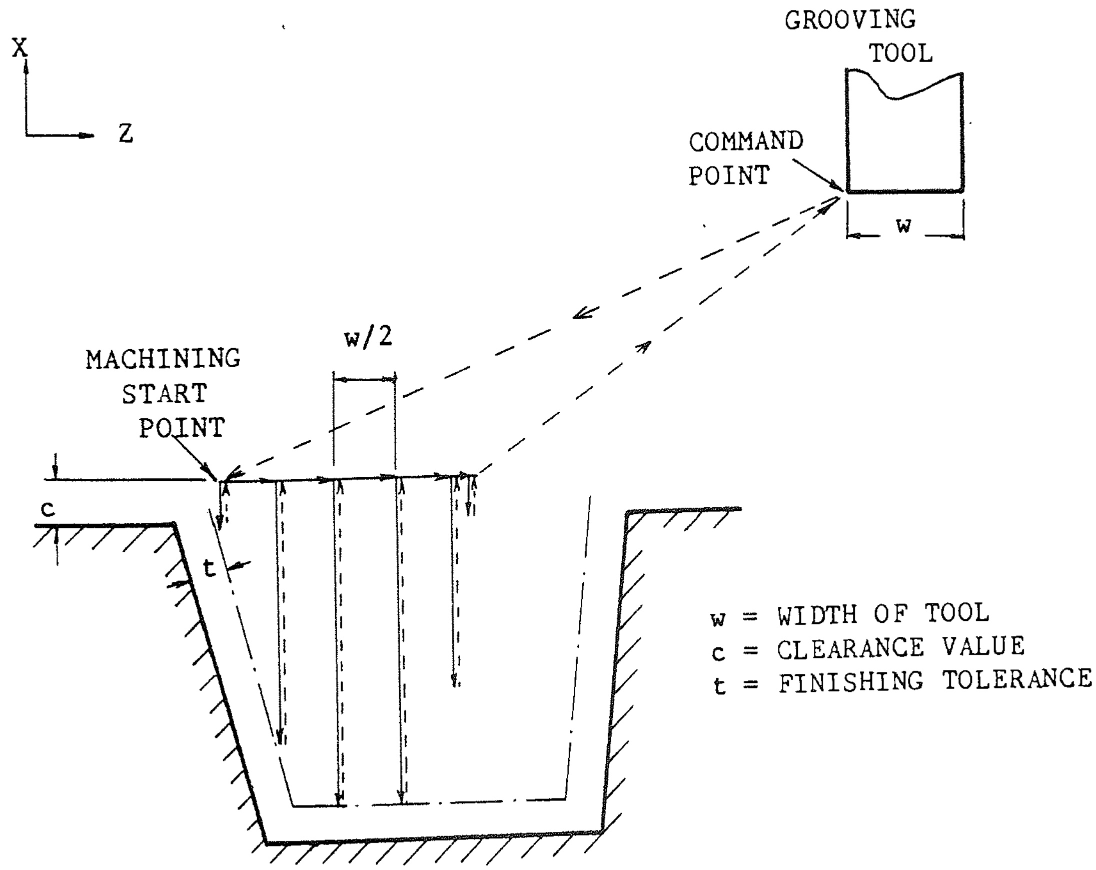


Fig. 8.26 GROOVING CYCLE

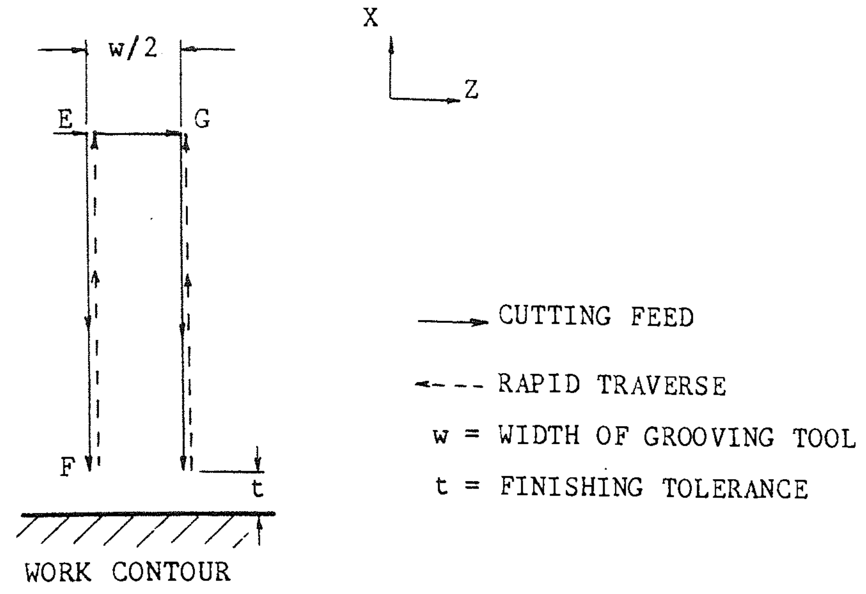


Fig. 8.27 3-STROKE CUTTING CYCLE FOR GROOVING

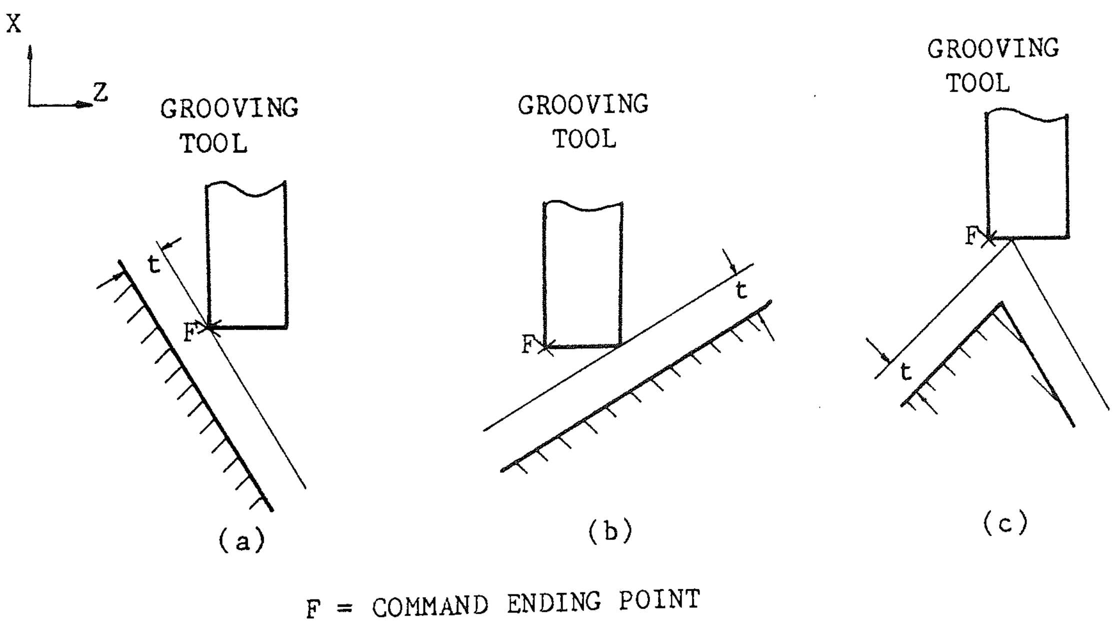
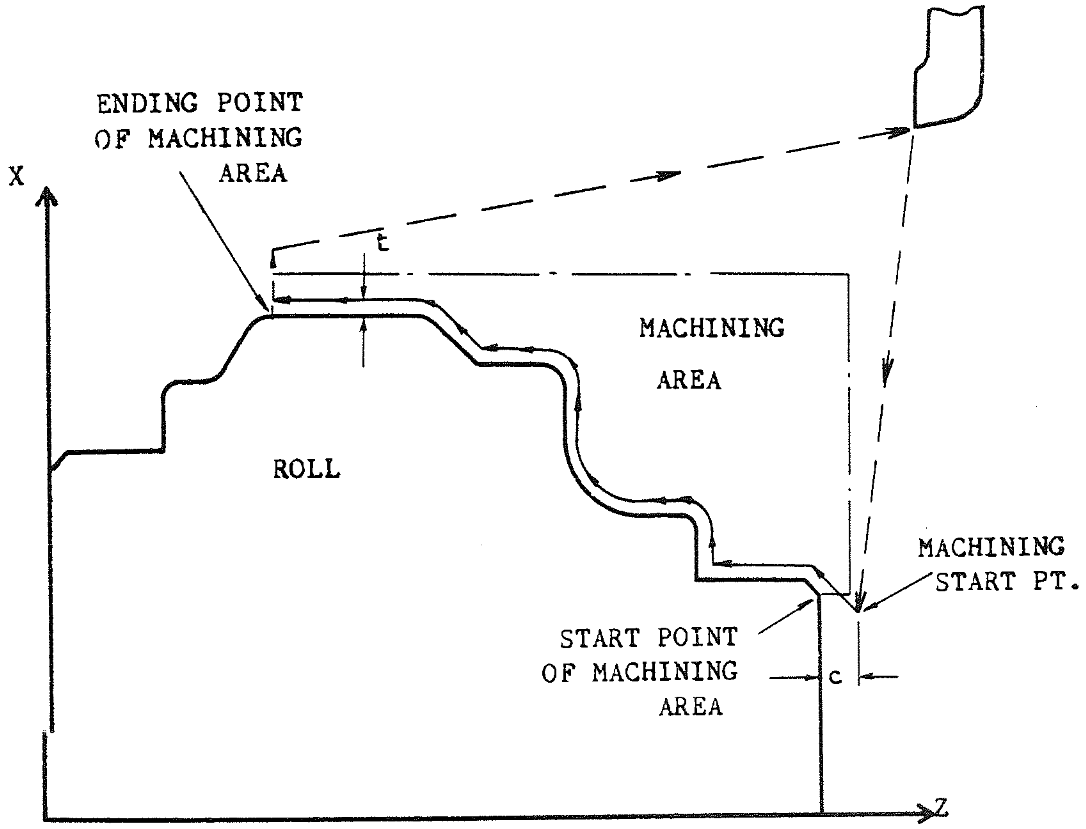


Fig. 8.28 ENDING POING OF THE CUTTING STROKE FOR GROOVING



RAPID TRAVERSE

CUTTING FEED

c = CLEARANCE VALUE

t = MATERIAL TOLERANCE (t = 0 FOR FINAL FINISHING)

Fig. 8.29 FINISHING CYCLE

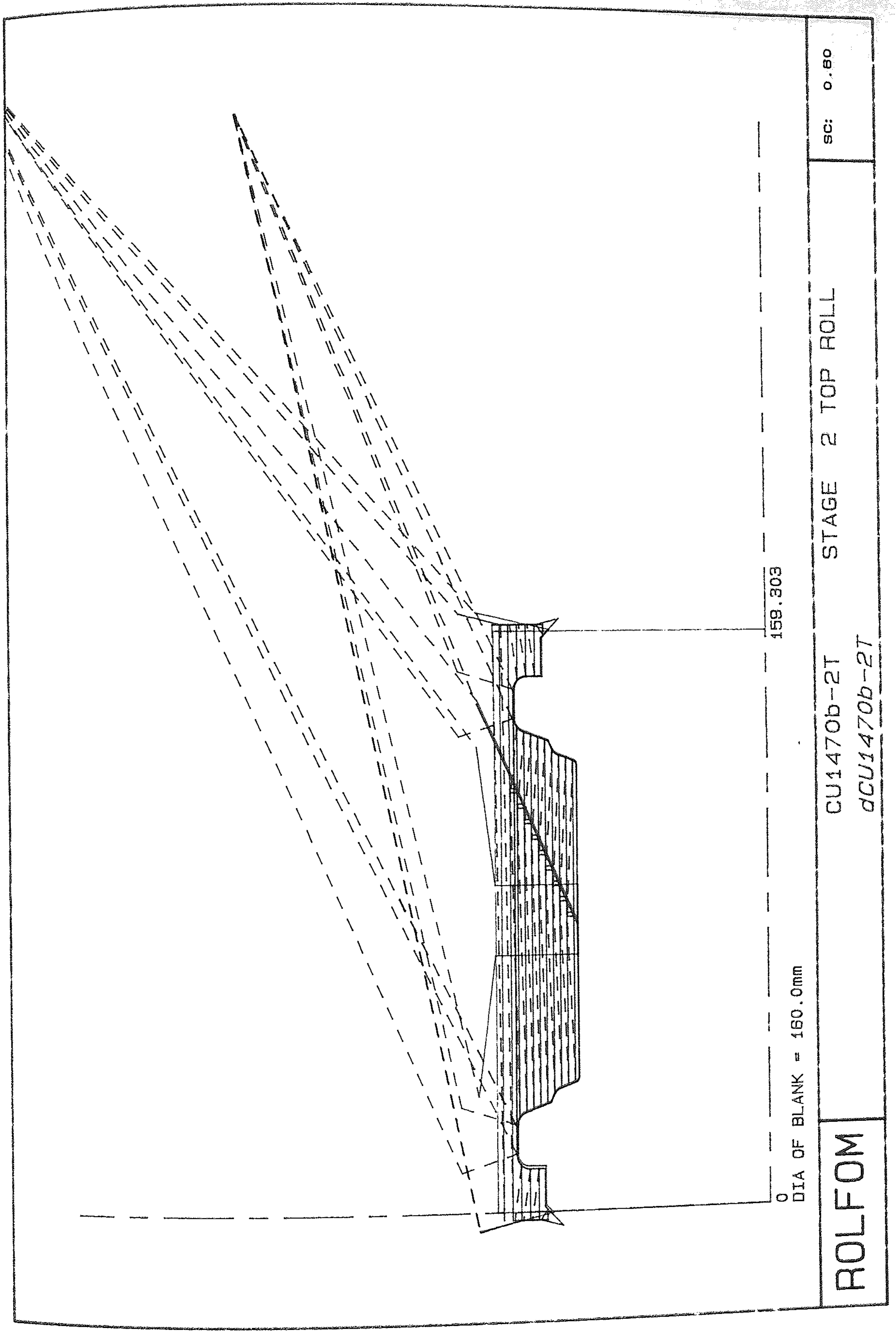


Fig. 8.30 A TOOL PATH PLOT (a) TOP ROLL

Fig. 8.31 NC PROGRAM GENERATED BY THE MFPOST POSTPROCESSOR

(a) TOP ROLL (PART 1)

```
O0201
N10G21
N15G50X-455.000Z320.000M08
N20T0202M41
N25G96S200M03
N30G0X-180.000Z164.303
N35G1X-155.000Z160.503F0.10
N40Z-2.799
N45G0X-159.000Z160.503
N50G1X-150.000
N55Z-2.800
N60G0X-154.000Z160.503
N65G1X-145.000
N70Z145.379
N75G0X-149.000Z160.503
N80G1X-140.000
N85Z146.400
N90G0X-144.000Z160.503
N95G1X-135.000
N100Z146.401
N105G0X-139.000Z160.503
N110G1X-130.000
N115Z157.644
N120G0X-134.000Z160.503
N125G1X-121.009Z162.139
N130X-127.009Z159.139
N135X-131.009Z157.139
N140G2X-131.536Z156.503I0.636K-0.636
N145G1Z146.402
N150X-139.736
N155G2X-147.536Z142.503I0.000K-3.899
N160G1Z134.606
N165Z23.099
N170Z14.200
N175G0X-455.000Z320.000
N180T0200
N185M01
N190G50X-300.000Z300.000M08
N195T0606M41
N200G96S200M03
N205G0X-170.000Z-5.000
N210G1X-145.000Z-1.199F0.25
N215Z13.403
N220G0X-149.000Z-1.200
N225G1X-140.000Z-1.199
N230Z12.900
N235G0X-144.000Z-1.200
N240G1X-135.000Z-1.199
N245Z12.900
N250G0X-139.000Z-1.200
N255G1X-130.000
N260Z1.659
N265G0X-134.000Z-1.200
N270G1X-121.009Z-2.836
N275X-127.009Z0.164
```

Fig. 8.31 (a) (PART 2)

```
N280X-131.009Z2.164
N285G3X-131.536Z2.800I0.636K0.636
N290G1Z12.900
N295X-141.736
N300G3X-147.536Z15.800I0.000K2.899
N305G1Z24.699
N310Z136.206
N315Z144.103
N320G0X-300.000Z300.000
N325T0600
N330M01
N335G50X-455.000Z320.000M08
N340T0202M41
N345G96S200M03
N350G0X-168.400Z139.498
N355G1X-146.000Z115.479
N360Z25.422
N365G0X-150.000Z115.479
N370G1X-146.000
N375X-142.000Z111.190
N380Z26.837
N385G0X-146.000Z111.190
N390G1X-142.000
N395X-138.000Z106.901
N400Z27.565
N405G0X-142.000Z106.901
N410G1X-138.000
N415X-134.000Z102.612
N420Z28.293
N425G0X-137.999Z102.612
N430G1X-134.000
N435X-130.000Z98.323
N440Z29.021
N445G0X-134.000Z98.323
N450G1X-129.999
N455X-126.000Z94.034
N460Z31.735
N465G0X-130.000Z94.034
N470G1X-125.999
N475X-122.000Z89.745
N480Z33.256
N485G0X-126.000Z89.745
N490G1X-122.000
N495X-118.000Z85.456
N500Z33.984
N505G0X-122.000Z85.456
N510G1X-118.000
N515X-114.000Z81.167
N520Z34.713
N525G0X-118.000Z81.167
N530G1X-114.000
N535X-111.976Z78.997
N540Z35.970
N545G3X-112.181Z35.505I-1.101K0.000
N550G1X-112.407Z35.263
```

Fig. 8.31 (a) (PART 3)

```
N555G3X-113.349Z34.832I-0.756K0.352
N560G1X-120.435Z33.541
N565X-122.697Z33.130
N570G2X-125.247Z32.289I1.124K-3.090
N575G1X-125.806Z31.852
N580G2X-126.636Z31.181I1.654K-1.491
N585G1X-127.995Z29.567
N590G3X-128.592Z29.278I-0.458K0.148
N595G1X-142.402Z26.764
N600G2X-147.536Z23.099I1.333K-3.665
N605G1X-148.400
N610G0X-455.000Z320.000
N615T0200
N620M01
N625G50X-300.000Z300.000M08
N630T0606M41
N635G96S200M03
N640G0X-168.400Z139.884
N645G1X-146.000Z115.865
N650Z133.882
N655G0X-150.000Z115.865
N660G1X-146.000
N665X-142.000Z111.576
N670Z132.468
N675G0X-146.000Z111.576
N680G1X-142.000
N685X-138.000Z107.287
N690Z131.739
N695G0X-141.999Z107.287
N700G1X-138.000
N705X-134.000Z102.998
N710Z131.010
N715G0X-137.999Z102.998
N720G1X-134.000
N725X-130.000Z98.709
N730Z130.282
N735G0X-134.000Z98.709
N740G1X-129.999
N745X-126.000Z94.420
N750Z127.567
N755G0X-130.000Z94.420
N760G1X-125.999
N765X-122.000Z90.131
N770Z126.046
N775G0X-126.000Z90.131
N780G1X-122.000
N785X-117.999Z85.842
N790Z125.318
N795G0X-122.000Z85.842
N800G1X-118.000
N805X-114.000Z81.553
N810Z124.589
N815G0X-118.000Z81.553
N820G1X-114.000
N825X-111.976Z79.383
```

Fig. 8.31 (a) (PART 4)

```
N830Z123.333
N835G2X-112.181Z123.798I-1.101K0.000
N840G1X-112.407Z124.040
N845G2X-113.350Z124.471I-0.756K-0.352
N850G1X-120.408Z125.756
N855X-122.699Z126.173
N860G3X-125.243Z127.013I1.125K3.089
N865G1X-125.794Z127.448
N870G3X-126.665Z128.164I1.648K1.493
N875G1X-128.001Z129.735
N880G2X-128.592Z130.025I-0.455K-0.148
N885G1X-142.404Z132.541
N890G3X-147.536Z136.206I1.334K3.665
N895G1X-148.400
N900G0X-300.000Z300.000
N905T0600
N910M01
N915G50X-455.000Z320.000M08
N920T1111M42
N925G96S300M03
N930G0X-170.000Z164.303
N935G1X-127.102Z159.186F0.10
N940X-131.102Z157.186
N945G2X-131.336Z156.903I0.283K-0.283
N950G1Z146.301
N955X-140.536
N960G2X-147.336Z142.903I0.000K-3.399
N965G1Z135.006
N970G0X-180.000Z130.406
N975G0X-455.000Z320.000
N980T1100
N985M01
N990G50X-455.000Z320.000M08
N995T1111M42
N1000G96S300M03
N1005G0X-170.000Z127.533
N1010G1X-159.200Z89.200
N1015X-111.776
N1020Z36.370
N1025G3X-112.075Z35.693I-1.601K0.000
N1030G1X-112.301Z35.451
N1035G3X-113.808Z34.762I-1.209K0.564
N1040G1X-120.891Z33.471
N1045X-123.155Z33.060
N1050G2X-125.254Z32.380I0.953K-2.620
N1055G1X-125.806Z31.949
N1060G2X-126.518Z31.387I1.254K-1.188
N1065G1X-127.879Z29.806
N1070G2X-127.896Z29.785I0.368K-0.158
N1075G3X-129.050Z29.208I-0.908K0.330
N1080G1X-142.860Z26.694
N1085G2X-147.336Z23.499I1.162K-3.195
N1090G1Z14.600
N1095G0X-180.000Z10.000
N1100G0X-455.000Z320.000
```

Fig. 8.31 (a) (PART 5)

```
N1105T1100
N1110M01
N1115G50X-300.000Z300.000M08
N1120T0303M42
N1125G96S300M03
N1130G0X-170.000Z-5.000
N1135G1X-127.102Z0.117
N1140X-131.102Z2.117
N1145G3X-131.336Z2.400I0.283K0.283
N1150G1Z13.000
N1155X-142.536
N1160G3X-147.336Z15.399I0.000K2.400
N1165G1Z24.299
N1170G0X-179.999Z28.899
N1175G0X-300.000Z300.000
N1180T0300
N1185M01
N1190G50X-300.000Z300.000M08
N1195T0303M42
N1200G96S300M03
N1205G0X-170.000Z31.770
N1210G1X-159.200Z70.800
N1215X-111.776
N1220Z122.933
N1225G2X-112.075Z123.610I-1.601K0.000
N1230G1X-112.301Z123.852
N1235G2X-113.808Z124.541I-1.209K-0.563
N1240G1X-120.949Z125.841
N1245X-123.156Z126.243
N1250G3X-125.259Z126.928I0.953K2.620
N1255G1X-125.804Z127.358
N1260G3X-126.517Z127.948I1.253K1.182
N1265G1X-127.878Z129.495
N1270G3X-127.895Z129.517I0.367K0.159
N1275G2X-129.050Z130.095I-0.908K-0.330
N1280G1X-142.862Z132.611
N1285G3X-147.336Z135.806I1.163K3.194
N1290G1Z143.703
N1295G0X-180.000Z148.303
N1300G0X-300.000Z300.000
N1305T0300
N1310M30
```

Fig. 8.31 NC PROGRAM GENERATED BY THE MFPOST POSTPROCESSOR

(b) BOTTOM ROLL (PART 1)

```
O0202
N10G21
N15G50X-455.000Z320.000M08
N20T0202M41
N25G96S200M03
N30G0X-180.000Z164.303
N35G1X-155.000Z160.503F0.25
N40Z-2.799
N45G0X-159.000Z160.503
N50G1X-150.000
N55Z-2.800
N60G0X-154.000Z160.503
N65G1X-145.000
N70Z124.105
N75G0X-149.000Z160.503
N80G1X-140.000
N85Z125.014
N90G0X-144.000Z160.503
N95G1X-135.000
N100Z126.064
N105G0X-139.000Z160.503
N110G1X-130.000
N115Z157.308
N120G0X-134.000Z160.503
N125G1X-125.000
N130Z159.389
N135G0X-129.000Z160.503
N140G1X-118.426Z162.403
N145X-124.428Z159.403
N150G2X-130.230Z156.503I0.000K-2.900
N155G1Z147.503
N160Z129.572
N165X-130.387Z129.544
N170G2X-132.368Z128.553I0.566K-1.557
N175G1X-133.564Z126.908
N180G3X-134.049Z126.529I-0.974K0.354
N185G1X-134.751Z126.177
N190G3X-136.283Z125.692I-1.482K1.486
N195G1X-138.534Z125.281
N200X-145.631Z123.990
N205G2X-148.790Z121.733I0.822K-2.258
N210G1Z35.970
N215G0X-455.000Z320.000
N220T0200
N225M01
N230G50X-300.000Z300.000M08
N235T0606M41
N240G96S200M03
N245G0X-170.000Z-5.000
N250G1X-145.000Z-1.200
N255Z35.198
N260G0X-149.000Z-1.200
N265G1X-140.000
N270Z34.289
N275G0X-144.000Z-1.200
```

Fig. 8.31 (b) (PART 2)

```
N280G1X-135.000
N285Z33.239
N290G0X-139.000Z-1.200
N295G1X-130.000
N300Z1.995
N305G0X-134.000Z-1.200
N310G1X-125.000
N315Z-0.086
N320G0X-129.000Z-1.200
N325G1X-118.426Z-3.100
N330X-124.428Z-0.100
N335G3X-130.230Z2.799I0.000K2.900
N340G1Z11.800
N345Z29.731
N350X-130.388Z29.759
N355G3X-132.368Z30.750I0.567K1.556
N360G1X-133.564Z32.395
N365G2X-134.047Z32.773I-0.974K-0.356
N370G1X-134.749Z33.124
N375G2X-136.284Z33.612I-1.483K-1.484
N380G1X-145.632Z35.313
N385G3X-148.790Z37.570I0.821K2.257
N390G1Z123.333
N395G0X-300.000Z300.000
N400T0600
N405M01
N410G50X-455.000Z320.000M08
N415T0808M41
N420G96S100M03
N425G0X-170.000Z153.303
N430G1X-156.000Z144.203F0.10
N435X-130.230
N440G0X-155.999
N445G1Z142.153
N450X-126.938
N455G0X-156.000
N460G1Z140.103
N465X-114.230
N470G0X-155.999
N475G1Z138.053
N480X-114.230
N485G0X-156.000
N490G1Z136.003
N495X-114.230
N500G0X-156.000
N505G1Z133.953
N510X-114.230
N515G0X-155.999
N520G1Z131.903
N525X-122.308
N530G0X-156.000
N535G1Z129.852
N540X-130.004
N545G0X-156.000
N550G1Z127.803
```

Fig. 8.31 (b) (PART 3)

```
N555X-133.393
N560G0X-155.999
N565G1Z126.765
N570X-134.516
N575G0X-156.000
N580G0X-455.000Z320.000
N585T0800
N590M01
N595G50X-455.000Z320.000M08
N600T0808M41
N605G96S100M03
N610G0X-170.000Z6.000
N615G1X-156.000Z11.000
N620X-130.230
N625G0X-155.999
N630G1Z13.050
N635X-126.939
N640G0X-156.000
N645G1Z15.100
N650X-114.230
N655G0X-156.000
N660G1Z17.150
N665X-114.230
N670G0X-156.000
N675G1Z19.200
N680X-114.230
N685G0X-156.000
N690G1Z21.250
N695X-114.230
N700G0X-155.999
N705G1Z23.300
N710X-122.308
N715G0X-156.000
N720G1Z25.349
N725X-130.007
N730G0X-155.999Z25.350
N735G1Z27.400
N740X-133.393Z27.399
N745G0X-156.000
N750G1Z28.438
N755X-134.517
N760G0X-155.999
N765G0X-455.000Z320.000
N770T0800
N775M01
N780G50X-455.000Z320.000M08
N785T1111M42
N790G96S300M03
N795G0X-170.000Z151.303
N800G1X-159.200Z137.700
N805X-114.030
N810Z133.183
N815X-128.538Z130.543
N820G2X-129.016Z130.031I0.137K-0.376
N825G1X-128.872Z129.832
```

Fig. 8.31 (b) (PART 4)

```
N830X-130.845Z129.474
N835G2X-132.228Z128.783I0.396K-1.087
N840G1X-133.424Z127.138
N845G3X-134.141Z126.576I-1.444K0.525
N850G1X-134.843Z126.224
N855G3X-136.740Z125.622I-1.837K1.838
N860G1X-139.000Z125.210
N865X-146.090Z123.920
N870G2X-148.589Z122.133I0.651K-1.787
N875G1Z36.370
N880G0X-179.999Z31.770
N885G0X-455.000Z320.000
N890T1100
N895M01
N900G50X-455.000Z320.000M08
N905T1111M42
N910G96S300M03
N915G0X-170.000Z31.000
N920G1X-159.200Z20.200
N925X-114.030
N930Z13.000
N935X-125.228
N940G2X-130.030Z10.600I0.000K-2.399
N945G1Z1.600
N950G2X-125.227Z-0.800I2.401K0.000
N955G0X-180.000Z-5.000
N960G0X-455.000Z320.000
N965T1100
N970M01
N975G50X-300.000Z300.000M08
N980T0303M42
N985G96S300M03
N990G0X-170.000Z8.000
N995G1X-159.200Z20.800
N1000X-114.030
N1005Z26.120
N1010X-128.538Z28.760
N1015G3X-129.016Z29.272I0.137K0.376
N1020G1X-128.873Z29.470
N1025X-130.846Z29.829
N1030G3X-132.228Z30.520I0.396K1.086
N1035G1X-133.424Z32.165
N1040G2X-134.140Z32.726I-1.444K-0.526
N1045G1X-134.842Z33.077
N1050G2X-136.742Z33.682I-1.837K-1.837
N1055G1X-146.090Z35.383
N1060G3X-148.590Z37.170I0.650K1.787
N1065G1Z122.933
N1070G0X-180.000Z127.533
N1075G0X-300.000Z300.000
N1080T0300
N1085M01
N1090G50X-300.000Z300.000M08
N1095T0303M42
N1100G96S300M03
```

Fig. 8.31 (b) (PART 5)

```
N1105G0X-170.000Z128.303
N1110G1X-159.200Z138.300
N1115X-114.030
N1120Z146.303
N1125X-125.228
N1130G3X-130.030Z148.703I0.000K2.399
N1135G1Z157.703
N1140G3X-125.228Z160.103I2.401K0.000
N1145G0X-180.000Z164.303
N1150G0X-300.000Z300.000
N1155T0300
N1160M30
```

Fig. 8.32 TERMINAL LISTING FOR THE CUTPLOT SOFTWARE

(a) TOP ROLL (PART 1)

```
                         ****** ROLFOM ******

               ** CUTTER LOCATION DATA PROGRAM **

SECTION NO. ?
14706


 PLEASE INPUT THE STAGE NO AND THE ROLLER NO. OF THE ROLL THAT REQUIRES NC DATA
NSTAGE ICON
  2     1

 INPUT THE CYCLE TYPE(ICYCLE), START EL. NO.(ISEL), ENDING EL. NO.(IEEL),
  AND THE STOCK LEFT FOR FINISHING (TOL)'
  (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
  ( FOR FINAL FINISHING, INPUT TOL=0.0)
  (++ INPUT    0       0       0       0 FOR TERMINATING INPUT ++)
PLEASE INPUT THE FOLLOWING:
CYCLE    STARTING    ENDING    STOCK
 TYPE    ELEMENT     ELEMENT    LEFT

  1         38          2       0.100

PLEASE INPUT THE TOOL NO ,DEPTH OF CUT,SPEED AND FEED
(CHOOSE FROM TOOL NOS 1-12 FOR TOOL IN TOOL LIBRARY,
ENTER 0 FOR TOOL NOT IN LIBRARY,
ENTER 0 FOR DEPTH OF CUT FOR FINISHING OR GROOVING CYCLE)


TOOL     DEPTH OF       SPEED     FEED
 NO.        CUT
  2        2.50         200       0.100

 INPUT THE CYCLE TYPE(ICYCLE), START EL. NO.(ISEL), ENDING EL. NO.(IEEL),
  AND THE STOCK LEFT FOR FINISHING (TOL)'
  (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
  ( FOR FINAL FINISHING, INPUT TOL=0.0)
  (++ INPUT    0       0       0       0 FOR TERMINATING INPUT ++)
PLEASE INPUT THE FOLLOWING:
CYCLE    STARTING    ENDING    STOCK
 TYPE    ELEMENT     ELEMENT    LEFT

  1         2          38       0.100

PLEASE INPUT THE TOOL NO ,DEPTH OF CUT,SPEED AND FEED
(CHOOSE FROM TOOL NOS 1-12 FOR TOOL IN TOOL LIBRARY,
ENTER 0 FOR TOOL NOT IN LIBRARY,
ENTER 0 FOR DEPTH OF CUT FOR FINISHING OR GROOVING CYCLE)

TOOL     DEPTH OF       SPEED     FEED
 NO.        CUT
  6        2.50         200       0.250

INPUT THE CYCLE TYPE(ICYCLE), START EL. NO.(ISEL), ENDING EL. NO.(IEEL),
  AND THE STOCK LEFT FOR FINISHING (TOL)'
  (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
  ( FOR FINAL FINISHING, INPUT TOL=0.0)
  (++ INPUT    0       0       0       0 FOR TERMINATING INPUT ++)
```

Fig. 8.32 (a) (PART 2)

```
PLEASE INPUT THE TOOL NO ,DEPTH OF CUT,SPEED AND FEED
(CHOOSE FROM TOOL NOS 1-12 FOR TOOL IN TOOL LIBRARY,
ENTER 0 FOR TOOL NOT IN LIBRARY,
ENTER 0 FOR DEPTH OF CUT FOR FINISHING OR GROOVING CYCLE)

TOOL     DEPTH OF        SPEED      FEED
 NO.        CUT
 11        0.00          300       0.100

++ DO YOU WANT TO SPECIFY A STARTING POINT? ++
         (INPUT 1 FOR YES,0 FOR NO)
   0

 INPUT THE CYCLE TYPE(ICYCLE), START EL. NO.(ISEL), ENDING EL. NO.(IEEL),
  AND THE STOCK LEFT FOR FINISHING (TOL)'
  (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
  ( FOR FINAL FINISHING, INPUT TOL=0.0)
  (++ INPUT     0       0        0       0 FOR TERMINATING INPUT ++)
PLEASE INPUT THE FOLLOWING:
CYCLE     STARTING    ENDING    STOCK
 TYPE     ELEMENT     ELEMENT    LEFT

  3         20           6       0.000

PLEASE INPUT THE TOOL NO ,DEPTH OF CUT,SPEED AND FEED
(CHOOSE FROM TOOL NOS 1-12 FOR TOOL IN TOOL LIBRARY,
ENTER 0 FOR TOOL NOT IN LIBRARY,
ENTER 0 FOR DEPTH OF CUT FOR FINISHING OR GROOVING CYCLE)

TOOL     DEPTH OF        SPEED      FEED
 NO.        CUT
 11        0.00          300       0.100

++ DO YOU WANT TO SPECIFY A STARTING POINT? ++
         (INPUT 1 FOR YES,0 FOR NO)
   1
++ PLEASE DEFINE THE STARTING POINT AND THE CUT-IN LINE ++
NOTE: THE CUT-IN LINE MUST INTERSECT WITH THE STARTING ELEMENT
LET THE STARTING PT BE (ZP,XP), AND THE ANGLE OF THE CUT-IN LINE BE AND
PLEASE INPUT ZP,XP & ANG
    ZP        XP       ANG
   90.0      80.0      90.0

 INPUT THE CYCLE TYPE(ICYCLE), START EL. NO.(ISEL), ENDING EL. NO.(IEEL),
  AND THE STOCK LEFT FOR FINISHING (TOL)'
  (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
  ( FOR FINAL FINISHING, INPUT TOL=0.0)
  (++ INPUT     0       0        0       0 FOR TERMINATING INPUT ++)
 PLEASE INPUT THE FOLLOWING:
CYCLE     STARTING    ENDING    STOCK
 TYPE     ELEMENT     ELEMENT    LEFT

  3         2            6       0.000

 PLEASE INPUT THE TOOL NO ,DEPTH OF CUT,SPEED AND FEED
(CHOOSE FROM TOOL NOS 1-12 FOR TOOL IN TOOL LIBRARY,
 ENTER 0 FOR TOOL NOT IN LIBRARY,
 ENTER 0 FOR DEPTH OF CUT FOR FINISHING OR GROOVING CYCLE)
```

Fig. 8.32 (a) (PART 3)

```
PLEASE INPUT THE FOLLOWING:
CYCLE     STARTING    ENDING    STOCK
 TYPE     ELEMENT     ELEMENT    LEFT

  4         20           7       0.100

 PLEASE INPUT THE TOOL NO ,DEPTH OF CUT,SPEED AND FEED
(CHOOSE FROM TOOL NOS 1-12 FOR TOOL IN TOOL LIBRARY,
ENTER 0 FOR TOOL NOT IN LIBRARY,
ENTER 0 FOR DEPTH OF CUT FOR FINISHING OR GROOVING CYCLE)

TOOL      DEPTH OF       SPEED     FEED
 NO.        CUT
 2          2.00          200      0.250

++ PLEASE DEFINE THE STARTING POINT AND THE CUT-IN LINE ++
NOTE: THE CUT-IN LINE MUST INTERSECT WITH THE STARTING ELEMENT
LET THE STARTING PT BE (ZP,XP), AND THE ANGLE OF THE CUT-IN LINE BE AND
PLEASE INPUT ZP,XP & ANG
    ZP         XP       ANG
   80.0       56.0      25.0

INPUT THE CYCLE TYPE(ICYCLE), START EL. NO.(ISEL), ENDING EL. NO.(IEEL),
  AND THE STOCK LEFT FOR FINISHING (TOL)'
  (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
  ( FOR FINAL FINISHING, INPUT TOL=0.0)
  (++ INPUT    0       0       0       0 FOR TERMINATING INPUT ++)
PLEASE INPUT THE FOLLOWING:
CYCLE     STARTING    ENDING    STOCK
 TYPE     ELEMENT     ELEMENT    LEFT

 4          20          33       0.100

 PLEASE INPUT THE TOOL NO ,DEPTH OF CUT,SPEED AND FEED
(CHOOSE FROM TOOL NOS 1-12 FOR TOOL IN TOOL LIBRARY,
 ENTER 0 FOR TOOL NOT IN LIBRARY,
 ENTER 0 FOR DEPTH OF CUT FOR FINISHING OR GROOVING CYCLE)

TOOL      DEPTH OF       SPEED     FEED
 NO.        CUT
  6        2.00           200      0.250

++ PLEASE DEFINE THE STARTING POINT AND THE CUT-IN LINE ++
NOTE: THE CUT-IN LINE MUST INTERSECT WITH THE STARTING ELEMENT
LET THE STARTING PT BE (ZP,XP), AND THE ANGLE OF THE CUT-IN LINE BE AND
PLEASE INPUT ZP,XP & ANG
    ZP         XP       ANG
    75.0       56.0      25.0

 INPUT THE CYCLE TYPE(ICYCLE), START EL. NO.(ISEL), ENDING EL. NO.(IEEL),
  AND THE STOCK LEFT FOR FINISHING (TOL)'
  (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
  ( FOR FINAL FINISHING, INPUT TOL=0.0)
  (++ INPUT    0       0       0       0 FOR TERMINATING INPUT ++)
PLEASE INPUT THE FOLLOWING:
CYCLE     STARTING    ENDING    STOCK
 TYPE     ELEMENT     ELEMENT    LEFT

 3          38          34       0.000
```

Fig. 8.32 (a) (PART 4)

```
TOOL      DEPTH OF        SPEED      FEED
 NO.         CUT
 3          0.00           300        0.100

++ DO YOU WANT TO SPECIFY A STARTING POINT? ++
          (INPUT 1 FOR YES,0 FOR NO)
   0

 INPUT THE CYCLE TYPE(ICYCLE), START EL. NO.(ISEL), ENDING EL. NO.(IEEL),
  AND THE STOCK LEFT FOR FINISHING (TOL)'
  (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
  ( FOR FINAL FINISHING, INPUT TOL=0.0)
  (++ INPUT    0        0        0       0 FOR TERMINATING INPUT ++)
PLEASE INPUT THE FOLLOWING:
CYCLE    STARTING    ENDING    STOCK
 TYPE    ELEMENT     ELEMENT    LEFT

  3         20          34      0.000

PLEASE INPUT THE TOOL NO ,DEPTH OF CUT,SPEED AND FEED
 (CHOOSE FROM TOOL NOS 1-12 FOR TOOL IN TOOL LIBRARY,
 ENTER 0 FOR TOOL NOT IN LIBRARY,
 ENTER 0 FOR DEPTH OF CUT FOR FINISHING OR GROOVING CYCLE)

TOOL      DEPTH OF        SPEED      FEED
 NO.         CUT
  3         0.00           300        0.100

++ DO YOU WANT TO SPECIFY A STARTING POINT? ++
          (INPUT 1 FOR YES,0 FOR NO)
   1
++ PLEASE DEFINE THE STARTING POINT AND THE CUT-IN LINE ++
NOTE: THE CUT-IN LINE MUST INTERSECT WITH THE STARTING ELEMENT
LET THE STARTING PT BE (ZP,XP), AND THE ANGLE OF THE CUT-IN LINE BE AND
PLEASE INPUT ZP,XP & ANG
    ZP        XP        ANG
   70.0      80.0      90.0

  INPUT THE CYCLE TYPE(ICYCLE), START EL. NO.(ISEL), ENDING EL. NO.(IEEL),
   AND THE STOCK LEFT FOR FINISHING (TOL)'
   (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
   ( FOR FINAL FINISHING, INPUT TOL=0.0)
   (++ INPUT    0        0        0       0 FOR TERMINATING INPUT ++)
PLEASE INPUT THE FOLLOWING:
CYCLE    STARTING    ENDING    STOCK
 TYPE    ELEMENT     ELEMENT    LEFT

   0        0          0        0.000


              **** END OF INPUT FILE ****
```

Fig. 8.32 TERMINAL LISTING FOR THE CUTPLOT SOFTWARE

(b) BOTTOM ROLL (PART 1)

```
                    ****** ROLFOM ******


              ** CUTTER LOCATION DATA PROGRAM **


 SECTION NO. ?
  1470b


PLEASE INPUT THE STAGE NO AND THE ROLLER NO. OF THE ROLL THAT REQUIRES NC DATA
   NSTAGE ICON
      2     2


   INPUT THE CYCLE TYPE(ICYCLE), START EL. NO.(ISEL), ENDING EL.NO.(IEEL),
    AND THE STOCK LEFT FOR FINISHING (TOL)
    (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
    ( FOR FINAL FINISHING, INPUT TOL=0.0)
    (++ INPUT    0        0        0       0 FOR TERMINATING INPUT ++)
        PLEASE INPUT THE FOLLOWING:
          CYCLE    STARTING    ENDING    STOCK
           TYPE    ELEMENT     ELEMENT    LEFT
            1        35          2         0.1


        PLEASE INPUT THE TOOL NO ,DEPTH OF CUT,SPEED AND FEED
        (CHOOSE FROM TOOL NOS 1-12 FOR TOOL IN TOOL LIBRARY,
        ENTER 0 FOR TOOL NOT IN LIBRARY,
        ENTER 0 FOR DEPTH OF CUT FOR FINISHING OR GROOVING CYCLE)
           TOOL      DEPTH OF       SPEED     FEED
            NO.         CUT
             2        2.50          200       0.250


  INPUT THE CYCLE TYPE(ICYCLE), START EL.NO.(ISEL), ENDING EL. NO.(IEEL),
   AND THE STOCK LEFT FOR FINISHING (TOL),
   (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
   ( FOR FINAL FINISHING, INPUT TOL=0.0)
   (++ INPUT    0        0        0        0 FOR TERMINATING INPUT ++)
        PLEASE INPUT THE FOLLOWING:
          CYCLE    STARTING    ENDING    STOCK
           TYPE    ELEMENT     ELEMENT    LEFT
            1        2           35        0.100


        PLEASE INPUT THE TOOL NO ,DEPTH OF CUT,SPEED AND FEED
        (CHOOSE FROM TOOL NOS 1-12 FOR TOOL IN TOOL LIBRARY,
        ENTER 0 FOR TOOL NOT IN LIBRARY,
        ENTER 0 FOR DEPTH OF CUT FOR FINISHING OR GROOVING CYCLE)
           TOOL      DEPTH OF       SPEED     FEED
            NO.         CUT
             6        2.50          200       0.250
```

Fig. 8.32 (b) (PART 2)

```
INPUT THE CYCLE TYPE(ICYCLE), START EL.NO.(ISEL), ENDING EL. NO.(IEEL),
 AND THE STOCK LEFT FOR FINISHING (TOL),
 (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
 ( FOR FINAL FINISHING, INPUT TOL=0.0)
 (++ INPUT    0       0       0       0 FOR TERMINATING INPUT ++)
      PLEASE INPUT THE FOLLOWING:
        CYCLE     STARTING    ENDING    STOCK
         TYPE     ELEMENT     ELEMENT    LEFT
          2          33          25      0.100


      PLEASE INPUT THE TOOL NO ,DEPTH OF CUT,SPEED AND FEED
      (CHOOSE FROM TOOL NOS 1-12 FOR TOOL IN TOOL LIBRARY,
      ENTER 0 FOR TOOL NOT IN LIBRARY,
      ENTER 0 FOR DEPTH OF CUT FOR FINISHING OR GROOVING CYCLE)
         TOOL      DEPTH OF        SPEED     FEED
          NO.         CUT
          8          0.00          100       0.100


INPUT THE CYCLE TYPE(ICYCLE), START EL.NO.(ISEL), ENDING EL. NO.(IEEL),
 AND THE STOCK LEFT FOR FINISHING (TOL),
 (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
 ( FOR FINAL FINISHING, INPUT TOL=0.0)
 (++ INPUT    0       0       0       0 FOR TERMINATING INPUT ++)
      PLEASE INPUT THE FOLLOWING:
        CYCLE     STARTING    ENDING    STOCK
         TYPE     ELEMENT     ELEMENT    LEFT
          2          4           12      0.100


      PLEASE INPUT THE TOOL NO ,DEPTH OF CUT,SPEED AND FEED
      (CHOOSE FROM TOOL NOS 1-12 FOR TOOL IN TOOL LIBRARY,
      ENTER 0 FOR TOOL NOT IN LIBRARY,
      ENTER 0 FOR DEPTH OF CUT FOR FINISHING OR GROOVING CYCLE)
         TOOL      DEPTH OF        SPEED     FEED
          NO.         CUT
          8          0.00          100       0.100


INPUT THE CYCLE TYPE(ICYCLE), START EL.NO.(ISEL), ENDING EL. NO.(IEEL),
 AND THE STOCK LEFT FOR FINISHING (TOL),
 (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
 ( FOR FINAL FINISHING, INPUT TOL=0.0)
 (++ INPUT    0       0       0       0 FOR TERMINATING INPUT ++)
      PLEASE INPUT THE FOLLOWING:
        CYCLE     STARTING    ENDING    STOCK
         TYPE     ELEMENT     ELEMENT    LEFT
          3          31          18      0.000


      PLEASE INPUT THE TOOL NO ,DEPTH OF CUT,SPEED AND FEED
      (CHOOSE FROM TOOL NOS 1-12 FOR TOOL IN TOOL LIBRARY,
      ENTER 0 FOR TOOL NOT IN LIBRARY,
      ENTER 0 FOR DEPTH OF CUT FOR FINISHING OR GROOVING CYCLE)
```

Fig. 8.32 (b) (PART 3)

```
        TOOL      DEPTH OF       SPEED     FEED
         NO.        CUT
         11         0.00          300       0.100


++ DO YOU WANT TO SPECIFY A STARTING POINT? ++
       (INPUT 1 FOR YES,0 FOR NO)
   1
++ PLEASE DEFINE THE STARTING POINT AND THE CUT-IN LINE ++
NOTE: THE CUT-IN LINE MUST INTERSECT WITH THE STARTING ELEMENT
LET THE STARTING PT BE (ZP,XP), AND THE ANGLE OF THE CUT-IN LINE BE AND
PLEASE INPUT ZP,XP & ANG
    ZP        XP       ANG
   138.5     80.0      90.0



   INPUT THE CYCLE TYPE(ICYCLE), START EL.NO.(ISEL), ENDING EL. NO.(IEEL),
    AND THE STOCK LEFT FOR FINISHING (TOL),
    (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
    ( FOR FINAL FINISHING, INPUT TOL=0.0)
    (++ INPUT    0        0        0        0 FOR TERMINATING INPUT ++)
         PLEASE INPUT THE FOLLOWING:
           CYCLE     STARTING    ENDING    STOCK
            TYPE     ELEMENT     ELEMENT    LEFT
             3          6           2       0.000


         PLEASE INPUT THE TOOL NO ,DEPTH OF CUT,SPEED AND FEED
         (CHOOSE FROM TOOL NOS 1-12 FOR TOOL IN TOOL LIBRARY,
         ENTER 0 FOR TOOL NOT IN LIBRARY,
         ENTER 0 FOR DEPTH OF CUT FOR FINISHING OR GROOVING CYCLE)
            TOOL      DEPTH OF       SPEED      FEED
             NO.        CUT
             11         0.00          300       0.100


++ DO YOU WANT TO SPECIFY A STARTING POINT? ++
       (INPUT 1 FOR YES,0 FOR NO)
   1
++ PLEASE DEFINE THE STARTING POINT AND THE CUT-IN LINE ++
NOTE: THE CUT-IN LINE MUST INTERSECT WITH THE STARTING ELEMENT
LET THE STARTING PT BE (ZP,XP), AND THE ANGLE OF THE CUT-IN LINE BE AND
PLEASE INPUT ZP,XP & ANG
    ZP        XP       ANG
  21.0       80.0      90.0

   INPUT THE CYCLE TYPE(ICYCLE), START EL.NO.(ISEL), ENDING EL. NO.(IEEL),
    AND THE STOCK LEFT FOR FINISHING (TOL),
    (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
    ( FOR FINAL FINISHING, INPUT TOL=0.0)
    (++ INPUT    0        0        0        0 FOR TERMINATING INPUT ++)
```

Fig. 8.32 (b) (PART 4)

```
        PLEASE INPUT THE FOLLOWING:
          CYCLE    STARTING    ENDING    STOCK
           TYPE    ELEMENT     ELEMENT    LEFT
            3         6          18      0.000
        PLEASE INPUT THE TOOL NO ,DEPTH OF CUT,SPEED AND FEED
        (CHOOSE FROM TOOL NOS 1-12 FOR TOOL IN TOOL LIBRARY,
        ENTER 0 FOR TOOL NOT IN LIBRARY,
        ENTER 0 FOR DEPTH OF CUT FOR FINISHING OR GROOVING CYCLE)
           TOOL      DEPTH OF        SPEED     FEED
            NO.         CUT
             3        0.00          300       0.100
++ DO YOU WANT TO SPECIFY A STARTING POINT? ++
        (INPUT 1 FOR YES,0 FOR NO)
   1
++ PLEASE DEFINE THE STARTING POINT AND THE CUT-IN LINE ++
NOTE: THE CUT-IN LINE MUST INTERSECT WITH THE STARTING ELEMENT
LET THE STARTING PT BE (ZP,XP), AND THE ANGLE OF THE CUT-IN LINE BE AND
PLEASE INPUT ZP,XP & ANG
    ZP        XP       ANG
  20.0       80.0     90.0


   INPUT THE CYCLE TYPE(ICYCLE), START EL.NO.(ISEL), ENDING EL. NO.(IEEL),
    AND THE STOCK LEFT FOR FINISHING (TOL),
    (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
    ( FOR FINAL FINISHING, INPUT TOL=0.0)
    (++ INPUT    0        0        0        0 FOR TERMINATING INPUT ++)
        PLEASE INPUT THE FOLLOWING:
          CYCLE    STARTING    ENDING    STOCK
           TYPE    ELEMENT     ELEMENT    LEFT
            3         31         35      0.000


        PLEASE INPUT THE TOOL NO ,DEPTH OF CUT,SPEED AND FEED
        (CHOOSE FROM TOOL NOS 1-12 FOR TOOL IN TOOL LIBRARY,
        ENTER 0 FOR TOOL NOT IN LIBRARY,
        ENTER 0 FOR DEPTH OF CUT FOR FINISHING OR GROOVING CYCLE)
           TOOL      DEPTH OF        SPEED     FEED
            NO.         CUT
             3        0.00          300       0.100


++ DO YOU WANT TO SPECIFY A STARTING POINT? ++
        (INPUT 1 FOR YES,0 FOR NO)
   1
++ PLEASE DEFINE THE STARTING POINT AND THE CUT-IN LINE ++
NOTE: THE CUT-IN LINE MUST INTERSECT WITH THE STARTING ELEMENT
LET THE STARTING PT BE (ZP,XP), AND THE ANGLE OF THE CUT-IN LINE BE AND
PLEASE INPUT ZP,XP & ANG
    ZP        XP       ANG
  139.0      80.0     90.0
```

Fig. 8.32 (b) (PART 5)

```
INPUT THE CYCLE TYPE(ICYCLE), START EL.NO.(ISEL), ENDING EL. NO.(IEEL),
 AND THE STOCK LEFT FOR FINISHING (TOL),
 (ICYCLE=1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISHING, 4 FOR POCKETING)
 ( FOR FINAL FINISHING, INPUT TOL=0.0)
 (++ INPUT    0        0        0        0 FOR TERMINATING INPUT ++)
      PLEASE INPUT THE FOLLOWING:
        CYCLE     STARTING    ENDING     STOCK
         TYPE     ELEMENT     ELEMENT     LEFT
          0          0           0       0.000


                **** END OF INPUT FILE ****
```

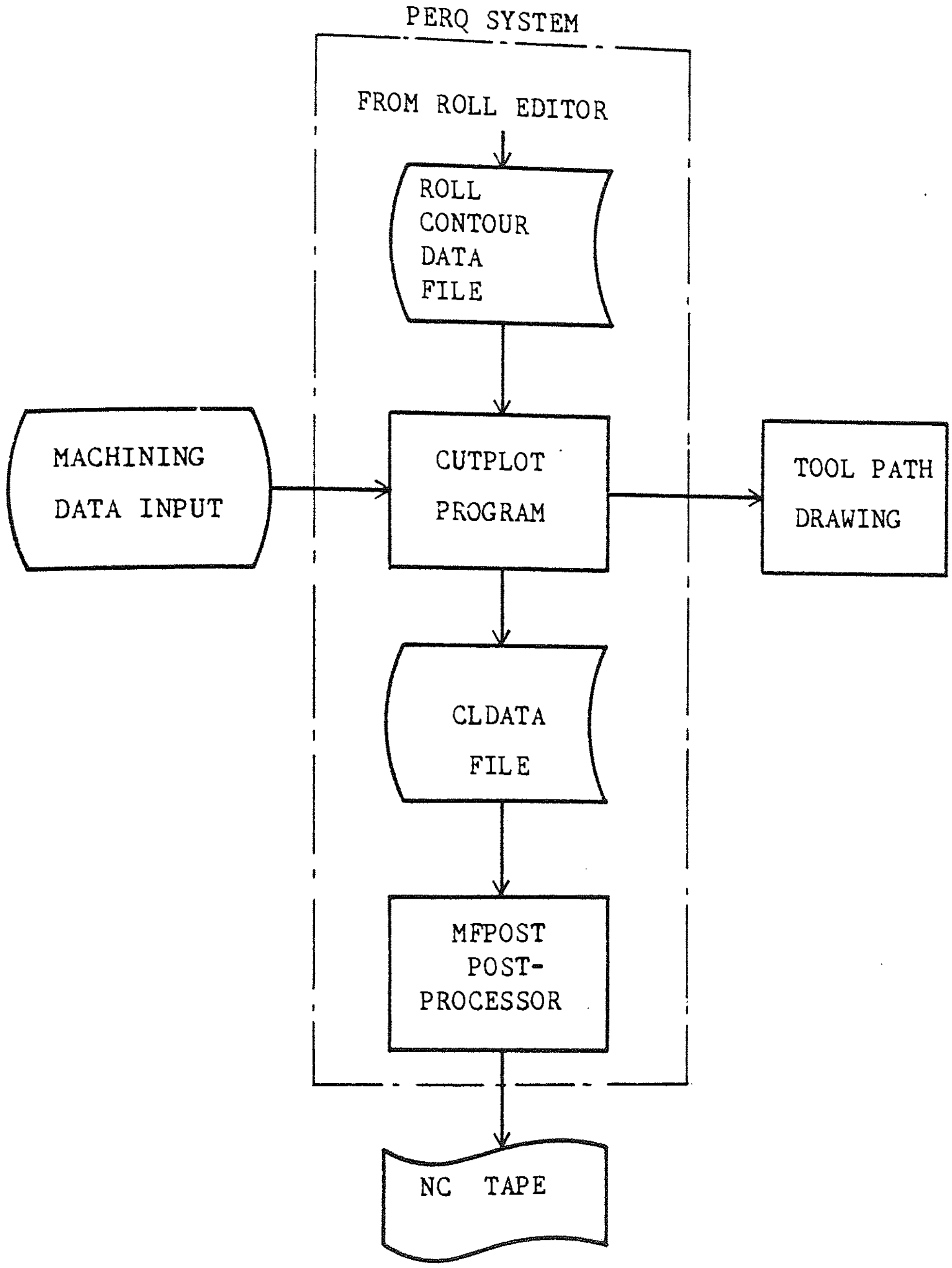


Fig. 8.33 PROCEDURE FOR PREPARING NC TAPE WITH CUTPLOT AND MFPOST PROGRAMS

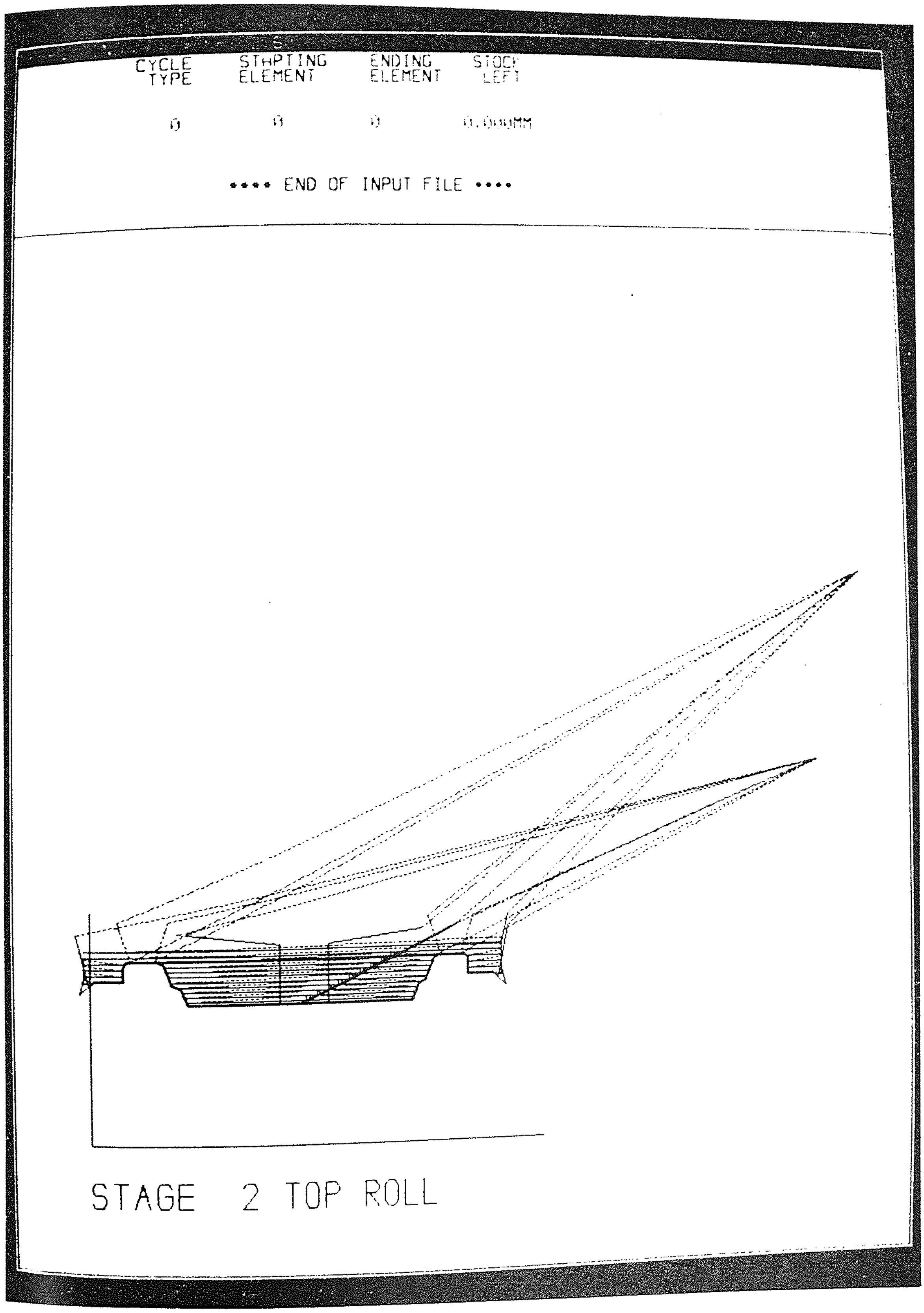


Plate 8.1 PERQ SCREEN DISPLAY OF A TYPICAL TOOL PATH PLOT GENERATED BY THE CUTPLOT PROGRAM

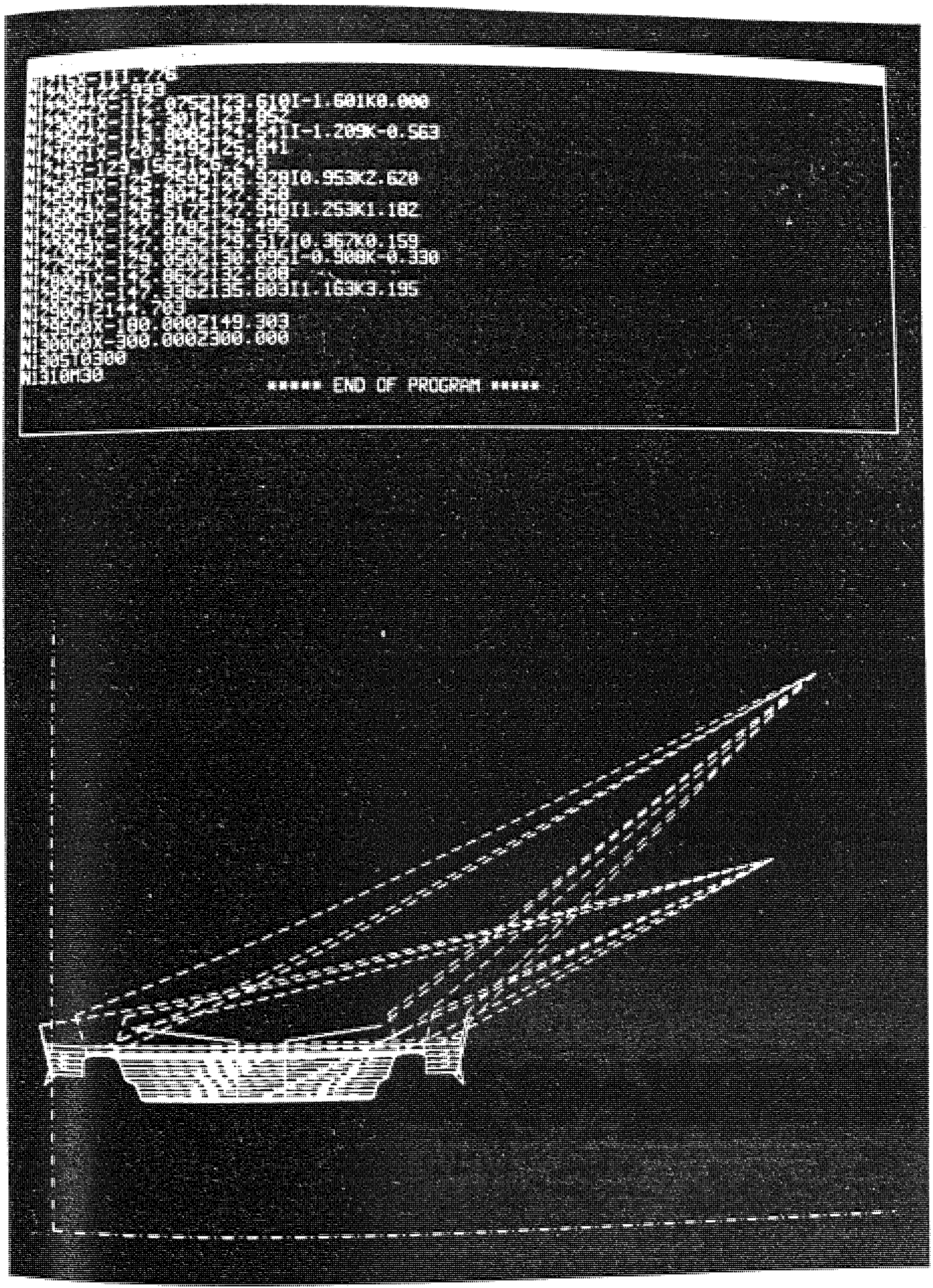


Plate 8.2 PERQ SCREEN DISPLAY OF A TYPICAL TOOL PATH PLOT GENERATED BY THE MFPOST PROCESSOR

Plate 8.3 A SET OF FORM-ROLLS PRODUCED BY CNC

Table 8.1 DATA STORED IN THE TLIB TOOL LIBRARY

TURRET POSITION	RAD	RZ	RX	WID	ZREF	XREF
1	2.5	0.0	-2.5	0.0	320.0	-455.0
2	0.8	0.8	-0.8	0.0	320.0	-455.0
3	0.4	-0.4	-0.4	0.0	300.0	-300.0
4*	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	6.0	320.0	-455.0
6	0.8	-0.8	-0.8	0.0	300.0	-300.0
7	0.8	0.8	-0.8	0.0	320.0	-455.0
8	0.0	0.0	0.0	4.1	320.0	-455.0
9*	0.0	0.0	0.0	0.0	0.0	0.0
10	0.8	-0.8	-0.8	0.0	300.0	-300.0
11	0.4	0.4	-0.4	0.0	320.0	-455.0
12	0.8	0.0	-0.8	0.0	320.0	-455.0

Notes :- (1) All dimensions are in mm

(2) * refers to turret position not in use

(3) RAD = tool tip nose radius

RZ = Z-coordinate of the tip nose circle centre

RX = X-coordinate of the tip nose circle centre

WID = width of the parting-off tool

ZREF = Z-coordinate of the tool offset position

XREF = X-coordinate of the tool offset position

Table 8.2 SUBROUTINES OF THE CUTPLOT PROGRAM (PART 1)

SUBROUTINE	LEVEL	FUNCTION
CUTPLOT	1	Calls upon subordinate subroutines to decode geometric data from the Roller Profile Data File and to invite the user to supply machining data input, hence to obtain the details of the cutting tool geometric data from the tool library, and then calculates the compensated tool paths corresponding to the appropriate machining cycle chosen. The cutter location data will then be output to an intermediate file (CLDATA). At the same time, routines will be called up to draw a GINO plot of the tool path on the PERQ screen
GEO1	2	Decodes the geometric data for the roll by retrieving the Roller Profile Data, which had been generated by the Roll Editor Program
CLPLOT1	2	Calls the GINO commands to draw the axes and the title for the cutter path plot
TLIB	2	Stores the tool geometric data for all the cutting tools deployed in the turret of the Mori Seki CNC lathe
CPR	2	Invites the user to supply machining data input, then calls upon the corresponding machining cycle modules to calculate the cutter path location
RINIT	3	Initializes the roll profile listing
WRIT4	3	Outputs the coordinates of the starting point of a machining cycle to the cutter location data file
WRIT5	3	Outputs the coordinates of the ending point of a machining cycle to the cutter location data file
FINE	3	Generates the cutter paths for finishing
ROUGH	3	Generates the cutter paths for roughing out
POCKET	3	Generates the cutter paths for the pocketing cycle

Table 8.2 (PART 2)

SUBROUTINE	LEVEL	FUNCTION
GROOVE	3	Generates the cutter paths for the grooving cycle to rough out concave profiles
PLOTHP	3	Calls upon the HP7221 plotter driver facilities to plot the cutter path drawing by the HP7221c plotter
OFFSET	4	(see the OFFSET module)
CUTIN	4	Defines the cut-in line and the cutting start point for a finishing cycle or pocketing cycle
WRIT1	4	Outputs the cutter location data for a 3-stroke cutting cycle for roughing out
WRIT2	4	Outputs the cutter location data for a single stroke tool path for linear interpolation
WRIT3	4	Outputs the cutter location data for circular interpolation
SIGN	4	(see the OFFSET module)

Table 8.3 SUBROUTINES OF THE OFFSET MODULE

SUBROUTINE	LEVEL	FUNCTION
OFFSET	1	Determines the cutting direction, calls up subroutines to calculate the offset tool paths for a machining area, then sets up the offset machining area data structure
INIT2	2	Initialises the offset roll profile data
SIGN	2	Determines the identification sign for a circular arc
ANGLN	2	Determines the angle of a given line (w.r.t. the Z-axis)
ANGARC	2	Determines the angle of the line joining a point and the centre of an arc (w.r.t. the Z-axis)
ANGDIF	2	Determines the angle between two tool paths
LNLN	2	Calculates the coordinates of the change point between two offset linear tool paths
LNCIR	2	Calculates the coordinates of the change point between two offset tool paths, one linear and the other circular
CIRCIR	2	Calculates the coordinates of the change point between two offset circular tool paths
STL	3	Determines the offset of a point on a line
NML	3	Determines the offset of a point on a circular arc
QUAD	3	Cauculates the coordinates for points on a circular arc by solving the quadratic equation

Table 8.4 SUBROUTINES OF THE MFPOST PROGRAM

SUBROUTINE	LEVEL	FUNCTION
MFPOST	1	Calls up subordinate subroutines to decode the CLDATA file, generate the NC tape program data and displays the cutting path on the PERQ screen
STVPTF	2	Calls up the GRAFIKS routines to set up text frame and viewport on the PERQ screen
DETAPE	2	(see the DETAPE module)
INLIN	2	Outputs the block of instruction for linear interpolation
INLIN2	2	Outputs the block of instruction for linear interpolation (G code is the same as the preceding block of instruction)
INCIR	2	Outputs the block of instruction for circular interpolation
INCIR2	2	Outputs the block of instruction for circular interpolation (G code is the same as the preceding block of instruction)

Table 8.5 SUBROUTINES OF THE DETAPE MODULE

SUBROUTINE	LEVEL	FUNCTION
DETAPE	1	Decodes data from the NC tape file and plots the path on the PERQ screen
DECHAR	2	Decodes a block of NC tape data according to the corresponding address
CLIPCR	2	Plots a circular arc with start and end points and given circle centre points, at the given direction

Table 8.6 SUBROUTINES OF THE CHKTAP PROGRAM

SUBROUTINE	LEVEL	FUNCTION
CHKTAP	1	Calls up subordinate subroutines to input data from an NC Tape and plots the tool path on th PERQ screen for checking
TAPEIN	2	Decodes data from the NC Tape File and plots the tool path
DECHAR	3	Decodes a block of NC tape data according to the corresponding address
CLIPCR	3	Plots a circular arc with start and end points and given centre points, at the given direction

Chart 8.1 HIERARCHY OF THE CUTPLOT PROGRAM (PART 1)

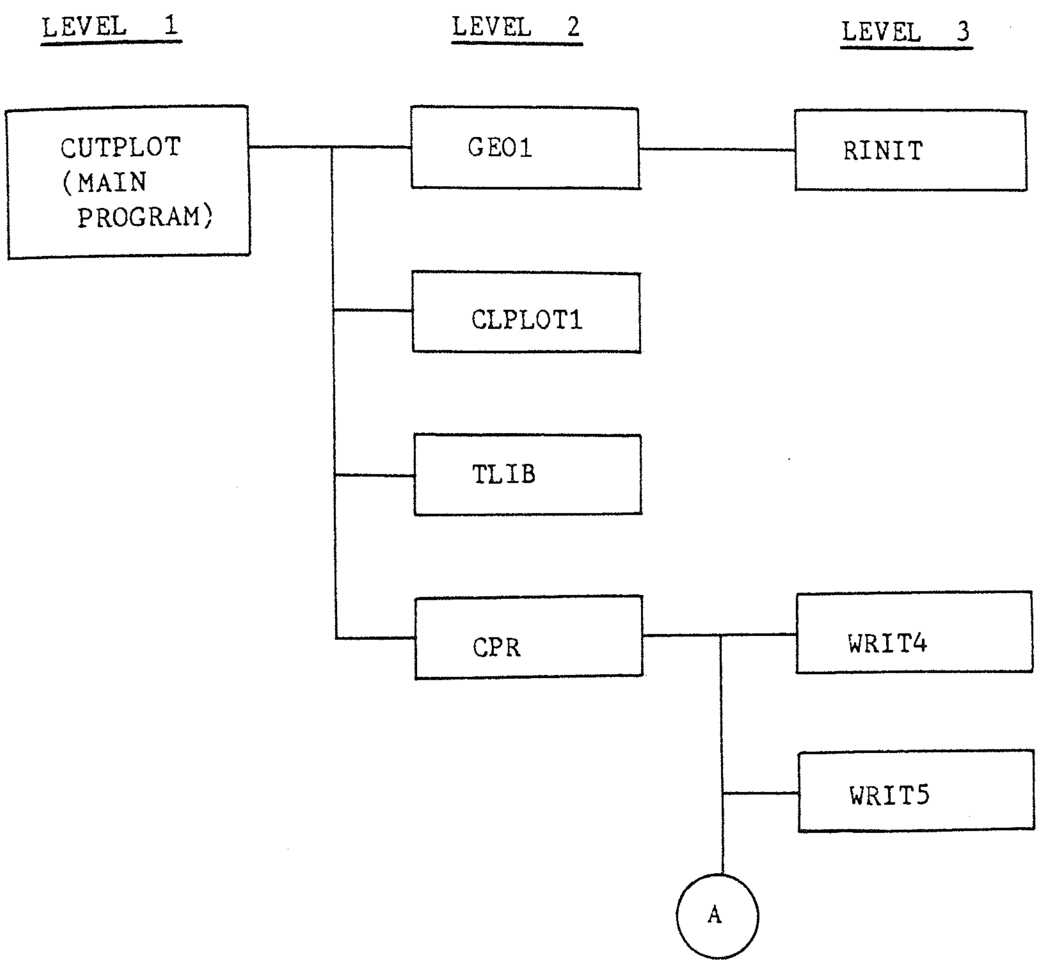


Chart 8.1 (PART 2)

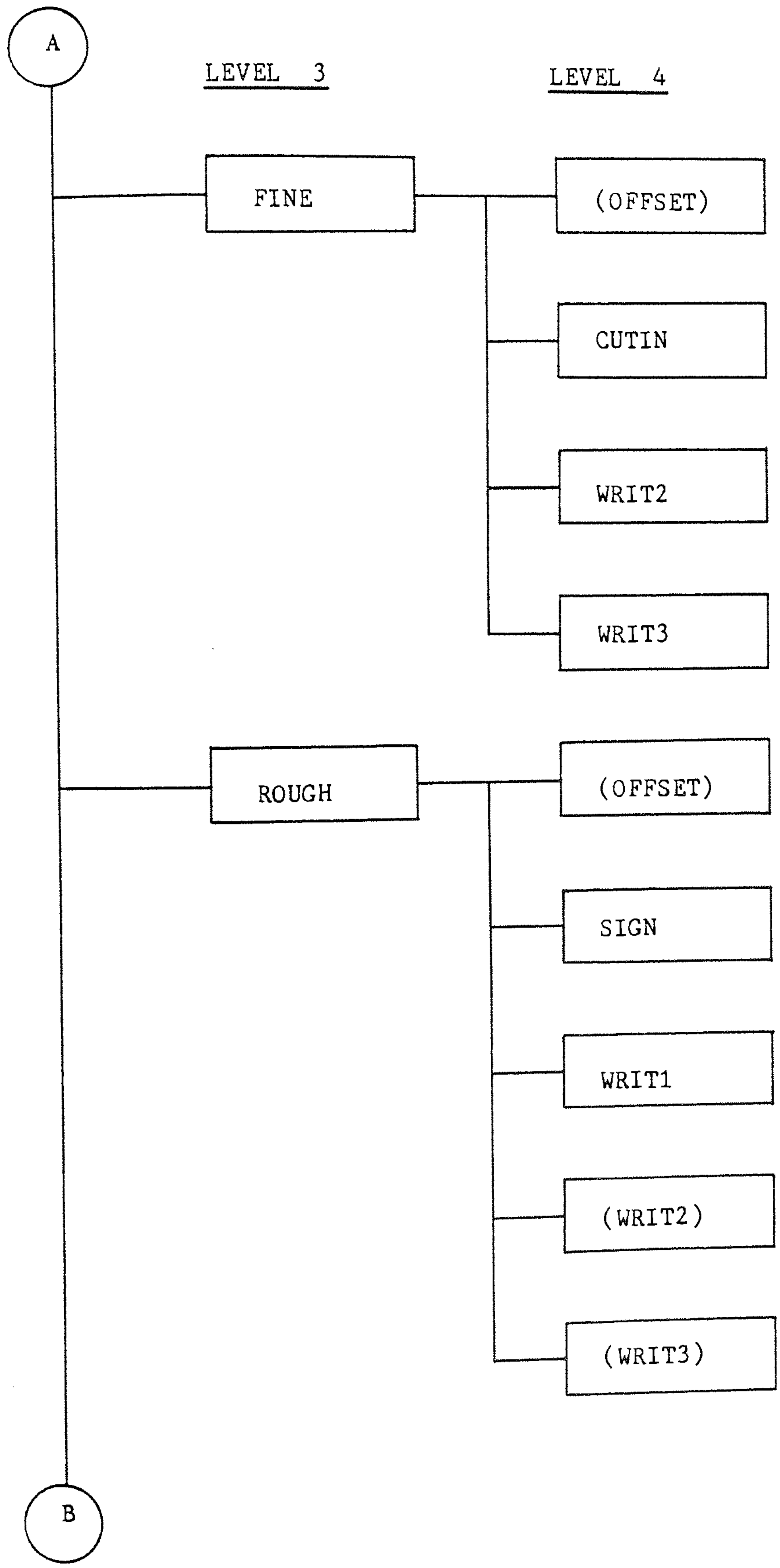


Chart 8.1 (PART 3)

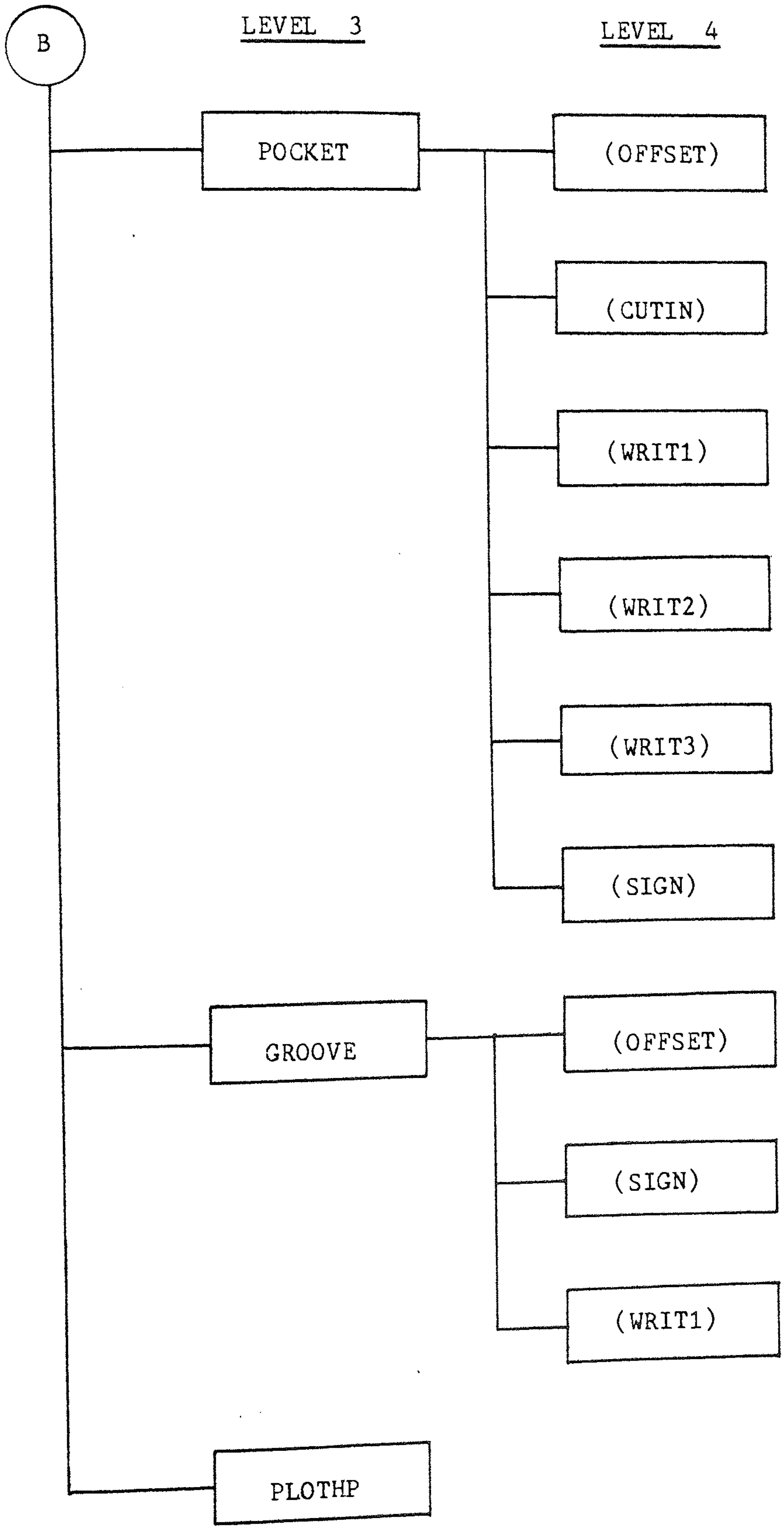


Chart 8.2 HIERARCHY OF THE OFFSET MODULE

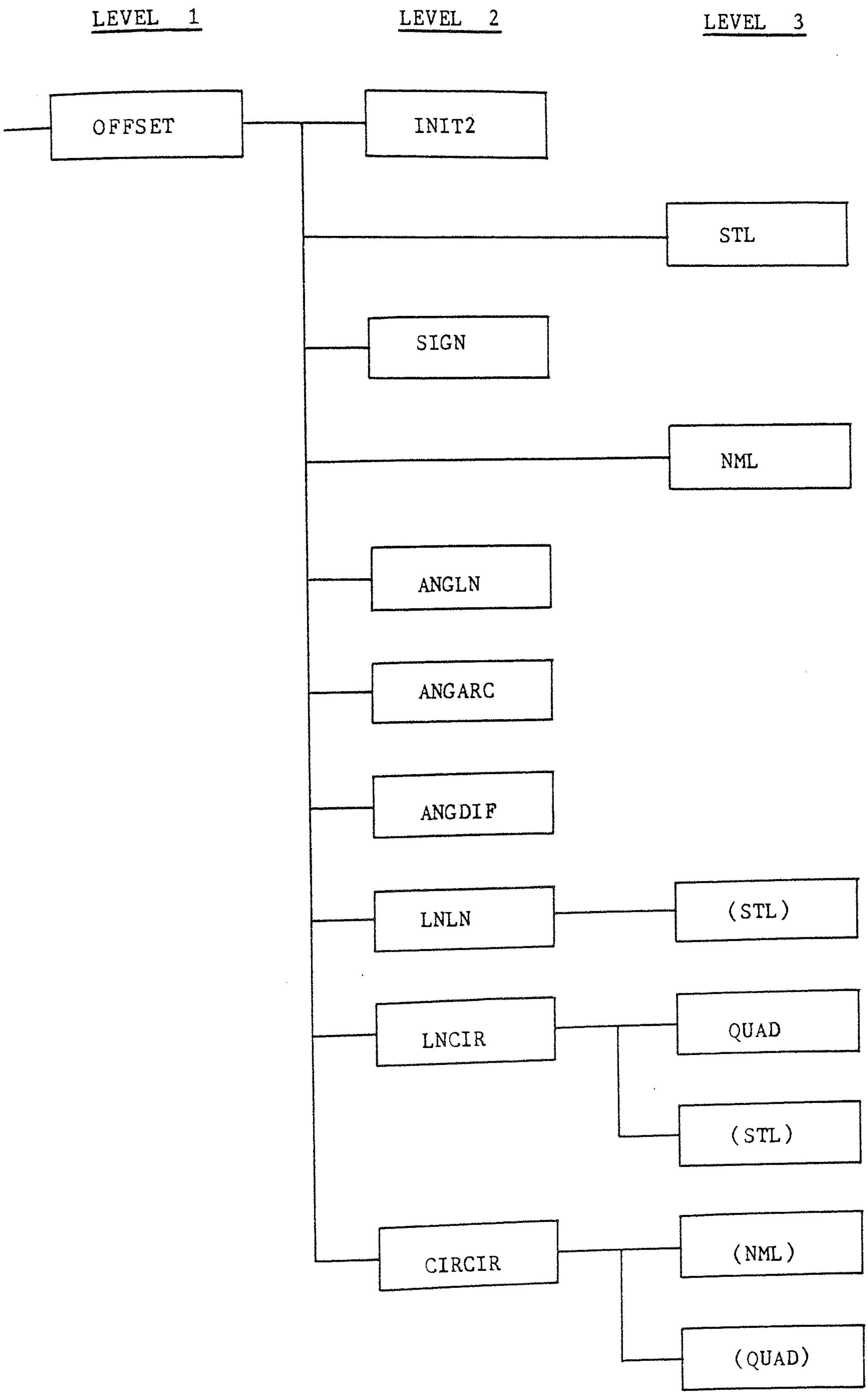


Chart 8.3 HIERARCHY OF THE MFPOST PROGRAM

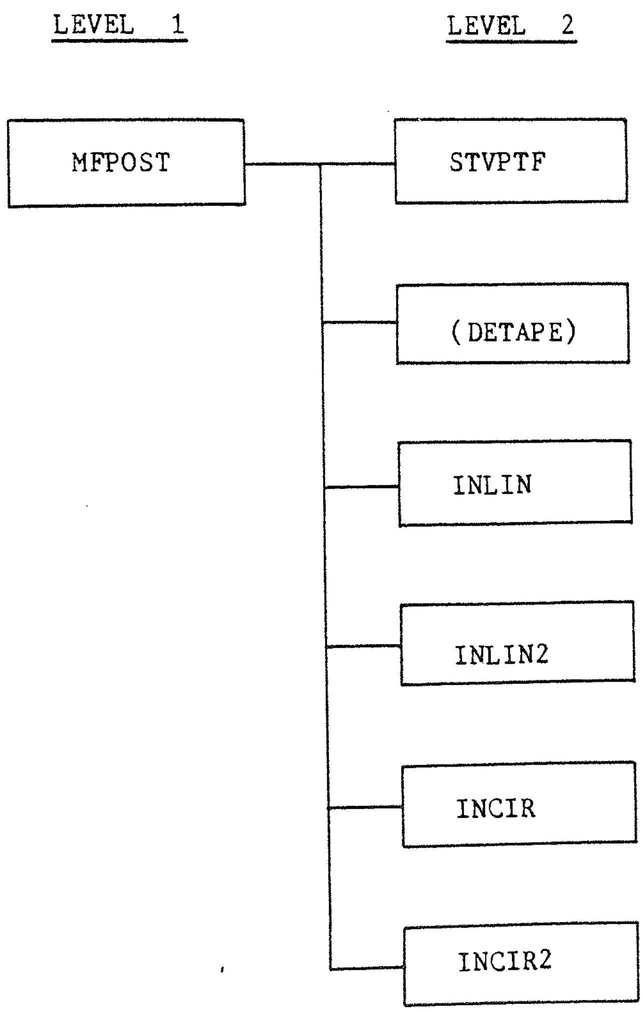


Chart 8.4 HIERARCHY OF THE DETAPE MODULE

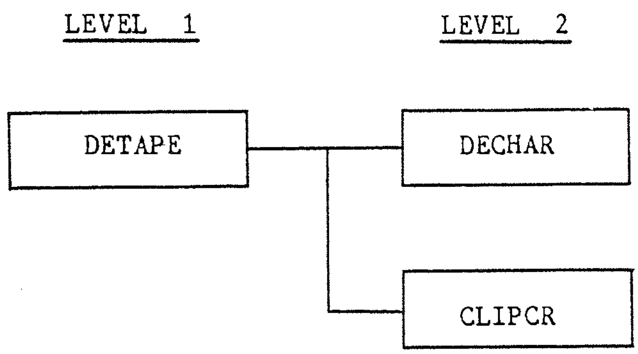
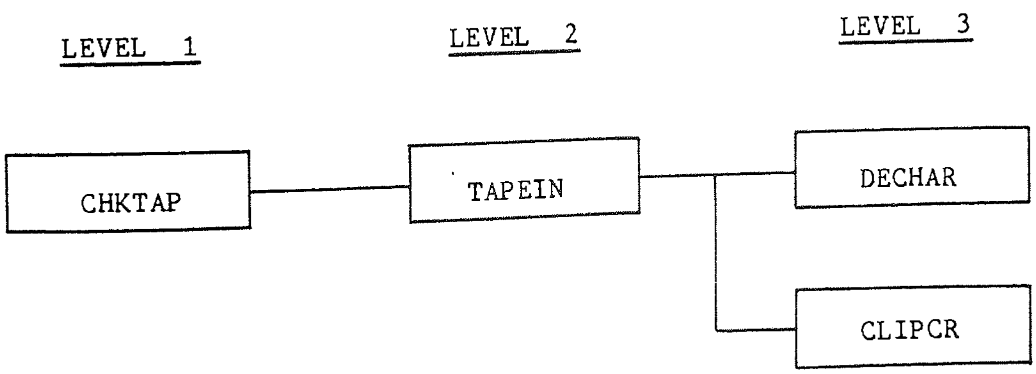


Chart 8.5 HIERARCHY OF THE CHKTAP PROGRAM



CHAPTER 9

CHAPTER 9

IMPLEMENTATION OF THE CAD/CAM SYSTEM

9.1 INTRODUCTION

In this research, all the developed software programs have been implemented successfully in the ICL PERQ computer system. As shown in Fig. 4.1, the programs have been integrated together to form a complete CAD/CAM system for the design and manufacture of form-rolls. The Finished Section Program, the Flower Pattern Program, and the Roll Design Program together form a complete roll design package which has the capability of generating complete and working roll designs. Facilities handling essential stages of roll designing function, starting from product or section definition right up to the production of roll designs are all available. Provisions have also been made for minor editing of roll profiles by the Roll Editor Program. The operation of the Roll Editor Program will generate the essential roll profile geometry data for NC tape-preparation process, as a matter of fact, this roll profile data file is the link for integrating the CAD and CAM functions. The Roll Machining Program, which is a special part-programming system, will decode the roll profile data for geometric data input and then produce the cutter location information. Finally, the Post-processor can then generate the actual NC tape program for the roll machining process.

As far as implementation is concerned, all the developed software programs have been stored in the PERQ fixed disc storage system, and, only simple commands are required for the execution of these programs.

9.2 THE PERQ FILE STORAGE SYSTEM

The 24 Mbyte fixed disc is the major memory storage device of the PERQ computer. This disc memory storage is divided into five partitions, and the names for these five partitions are BOOT, USER, PART 1, PART 2, and PART 3. All the files containing the PERQ operating system, the FORTRAN compiler, the GINO routines and the GRAFIKS routines are stored in the BOOT partition. While all the CAD/CAM application software developed in this research are stored in files in the USER partition. Therefore, the other three partitions PART 1, PART 2, and PART 3 are available to the user, during the execution of the CAD/CAM application software, for the storage of input and output data files.

9.3 INPUT AND OUTPUT DATA STORAGE

During the operation of the CAD/CAM software programs, the user will be led to supply input data to the system through the interactive man/machine dialogue. However, the input information would be dumped into designated data files, which could then be stored in the fixed disc storage as permanent records. Hence, if it is required to re-execute a program, it is then not necessary for the user to supply the input data through the man/machine dialogue again, but the system will retrieve the data from the designated files automatically.

The Finished Section Program is the first stage of the CAD/CAM software. The execution of this program requires a set of finished section design data through the interactive man/machine dialogue (Fig. 5.5). The input information will be stored in a section definition data file under a file name (D"SECNO"), where "SECNO" is the section number, which can contain up to 10 alphanumeric characters, and is the part number of the finished section. Meanwhile, an intermediate section data file (T-D"SECNO") will be generated and stored in the disc memory for later stage of processing.

During the operation of the Flower Pattern Program, the intermediate section data file (T-D"SECNO") will be retrieved for decoding the section definition data. In addition, the bending information, being supplied by the user through the man/machine dialogue, as shown in Fig. 5.6, will be dumped into a bending data file. The file name of this bending data file is (F"SECNO"). Similar to the Finished Section Program, an intermediate output file (T-F"SECNO") will be generated and kept in the disc memory storage for later stage of processing.

As it has been mentioned earlier, the Template program is not required for the integrated CAD/CAM software, but it has been implemented on the PERQ system as a supporting function. Nevertheless, the execution of the Template program does not require any special data input, instead, the program will capture information from the two intermediate output files (T-D"SECNO") and (T-F"SECNO"), which should have been generated by the Finished Section Program and the Flower Pattern Program respectively.

For the operation of the Roll Design Program, the section definition data and bending data will be retrieved directly from the files (T-D"SECNO") and (F"SECNO") respectively. However, a set of roll design data is required to be given by the user through the man/machine dialogue, as shown in Fig. 5.8. In this case, the roll design information will be kept in the disc memory in a file designated by the file name (R"SECNO"). Besides, the execution of the Roll Design Program will generate a roll profile data file, with a file name as (A-R"SECNO"), which contains the roll profile geometry data for all rolls in all rolling stages.

The execution of the Roll Editor Program, on the other hand, will pick up the appropriate roll geometry information from the file (A-R"SECNO"). The concerned roll profile data will then be modified interactively, again, through specially designed man/machine dialogue. Then, the updated roll profile data will be dumped into a file (E"SECNO"-"s""r"), where "s" is the stage number and "r" is the code which equals T for a top roll, B for a bottom roll, L or R for a left or right hand side roll respectively.

This roll profile data file (E"SECNO"-"s""r") will be retrieved by the Roll Machining Program for geometric data input, while the user has to provide the machining data input through the interactive man/machine dialogue. Simultaneously, the machining data input are recorded in a file with the file name (CU"SECNO"-"s""r"). With regard to the output, the cutter location data will be dumped into a file called CLDATA. This CLDATA file with then be interpreted by the Post-processor, and consequently, the NC tape program can be generated and stored in a file with a file name (NC"SECNO"-"s""r"). Thereafter, an NC control tape can then be obtained by copying this file to a tape punch.

If it is desired to check the syntax of the NC tape program, the Tape Checking Program can be used. However, this program serves only to decode but without manipulating the NC machining data from a (NC"SECNO"-"s""r") file. In fact, similar to the Template Program, this Tape Checking Program is only an optional function but not a necessary stage of the complete CAD/CAM system.

In essence, the above mentioned data files will be generated during the different stages of the execution of the CAD/CAM software system, as summarized in Table 9.1. These data files will be kept in a partition of the fixed disc memory storage. With the exception of the partitions BOOT and USER, which have been used for storing the system utility software and the CAD/CAM application software programs respectively, the other three partitions PART 1, PART 2, and PART 3 are available for the user to store these files.

9.4 THE PROCESSING COMMANDS

In order to facilitate the execution of the CAD/CAM software, the developed programs are deployed into different command files for different stages of execution of the complete CAD/CAM software.

With the existing processing arrangements on the PERQ computer system, the following processing commands are used:-

(1) In order to execute the Finished Section Program, whereas section definition data is required through the man/machine dialogue, the command is:
@SECTION

(2) On the other hand, if there exists a (D"SECNO") file, the command for re-executing the Finished Section Program, without supplying data through the terminal keyboard, is:
@SECTIONX

(3) The command for executing the Flower Pattern Program, whereas bending data is to be entered through the man/machine dialogue is:
@FLOWER

(4) On the other hand, with an existing (F"SECNO") file, the command for re-executing the Flower Pattern Program is:
@FLOWERX

(5) The command for processing the Template Program is:
@TEMPLATE

(6) The command for executing the Roll Design Program, whereas roll design data is to be entered through the man/machine

dialogue is:

@ROLL

(7) On the other hand, the command for re-executing the Roll Design Program, while there exists a (R"SECNO") file, is:

@ROLLX

(8) The command for executing the Roll Editor Program is:

@ROEDIT

(9) The command for executing the Roll Machining Program and the Post-processor consecutively is:

@TAPE

In this case, machining data for the Roll Machining Program has to be entered by the user through the man/machine dialogue.

(10) If the machining data input file exists, with the file name (CU"SECNO"-"s""r"), the command for re-executing the Roll Machining Program and then the Post-processor consecutively is:

@TAPEX

(11) Lastly, the command for executing the Tape Checking Program is:

@CHKTAPE

The contents of all the command files are given in Appendix 5, and Table 9.2 outlines the commands for execution at different stages of the CAD/CAM software.

Table 9.1 DATA FILES INVOLVED IN DIFFERENT STAGES IN THE CAD/CAM SOFTWARE SYSTEM

PROGRAM	INPUT DATA FILE	OUTPUT DATA FILE
FINISHED SECTION (R1PLOT)	<D"SECNO">	<T-D"SECNO">
FLOWER PATTERN (R2PLOT)	<T-D"SECNO"> <F"SECNO">	<T-F"SECNO">
TEMPLATE (R3PLOT)	<T-D"SECNO"> <T-F"SECNO">	
ROLL DESIGN (R4PLOT)	<T-D"SECNO"> <F"SECNO"> <R"SECNO">	<A-R"SECNO">
ROLL EDITOR (ROEDIT)	<A-R"SECNO">	<E"SECNO"-"s""r">
ROLL MACHINING (CUTPLOT)	<E"SECNO"-"s""r"> <CU"SECNO"-"s""r">	CLDATA
POST-PROCESSOR (MFPOST)	CLDATA	<NC"SECNO"-"s""r">
TAPE CHECKING (CHKTAP)	<NC"SECNO"-"s""r">	

NB :- "SECNO" : Section No. (of the Finished Section), can contain up to 10 alphanumeric characters

"s" : Stage No. (1 - 99)

"r" = T for Top roll
= B for Bottom roll
= L for Left side roll
= R for Right side roll

Table 9.2 THE PROCESSING COMMANDS

COMMAND	DESCRIPTION
@SECTION	Operating the Finished Section Program, while acquires data interactively through the keyboard
@SECTIONX	Re-executing the Finished Section Program
@FLOWER	Operating the Flower Pattern Program, while acquires data interactively through the keyboard
@FLOWERX	Re-executing the Flower Pattern Program
@TEMPLATE	Operating the Template Program
@ROLL	Operating the Roll Design Program, while acquires data interactively through the keyboard
@ROLLX	Re-executing the Roll Design Program
@ROEDIT	Operating the Roll Editor Program
@TAPE	Generating the NC Tape Program by executing the Roll Machining Program and the Post-processor consecutively, data input is acquired through the man/machine dialogue
@TAPEX	Re-executing the Roll Machining Program and the Post-processor consecutively for the generation of the NC Tape Program
@CHKTAPE	Operating the Tape Checking Program

CHAPTER 10

CHAPTER 10

CONCLUSIONS AND FUTURE WORK

10.1 CONCLUSIONS

The objectives of this research have been achieved, as in accordance with the development programme, computer software facilities have been developed to aid the form-roll design process, and to prepare the NC control tape program for the manufacture of form-rolls on a CNC lathe. All the computer programs have been successfully implemented, while being integrated together as a complete CAD/CAM software package, on the ICL PERQ computer system. It has been shown that the developed CAD/CAM software covers all aspects of work in form-roll design and manufacture. In summing-up, the following results and conclusions are obtained.

1. A stage by stage approach was chosen for computerising the activities involved in form-roll design and manufacture. The complete development programme had been executed in two distinct phases. The first phase, which had been completed by Ng [50] in his project, concerned with the establishment of CAD facilities for form-roll design. While on the other hand, this research has been carried out to comply with the second phase of the development programme, that is, to develop the CAM facilities for form-roll production, and hence set up an integrated CAD/CAM system. Computer programs had then been designed and

developed in both phases, through the stage-by-stage approach, to enhance the form-roll design and manufacture process. With such an approach, disruption to the existing design and manufacturing activity in the company has been reduced to a minimum, resulting in a smooth change over from conventional methods to the use of the new technology. Besides, this approach permits the development of a dedicated CAD/CAM system, which can be tailored to suit the company's special interests and requirements.

2. The Finished Section Program; the Flower Pattern Program; the Template Program and the Roll Design Program, which were developed in Phase 1 of the development programme, were originally written in FORTRAN IV, utilising the GINO-F library routines for processing graphics, and had been implemented as a batch processing system accessible via a remote terminal linked to the University's ICL 1904s computer. This was, however, only a short-term arrangement. It has been the long term objective of this research, referring to Phase 2 of the development programme, to establish a stand-alone CAD/CAM system in the company. Accordingly, an ICL PERQ single-user computer system has been purchased. The four above mentioned CAD software programs had then been modified and implemented on the PERQ computer. In the first place, all these four program units had been converted to adapt to the PERQ's FORTRAN 77 compiler, and secondly, making use of the interactive and instant computing characteristic of the PERQ, question/answer type man-machine dialogues had been introduced into these program

units for data acquisition. Similarly, the Roll Editor Program; the Roll Machining Program; the CNC Lathe Post-processor and the Tape Checking Program, which were designed and developed in this research, were all written in FORTRAN 77 and utilising interactive question/answer man-machine dialogues for data acquisition. With this approach, it has been found out that the CAD/CAM software system on the PERQ has provided a high degree of user-friendliness which could not be achieved by the batch processing system. Furthermore, running time and lead time have been found to be reduced in executing the software programs on the PERQ computer, as the user can obtain the results instantaneously, while modifications or corrections for the design input can be made immediately and the programs can be re-execute again.

3. The Roll Editor Program has been developed to provide programming facilities for executing minor design changes and final refinements. A roll profile comprises of linear and circular elements. While designing this program, algorithms were constructed to handle the analytic geometry of lines and circles in a 2-d co-ordinate system. Independant program modules were then developed to incorporate the different algorithms. The Roll Editor Program has been equipped with four editing functions, namely, inserting a new element; replacing an old element by a new one; modifying a sharp corner by a chamfer or blending arc; and deleting an existing element. It has been found that these four editing functions are, in general terms, sufficient enough for the user to carry out minor roll design modifications.

Besides, GRAFIKS routines were incorporated in the Roll Editor Program for the generation of interactive graphics on the PERQ terminal screen. Hence, the operation of the Roll Editor Program can dynamically change the displayed roll profile drawing. Also, it has been found possible to exploit the cursor control facilities provided by GRAFIKS, to establish user-friendly commands for the user to enlarge a particular part of the roll profile, so as to obtain a larger and clearer view of the details. In all cases, the Roll Editor Program has been found to be a versatile tool for the user, to perform minor roll design modifications, for instance to modify a sharp corner by a blending arc or a chamfer.

4. Using a CNC lathe can actually enhance the form-roll machining process. In this research, standard tooling and workpiece set up procedures have been specified. In contrast to the conventional manual roll turning process, NC turning does not require, in the first place, the full size wire template as a production aid, and also on the other hand, NC turning does not rely upon the skill of the operator. However, it was found that the efficiency of NC roll turning depends mainly on the part-programming method. Manual part-programming was found to be inappropriate because it is very time-consuming and prone to errors due to the complex roll profile geometry. The use of FAPT, which is an APT like computer-aided part-programming system, was found to be helpful in simplifying the manual part-programming process. Nevertheless, this approach was found to

be lacking of a direct link between NC part-programming and the CAD software for roll design, and hence, repetitive effort was required to define the roll profile geometry. Thus, the FAPT system was used only in the interim, while the special-purpose roll machining program was being developed.

5. The development of the special-purpose roll machining part-programming system on the PERQ computer has established the link between the CAD and CAM functions. This program has been found to be easy to execute and hence can minimise the user's burden in preparing the NC control tape programs for machining form-rolls. In the first place, the geometric data of a roll will be decoded directly by the program, by retrieving the corresponding Roll Contour Data File from the PERQ's fixed disc memory storage. Thus, the user is no more required to supply the geometric data again, and hence there is no possibility in defining an incorrect roll profile geometry. On the other hand, the user needs only to supply, through an interactive question/answer dialogue via the PERQ screen and the keyboard, some simple machining data input. The input format was designed in such a way that only numeric codes are required, therefore, those cumbersome alphanumeric syntax and vocabulary as used in most of the APT like part-programming systems are not needed. This approach has helped to eliminate the requirement of the user's special knowledge in NC part-programming. Besides, the user can observe on the screen, the tool location paths generated by the program. This provides a spontaneous check for the user to see if the

correct shape of roll is being machined in the correct manner.

While designing this Roll Machining Program, it was found that according to the geometry of the roll profile, which can either be concave or convex, four different types of machining cycles are sufficient enough to deal with the roughing out and finishing cut of all roll profiles. In essence, the roughing cycle was designed for the area clearance of convex profiles; while the grooving and pocketing cycles were designed for the area clearance of concave profiles; and the finishing cycle was designed for finishing profiling.

6. The Roll Machining Program was so designed that the tool placements generated for the cutting process are dumped into the cutter location data file CLDATA. This CLDATA file is independant of the tape format for any NC turning machine. In this research, a post-processor program has been designed and developed to convert the CLDATA file into the NC tape program for using on the Mori-Fanuc CNC lathe system. If in case it is required in the future to employ another NC lathe system for the roll machining process, the post-processor can be modified readily to adapt to the different program formats and codes.

Again, the interactive graphics facilities has been incorporated into the post-processor program, thereby, the tool path plot based on the NC tape program will be displayed on the PERQ screen while the post-processor program is being executed.

This approach, therefore, provides a check on the syntax of the actual NC tape program.

7. A Tape Checking program has also been developed, to provide the user a facility for re-checking the information contained in the NC control tape program. Similar to the Post-processor Program, the execution of the Tape Checking Program will produce a plot of the tool path on the PERQ screen, therefore acts as a means for visually inspecting the NC tape program before loading the tape in the CNC lathe for actual machining.

8. All the developed software programs, for both CAD and CAM, have been successfully implemented on the PERQ computer. These CAD and CAM programs are deployed into different command files in the PERQ's fixed disc memory storage. It is possible, therefore, to call up the simple commands for executing the different stages of the complete CAD/CAM software. While operating the software, the user will be led to supply data input through the question/answer type man-machine dialogues, training requirement for learning to use the system has therefore found to be trivial. All input and output data will be dumped automatically into data files accordingly. The arrangement of these data files, in effect, gives a coherent link for the CAD and CAM software programs.

9. All the application software programs were written in the commonly used FORTRAN 77, and were developed into program modules. This approach allows the application programs to be

maintained easily. For any further developments of the CAD/CAM system, the modules' architecture is such that modification and interfacing with other modules is a relatively straightforward task.

10. It has been shown that smaller and medium size manufacturing companies can gain benefits from the CAD/CAM technology by investing in an inexpensive, single-user stand-alone computer system. In this research, the development of the interactive minicomputer based CAD/CAM system has been found beneficial in all aspects of form-roll design and manufacture, including the design and draughting of the finished section; the flower pattern and the roll design; roll editing and also the generation of NC part programs for machining the rolls on a CNC lathe.

10.2 FUTURE WORK

In this research, the complete form-roll design and manufacture process has been computerised. It is, nevertheless, possible and worthwhile to enhance and broaden the scope of the present CAD/CAM package by carrying out researches into several related areas.

The present computerised roll design software is based on the classical method of designing rolls, starting with the flower diagram. However, the roll designer is required to have the knowledge and expertise to determine, mainly through trial and error, the number of forming stages required. Future work can be undertaken to establish a precise method for determining the forming sequence. This has to be done by studying the mechanism of the metal-forming process in cold roll forming. To begin with, research can be done by carrying out analytical and experimental work on the stress/strain distribution on the metal strip during the cold roll forming process. More advanced work can be undertaken to develop a finite element stress/strain analyses algorithm, and hence the existing CAD/CAM software can be enhanced by the addition of the computer-aided finite element stress/strain analyses module. Furthermore, it is also of practical importance to perform a study to establish methods for predicting springback allowance in the roll forming process.

Concerning the foll forming process itself, the finished stock, at the end of a roll forming line, enters a mating die block aperature on a cut-off die, which cuts the stock to appropriate lengths. As the cross-section of the die-block closely resembles the finished section, it will then be possible to create the particular geometric statements required by an NC EDM machine, which is used for manufacturing die-blocks. Thus, one possible area for further development is the CAD/CAM of die blocks.

On the other hand, the present CAD/CAM package can be further extended by adding in facilities to obtain the cost estimate of a form-roll. This can be done by calculating the material cost according to the material chosen and the size of the roll blank, while the machining cost is a function of the cutting speed and feedrate used in the programmed tool paths. Furthermore, a technological data base can be established, so that the cutting speed and feedrate used in the NC program can be determined automatically according to the roll material and cutting tool chosen.

Finally, the special-purpose roll turning processor can be elaborated into a general purpose part-programming system for NC turning. This can be done by firstly setting up a geometric description model for geometric data input of any lathe jobs. Secondly, more machining cycles, for instance threading cycles and boring cycles, can be included into the program to cover all the jobs done on a turning machine.

APPENDICES

APPENDIX 1

PROGRAM LISTING OF THE HP7221C PLOTTER DRIVER - THE 'HP7221' MODULE

```
      SUBROUTINE OPENHP(NCHANL,HPFILE)
C
C  THIS SUBROUTINE IS TO INITIALISE THE HP7221C PLOTTER
C  PLOTTING DATA WILL BE OUTPUT TO A FILE CALLED HPFILE
C
      CHARACTER HPFILE*(*),STRING*5
C
      COMMON /BHP1/N
      COMMON /BHP2/XSHIFT,YSHIFT,XORIG,YORIG
      COMMON /BHP3/DWGSC
C
      N=NCHANL
      OPEN(NCHANL,FILE=HPFILE)
C
C  THE PLOTTER WILL RESET ALL GRAPHICS VARIABLES TO THEIR DEFAULT
C   VALUES. HOWEVER, THE GRID SYSTEM WILL BE RESET HERE TO USE
C   1 PLOTTER UNIT TO REPRESENT 1 MACHINE UNIT, THEREFORE,
C   1 PLOTTER UNIT = 0.025MM
C
C  INITIALISE THE ORIGIN
      XORIG=0.0
      YORIG=0.0
      XSHIFT=0.0
      YSHIFT=0.0
C
C  INITIALISE THE SCALE OF THE DWG
      DWGSC=1.0
C  TURN ON THE PLOTTER
      CALL PLOTON
C
C  INITIALISATION
      WRITE(NCHANL,1)
1     FORMAT(' ~_')
C  RESET GRID SYSTEM
      CALL MBP(15200,10000,I,STRING)
      WRITE(NCHANL,2) STRING(:I)
2     FORMAT(' ~S',A,'}')
      RETURN
      END

      SUBROUTINE PLOTON
C  THIS SUBROUTINE IS TO TURN ON THE PLOTTER
      CHARACTER CA*1
      COMMON /BHP1/NCHANL
C
      CA=CHAR(27)
      WRITE(NCHANL,1) CA
1     FORMAT(' ',A,'.(')
      RETURN
      END

      SUBROUTINE CLOSHP
C  THIS SUBROUTINE IS TO TURN OFF THE PLOTTER
C   ANY EXISTING PEN WILL BE PUT BACK FIRST,
C   THE PEN HOLDER WILL BE BACK TO THE UPPER RIGHT POSITION
C
```

```
      CHARACTER STRING*5,CA*1
      COMMON /BHP1/NCHANL
C
      CALL PEN(0)
C  GO BACK TO THE TOP RIGHT HAND CORNER
      CALL HPUNIT(390.0,IX)
      CALL HPUNIT(260.0,IY)
      CALL MBP(IX,IY,I,STRING)
      WRITE(NCHANL,1) STRING(:I)
1     FORMAT(' p',A,'}')
C  TURN OFF THE PLOTER
      CA=CHAR(27)
      WRITE(NCHANL,2) CA
2     FORMAT(' ',A,'.)')
      RETURN
      END

      SUBROUTINE PEN(K)
C  THIS IS TO SELECT A PEN WITH NUMBER EQUALS TO K
      CHARACTER PK*1
      COMMON /BHP1/NCHANL
C
      IF(K.LT.0.OR.K.GT.8) THEN
      PRINT*,' *** ERROR *** PEN FOR PLOTTER SHLD BE 1-8'
      K=1
      END IF
C
      PK=CHAR(K+64)
      WRITE(NCHANL,1) PK
1     FORMAT(' v',A,'}')
      RETURN
      END

      SUBROUTINE HPSCAL(NSCALE)
C  THIS IS TO SPECIFY THE SCALE OF THE DWG
C
      REAL NSCALE
      COMMON /BHP3/DWGSC
C
      DWGSC=NSCALE
      RETURN
      END

      SUBROUTINE SHIFTA(XNORIG,YNORIG)
C
C   THIS IS TO SHIFT THE ORIGIN TO THE DESIRED POINT
C   THE NEW ORIGIN IS (XNORIG,YNORIG) IN THE ABS. COOR. SYSTEM
C
      COMMON /BHP2/XSHIFT,YSHIFT,XORIG,YORIG
      XSHIFT=XNORIG-XORIG
      YSHIFT=YNORIG-YORIG
      XORIG=XORIG+XSHIFT
      YORIG=YORIG+YSHIFT
      RETURN
      END
```

```
      SUBROUTINE SHIFTI(XSHIFT,YSHIFT)
C
C   THIS IS TO SHIFT THE ORIGIN BY A DISTANCE (XSHIFT,YSHIFT)
C
      COMMON /BHP2/XSHT,YSHT,XORIG,YORIG
      XSHT=XSHIFT+XSHT
      YSHT=YSHIFT+YSHT
      XORIG=XORIG+XSHIFT
      YORIG=YORIG+YSHIFT
      RETURN
      END

      SUBROUTINE DRAWA(X,Y)
C  THIS IS TO DRAW A LINE TO A DEFINED POSITION (X,Y) ACCORDING
C   TO THE ABSOLUTE COORDINATE SYSTEM (X,Y IN MM)
      CHARACTER STRING*5
      COMMON /BHP1/NCHANL
      COMMON /BHP2/XSHIFT,YSHIFT,XORIG,YORIG
      COMMON /BHP3/DWGSC
C
      XP=(X+XSHIFT)*DWGSC
      YP=(Y+YSHIFT)*DWGSC
C
      CALL HPUNIT(XP,IX)
      CALL HPUNIT(YP,IY)
      CALL MBP(IX,IY,I,STRING)
      WRITE(NCHANL,1) STRING(:I)
 1    FORMAT(' q',A,'}')
      RETURN
      END

      SUBROUTINE MOVEA(X,Y)
C  THIS IS TO MOVE THE PEN TO A DEFINED POSITION (X,Y) ACCORDING
C   TO THE ABSOLUTE COORDINATE SYSTEM (X,Y IN MM)
      CHARACTER STRING*5
      COMMON /BHP1/NCHANL
      COMMON /BHP2/XSHIFT,YSHIFT,XORIG,YORIG
      COMMON /BHP3/DWGSC
C
      XP=(X+XSHIFT)*DWGSC
      YP=(Y+YSHIFT)*DWGSC
C
      CALL HPUNIT(XP,IX)
      CALL HPUNIT(YP,IY)
      CALL MBP(IX,IY,I,STRING)
      WRITE(NCHANL,1) STRING(:I)
 1    FORMAT(' p',A,'}')
      RETURN
      END

      SUBROUTINE DRAWI(X,Y)
C  THIS IS TO DRAW A LINE TO A DEFINED POSITION (X,Y) INCREMENTALLY
C  (X,Y IN MM)
      CHARACTER STRING*6
      COMMON /BHP1/NCHANL
      COMMON /BHP3/DWGSC
```

```
C
      XP=X*DWGSC
      YP=Y*DWGSC
      CALL HPUNIT(XP,IX)
      CALL HPUNIT(YP,IY)
      CALL PMB(IX,IY,I,STRING)
      WRITE(NCHANL,1) STRING(:I)
1     FORMAT(' s',A,'}')
      RETURN
      END

      SUBROUTINE MOVEI(X,Y)
C  THIS IS TO MOVE THE PEN TO A DEFINED POSITION (X,Y) INCREMENTALLY
C  (X,Y IN MM)
      CHARACTER STRING*6
      COMMON /BHP1/NCHANL
      COMMON /BHP3/DWGSC
C
      XP=X*DWGSC
      YP=Y*DWGSC
      CALL HPUNIT(XP,IX)
      CALL HPUNIT(YP,IY)
      CALL PMB(IX,IY,I,STRING)
      WRITE(NCHANL,1) STRING(:I)
1     FORMAT(' r',A,'}')
      RETURN
      END

      SUBROUTINE ARC(XS,YS,XE,YE,XC,YC,ISENSE)
C  THIS SUBROUTINE IS TO PLOT AN ARC WITH GIVEN STARTING AND ENDING
C   POINTS, WITH KNOWN CENTRE AND THE DIRECTION OF THE ARC
C    ISENSE = 1    FOR CCW ARC
C    ISENSE =-1    FOR CW ARC
      CHARACTER ASCODE*3,ESCODE*3,SENSE,RCODE*3
      INTEGER QUAD(2)
C
C  OBTAIN THE ANGLE FOR STARTING AND ENDING
      COMMON /BHP1/NCHANL
      COMMON /BHP2/XSHIFT,YSHIFT,XORIG,YORIG
      COMMON /BHP3/DWGSC
C
      XSP=(XS+XSHIFT)*DWGSC
      YSP=(YS+YSHIFT)*DWGSC
      XEP=(XE+XSHIFT)*DWGSC
      YEP=(YE+YSHIFT)*DWGSC
      XCP=(XC+XSHIFT)*DWGSC
      YCP=(YC+YSHIFT)*DWGSC
C
      IF(ABS(XSP-XCP).LE.0.05) XCP=XSP
      IF(ABS(YSP-XCP).LE.0.05) YCP=YSP
      IF(ABS(XEP-XCP).LE.0.05) XEP=XCP
      IF(ABS(YEP-YCP).LE.0.05) YEP=YCP
C
      R=SQRT((YSP-YCP)**2+(XSP-XCP)**2)
      DO 100 I=1,2
      IF(I.EQ.1) THEN
```

```
      XP=XSP
      YP=YSP
      ELSE
      XP=XEP
      YP=YEP
      END IF
C
      IF(ABS(XP-XCP).GT.R) THEN
      RADIAN=1.0
      ELSE
      RADIAN=ABS(XP-XCP)/R
      END IF
C
      THETA=ACOS(RADIAN)*180.0/3.14159
      IF(XP.GE.XCP.AND.YP.GE.YCP) QUAD(I)=1
      IF(XP.LT.XCP.AND.YP.GE.YCP) QUAD(I)=2
      IF(XP.LE.XCP.AND.YP.LT.YCP) QUAD(I)=3
      IF(XP.GT.XCP.AND.YP.LT.YCP) QUAD(I)=4
      IF(QUAD(I).EQ.2) THETA=180.0-THETA
      IF(QUAD(I).EQ.3) THETA=180.0+THETA
      IF(QUAD(I).EQ.4) THETA=360.0-THETA
C
      IF(I.EQ.1) THETAS=THETA
      IF(I.EQ.2) THETAE=THETA
100   CONTINUE
      IF(ISENSE.EQ.1) SENSE='u'
      IF(ISENSE.EQ.-1) SENSE='t'
      CALL HPUNIT(R,IR)
      CALL MBN(IR,I1,RCODE)
      CALL MBA(THETAS,I2,ASCODE)
      CALL MBA(THETAE,I3,ESCODE)
      WRITE(NCHANL,1) SENSE,RCODE(:I1),ASCODE(:I2),ESCODE(:I3)
1     FORMAT(' ',A1,A,A,A,'}')
      RETURN
      END

      SUBROUTINE DASFIX(DASH)
C  THIS IS TO USE FIXED DASH LINE FONT - MAKE UP OF
C   A 1-1 PATTERN, IN A 'TOTAL' MM INTERVAL
      CHARACTER CO*3
      COMMON /BHP1/NCHANL
C
      IF(DASH.EQ.0.0) THEN
C  THIS IS TO TERMINATE THE DASH LINE MODE
      WRITE(NCHANL,9)
9     FORMAT(' ~Q')
      RETURN
      END IF
      TOTAL=DASH*2.0
      CALL HPUNIT(TOTAL,IT)
      CALL MBN(IT,I,CO)
      WRITE(NCHANL,1) CO(:I)
1     FORMAT(' ~Q!A',A)
      RETURN
      END
```

```
      SUBROUTINE DASVAR(TOTAL)
C  THIS IS TO USE VARIABLE DASH LINE FONT - MAKE UP OF
C   A FIXED PATTERN, BEING 4-2-2-4
      CHARACTER CO*3
      COMMON /BHP1/NCHANL
C
      CALL HPUNIT(TOTAL,IT)
      CALL MBN(IT,I,CO)
      WRITE(NCHANL,1) CO(:I)
1     FORMAT(' ~R$B"B$',A)
      RETURN
      END

      SUBROUTINE ROTATE(ANGLE)
C  THIS SUBROUTINE EVOKES THE ROTATION OPTION, ROTATION WILL
C   BE DONE IN A CCW DIRECTION, AT AN ANGLE CUMULATED BY 'ANGLE'
      CHARACTER TEXT*3
      COMMON /BHP1/NCHANL
C
      IF(ANGLE.EQ.0.0) THEN
C  THIS IS TO PUT THE ROTATION BACK TO THE DEFAULT VALUE, IE 0 DEG
      WRITE(NCHANL,9)
9     FORMAT(' w')
      RETURN
      END IF
C
      CALL MBA(ANGLE,I,TEXT)
      WRITE(NCHANL,1) TEXT(:I)
1     FORMAT(' w',A)
      RETURN
      END


      SUBROUTINE ROTATX
C  THIS SUBROUTINE EVOKES THE ROTATION OPTION, ROTATION WILL
C   BE DONE IN A CCW DIRECTION, AT AN ANGLE EQUALS TO THE LAST
C   ANGLE PLOTTED
      COMMON /BHP1/NCHANL
C
      WRITE(NCHANL,1)
1     FORMAT(' x')
      RETURN
      END

      SUBROUTINE CHRSIZ(WID,HT)
C  THIS IS TO SET THE SIZE OF A CHARACTER
      REAL LINE
      CHARACTER TEXT*5
      COMMON /BHP1/NCHANL
C
      SPACE=WID*3/2
      LINE=HT*2
      CALL HPUNIT(SPACE,IS)
      CALL HPUNIT(LINE,IL)
      CALL MBP(IS,IL,I,TEXT)
      WRITE(NCHANL,1) TEXT(:I)
```

```
1     FORMAT(' ~%',A)
      RETURN
      END

      SUBROUTINE CHRSLT(ANGLE)
C  THIS IS TO SPECIFY THE SLANT ANGLE FOR LABEL SLANT MODE
      CHARACTER TEXT*3
      COMMON /BHP1/NCHANL
C
      CALL MBA(ANGLE,I,TEXT)
      WRITE(NCHANL,1) TEXT(:I)
1     FORMAT(' ~/',A)
      RETURN
      END

      SUBROUTINE CHRSTR(STRING)
C  THIS IS TO PLOT A SPECIFIED LABEL STRING
      CHARACTER CA,CB,STRING*(*)
      COMMON /BHP1/NCHANL
C
      CA=CHAR(39)
      CB=CHAR(3)
      WRITE(NCHANL,1) CA,STRING,CB
1     FORMAT(' ~',A,A,A)
      RETURN
      END

      SUBROUTINE CHRINT(INT,N)
C  THIS IS TO PLOT THE INTEGER INT AS A LABEL STRING
C   N IS THE WIDTH OF THE INGERGER FIELD
      CHARACTER CA,CB,STRING*10
      COMMON /BHP1/NCHANL
C
      CA=CHAR(39)
      CB=CHAR(3)
      WRITE(STRING,9) INT
9     FORMAT(I10)
      I=10-N+1
      WRITE(NCHANL,1) CA,STRING(I:),CB
1     FORMAT(' ~',A,A,A)
      RETURN
      END

      SUBROUTINE CHRFON(NA,NB)
C  THIS IS TO SELECT THE STANDARD AND ALTERNATE FONTS
      CHARACTER TEXT*5
      COMMON /BHP1/NCHANL
C
      IF(NA.LT.0.OR.NA.GT.5) THEN
      NA=0
      PRINT*,' *** ERROR *** NO. OF FONT IS 0 - 5'
      END IF
      IF(NB.LT.0.OR.NB.GT.5) THEN
      NB=0
      PRINT*,' *** ERROR *** NO. OF FONT IS 0 - 5'
      END IF
```

```
      CALL MBP(NA,NB,I,TEXT)
      WRITE(NCHANL,1) TEXT(:I)
1     FORMAT(' ~P',A)
      RETURN
      END

      SUBROUTINE FONTSD
C  THIS IS TO SET THE STD FONT AS THE CURRENT ONE
      CHARACTER CODE,CA,CB
      COMMON /BHP1/NCHANL
C
      CA=CHAR(39)
      CB=CHAR(3)
      CODE=CHAR(15)
      WRITE(NCHANL,1) CA,CODE,CB
1     FORMAT(' ~',A,A,A)
      RETURN
      END

      SUBROUTINE FONTAL
C  THIS IS TO SET THE ALTERNATIVE FONT AS THE CURRENT ONE
      CHARACTER CODE,CA,CB
      COMMON /BHP1/NCHANL
C
      CA=CHAR(39)
      CB=CHAR(3)
      CODE=CHAR(14)
      WRITE(NCHANL,1) CA,CODE,CB
1     FORMAT(' ~',A,A,A)
      RETURN
      END

      SUBROUTINE HPUNIT(X,N)
C  THIS SUBROUTINE SERVES TO CHANGE THE INPUT DIMENSIONS IN MM TO
C  MACHINE UNITS FOR THE HP7221C PLOTTER
C     1 MACHINE UNIT= 0.025MM
C
      XN=X/0.025
      N=INT(XN)
      RETURN
      END

      SUBROUTINE SBN(N,I,ACODE)
C  THIS IS TO GENERATE THE ASCII CHARACTER FOR SINGLE BYTE NUMBER
C
      CHARACTER ACODE*1
      IF(N.LT.0.OR.N.GT.63) THEN
      PRINT*,' *** ERROR *** SBN OUT OF RANGE (0 - 63)'
      RETURN
      END IF
C
      IF(N.LT.32) N=N+64
      I=1
      ACODE=CHAR(N)
      RETURN
      END
```

```
      SUBROUTINE MBN(N,I,TEXT)
C  THIS IS TO GENERATE THE ASCII CHARACTER FOR MULTIPLE BYTE NUMBER
C
      CHARACTER ACODE*1,AC1*1,AC2*2,AC3*3,TEXT*3
C
      IF(N.LT.0.OR.N.GT.32768) THEN
      PRINT*,' *** ERROR *** MBN OUT OF RANGE (0 - 32767)'
      RETURN
      END IF
C
      IF(N.LE.15) THEN
      AC1=CHAR(N+96)
      I=1
      TEXT=AC1
C
      ELSE IF(N.GT.15.AND.N.LE.1023) THEN
      N1=N/64
      N2=N-64*N1
      CALL SBN(N2,IBYTE,ACODE)
      AC1=CHAR(N1+96)
      AC2=AC1//ACODE
      I=2
      TEXT=AC2

C
      ELSE
      N1=N/4096
      NR=N-4096*N1
      N2=NR/64
      N3=NR-64*N2
      AC1=CHAR(N1+96)
      CALL SBN(N2,IBYTE,ACODE)
      AC2=AC1//ACODE
      CALL SBN(N3,IBYTE,ACODE)
      AC3=AC2//ACODE
      I=3
      TEXT=AC3
      END IF
C
      RETURN
      END

      SUBROUTINE MBP(NX,NY,I,TEXT)
C  ASCII CODE FOR MULTIPLE PAIR OF NUMBERS
C
      CHARACTER AC1,AC2,AC3,AC4,AC5,TEXT*5
C
      N=NX
      IF(NY.GE.NX) N=NY
      IF(N.LT.0.OR.N.GT.16383) THEN
      PRINT*,'*** ERROR *** MBP OUT OF RANGE (0 - 16383)'
      END IF
C
      IF(N.LE.3) THEN
      I=1
```

```
      NP1=NY+96+4*NX
      TEXT=CHAR(NP1)
C
      ELSE IF(N.GT.3.AND.N.LE.31) THEN
      I=2
      NX1=NX/2
      NX2=NX-2*NX1
      NP1=NX1+96
      NP2=NY+32*NX2
      AC1=CHAR(NP1)
      CALL SBN(NP2,IBYTE,AC2)
      TEXT=AC1//AC2
C
      ELSE IF(N.GT.31.AND.N.LE.255) THEN
      I=3
      NX1=NX/16
      NX2=NX-16*NX1
      NY2=NY/64
      NY3=NY-64*NY2
      NP1=NX1+96
      NP2=NY2+4*NX2
      NP3=NY3
      AC1=CHAR(NP1)
      CALL SBN(NP2,IBYTE,AC2)
      CALL SBN(NP3,IBYTE,AC3)
      TEXT=AC1//AC2//AC3
C
      ELSE IF(N.GT.255.AND.N.LE.2047) THEN
      I=4
      NX1=NX/128
      NXR=NX-128*NX1
      NX2=NXR/2
      NX3=NXR-2*NX2
      NY3=NY/64
      NY4=NY-64*NY3
      NP1=96+NX1
      NP2=NX2
      NP3=NY3+32*NX3
      NP4=NY4
      AC1=CHAR(NP1)
      CALL SBN(NP2,IBYTE,AC2)
      CALL SBN(NP3,IBYTE,AC3)
      CALL SBN(NP4,IBYTE,AC4)
      TEXT=AC1//AC2//AC3//AC4
C
      ELSE
      I=5
      NX1=NX/1024
      NXR=NX-1024*NX1
      NX2=NXR/16
      NX3=NXR-16*NX2
      NY3=NY/4096
      NYR=NY-4096*NY3
      NY4=NYR/64
      NY5=NYR-64*NY4
      NP1=96+NX1
```

```
      NP2=NX2
      NP3=NY3+4*NX3
      NP4=NY4
      NP5=NY5
      AC1=CHAR(NP1)
      CALL SBN(NP2,IBYTE,AC2)
      CALL SBN(NP3,IBYTE,AC3)
      CALL SBN(NP4,IBYTE,AC4)
      CALL SBN(NP5,IBYTE,AC5)
      TEXT=AC1//AC2//AC3//AC4//AC5
      END IF
C
      RETURN
      END

      SUBROUTINE PMB(NX,NY,I,TEXT)
C
C  PAIR OF MULTIPLE BYTE NUMBER
      CHARACTER TEXT*6,AC1,AC2,AC3,ACX*3,ACY*3,ACODE*3
C
      K=1
      N=NX
10    IF(N.LT.-16384.OR.N.GT.16383) THEN
      PRINT*,'*** ERROR *** PMB OUT OF RANGE (-16384 to 16383)'
      RETURN
      END IF
C
      IF(ABS(N).LE.16.AND.N.LE.15) THEN
      IF(N.LT.0) N=N+32
      NP1=N+64/K
      IF(K.EQ.1) J=1
      IF(K.EQ.2) L=1
      ACODE=CHAR(NP1)
C
      ELSE IF(ABS(N).LE.512.AND.N.LE.511) THEN
      IF(K.EQ.1) J=2
      IF(K.EQ.2) L=2
      IF(N.LT.0) N=N+1024
      N1=N/32
      N2=N-32*N1
      NP1=N1+64/K
      NP2=N2+64/K
      AC1=CHAR(NP1)
      AC2=CHAR(NP2)
      ACODE=AC1//AC2
C
      ELSE
      IF(K.EQ.1) J=3
      IF(K.EQ.2) L=3
      IF(N.LT.0) N=N+16384+16384
      N1=N/1024
      NR=N-1024*N1
      N2=NR/32
      N3=NR-32*N2
      NP1=N1+64/K
      NP2=N2+64/K
```

```
      NP3=N3+64/K
      AC1=CHAR(NP1)
      AC2=CHAR(NP2)
      AC3=CHAR(NP3)
      ACODE=AC1//AC2//AC3
      END IF
C
      IF(K.EQ.1) THEN
      ACX=ACODE
      K=2
      N=NY
      GOTO 10
      END IF
C
      I=J+L
      TEXT=ACX(:J)//ACODE(:L)
      RETURN
      END

      SUBROUTINE MBA(ANGLE,I,TEXT)
C  MULTIPLE-BYTE ANGLE
      CHARACTER TEXT*3,AC1,AC2,AC3
C
      IF(ANGLE.LT.0.AND.ANGLE.GT.360.0) THEN
      PRINT*,'*** ERROR *** MBA OUT OF RANGE (0 - 360)'
      END IF
C
      NP1=0
      IF(ANGLE.GE.180.0) THEN
      NP1=8
      ANGLE=ANGLE-180.0
      END IF
C
      NA=(ANGLE/90)*16384
      NA1=NA/4096
      NP1=NP1+NA1+96
      NR=NA-4096*NA1
      IF(NR.EQ.0) THEN
      TEXT=CHAR(NP1)
      I=1
      ELSE
      NA2=NR/64
      NA3=NR-64*NA2
      AC1=CHAR(NP1)
      CALL SBN(NA2,IBYTE,AC2)
      IF(NA3.EQ.0) THEN
      I=2
      TEXT=AC1//AC2
      ELSE
      CALL SBN(NA3,IBYTE,AC3)
      I=3
      TEXT=AC1//AC2//AC3
      END IF
      END IF
      RETURN
      END
```

APPENDIX 2

OUTLINE OF THE 'HP7221' GRAPHIC ROUTINES AND FUNCTIONS

Module HP7221

Subroutines for driving the Hewelett-Packard 7221C drafting plotter

Author : T. N. Wong

Date : July 1983

This is a library of subroutines for attaching the Hewelett-Packard 7221C drafting plotter to the ICL PERQ computer and driving it using the RS232 interface.

The routines are designed to be incorporated into a Fortran program and should provide all the facilities required to produce good hard-copy output.

Given below are the graphic routines and their parameters and their functions.

OPENHP(NCHANL, HPFILE) : To initialise the plotter, open the channel no. NCHANL for the file HPFILE, which will be used for recording all the pseudo codes for the drawing.

Input Parameters :

NCHANL - integer for the channel number

HPFILE - the name of the output file containing the graphic codes

HPSCALE(SCALE) : To specify the scale of the drawing.

Input Parameters :

SCALE - real number for the designated scale

SHIFTA(XNORIG, YNORIG) : To shift the origin to the desired point with coordinate (XNORIG, YNORIG). Initially, the origin of the drawing is at the lower left hand corner.

Input Parameters :

XNORIG, YNORIG - real variables for the coordinates of the new origin

SHIFTI(XSHIFT, YSHIFT) : To shift the origin to a new position which is at a distance (XSHIFT, YSHIFT) away from the current position.

Input Parameters :

XSHIFT, YSHIFT - real variables for the incremental coordinate distances from the current origin

PEN(K) : To select a pen for drawing. Altogether eight pens are held in the HP7221C plotter.

Input Parameters :

K - integer variable for the number of the pen stall, K should be from 1 to 8.

DRAWA(x, y) : To draw a line from the current position to the specifed position of absolute coordinates (x,y).

Input Parameters :

x, y - real variables for the end point relative to the current axes

MOVEA(x, y) : To move the pen from the current position to the specifed position of absolute coordinates (x,y).

Input Parameters :

x, y - real variables for the end point relative to the current axes

DRAWI(x, y) : To draw a line to the specifed end point of incremental coordinate distances (x,y) away from the current position.

Input Parameters :

x, y - real variables for the incremental coordinate distances from the current position to the end point

MOVEI(x, y) : To move the pen from the current position to the specifed end point which is at a incremental coordinate distances (x,y) away.

Input Parameters :

x, y - real variables for the incremental coordinate distances from the current position to the end point

ARC(xs, ys, xe, ye, xc, yc, ISENSE) : To draw an arc with given starting and ending points, with known centre. The direction of the arc should be given by the code ISENSE.

Input Parameters :

xs, ys - real variables for the absolute coordinates of the start pt.
xe, ye - real variables for the absolute coordinates of the end pt.
xc, yc - real variables for the absolute coordinates of the centre
ISENSE = 1 the arc is drawn in an anticlockwise direction
= -1 the arc is drawn in an clockwise direction

DASFIX(DASH) : To select the fixed dash line mode.

Input Parameters :

DASH - real variable for the length of the dashed line in mm. Input 0.0 for Dash to return to the solid line mode.

DASVAR(TOTAL) : To select the chained dash line mode.

Input Parameters :

TOTAL - real variable for the total length of the line pattern in mm. The mode is fixed at a 4-2-2-4 pattern

ROTATE(ANGLE) : Evokes the rotation option, and rotation will be done in a CCW direction at an angle cumulated by 'ANGLE'.
Input Parameters :
ANGLE - real variable for the incremental angle of rotation

ROTATX : Evokes the rotation option, and rotation will be done in a CCW direction, at an angle equals to the last angle plotted.

CHRSIZ(WID, HT) : To select characters of a certain width and height.
Input Parameters :
WID - real variable for the width of character box
HT - real variable for the height of character

CHRSLT(ANGLE) : To specify the slant angle for the label slant mode.
Input Parameters :
ANGLE - real variable for the slant angle

CHRSTR(TEXT) : To output a string of alphanumeric characters TEXT.
Input Parameters :
TEXT - a string of alphanumeric characters

CHRINT(INT, N) : To output the value of an integer variable.
Input Parameters :
INT - integer variable
N - field width that will be large enough to accomodate all digits of INT including its sign.

CHRFON(NA, NB) : To select the standard and alternative fonts from font numbers 0 to 5.
Input Parameters :
NA - integer from 0-5 for the standard font number
NB - integer from 0-5 for the alternative font number

FONTSD : To set the standard font as the current font.

FONTAL : To set the alternative font as the current font.

CLOSHP : To terminate and close the plot.

APPENDIX 3

NC PART PROGRAM FORMAT FOR THE MORI-FANUC CNC LATHE SYSTEM

1. BASIC PROGRAM PATTERN

2. G CODE LIST

3. M CODE LIST

Basic Program Pattern

When making part programs for individual cutting tools (O.D. cutting tool, thread cutting tool, etc.), the following basic patterns must be strictly observed.

Ø0001		Program No.	
N001	G50	X, Z (absolute coordinates), S300 (Max. spindle rpm with clamping; rpm does not exceeds 300)	
N002	G00	T0101 Tool offset #01 Tool station #01	Spindle speed range M41 (low speed) M42 (high speed)
N003	G96	S150 (cutting speed 150 m/min)	M03 (spindle normal rotation)
	G97	S150 (spindle speed 150 rpm)	M04 (spindle reverse rotation)
N004	(G00)	Approach to workpiece in rapid feed	
N005	G01	Approach to workpiece in slightly faster cutting feed (for safety reason)	
		↑ O.D. and I.D. cutting Thread cutting Grooving etc. ↓	
N050	G00	X, Z (The numerals to be commanded in this block must be identical to those in G50 block.)	
N051		T0100 Tool offset cancel	

G Code List

G Code	Function
G00	Positioning
G01	Linear interpolation
G02	Circular interpolation (Counterclockwise)
G03	Circular interpolation (Clockwise)
G04	Dwell
G20	Inch data input
G21	Metric data input
G27	Zero return check
G28	Zero return
G29	Return from zero
G32	Thread cutting
G40	Tool tip R compensation cancel
G41	Tool tip R compensation right
G42	Tool tip R compensation left
G50	Absolute coordinates preset, Max. spindle speed setting
G70	Finish cycle
G71	O.D., I.D. rough cutting cycle
G72	End surface rough cutting cycle
G73	Closed loop cutting cycle
G74	End surface cutting off cycle
G75	O.D., I.D. rough cutting cycle
G76	Thread cutting cycle
G90	Cutting cycle A
G92	Thread cutting cycle
G94	Cutting cycle B
G96	Surface speed constant control
G97	Surface speed constant control cancel
G98	Feed per time
G99	Feed per spindle rotation

M Code List

Code	Function		Description	
M00	Program temporary		* Spindle stop, cutting oil supply stop, feed stop * Used to stop machine operation during program * Operation can be resumed by auto start button	
M01	Optional stop		* ON/OFF of M01 can be controlled by switch on control panel	
M02	Program end		* Used at the end of an entire program * NC unit is reset, stopping all operations of machine	
M03	Spindle rotation	Normal	* Viewed from tail stock	CCW
M04		Reverse		CW
M05	Spindle stop			
M08	Cutting oil supply	Discharge		
M09		Stop		
M10	Chuck	Clamping	* Effective only when M00 Delete in ON	
M11		Unclamping	* Effective operation is obtained when bar feeder is used	
M12	Tail stock spindle	OUT		
M13		IN		
M17	Tool post rotation	Normal		
M18		Reverse		
M21	Tail stock direction	Forward		
M22		Backward		

Code	Function		Description
M23	Chamfer-ing	Clamping	* Effective for thread cutting cycle (G92)
M24		Unclamping	* Finishing-up of thread cutting is performed
M30	Reset and rewind		
M41	Spindle	Low speed range	* Low/high speed range switching is possible while spindle is rotating
M42		High speed range	* These codes are not used on models without low/ high speed range classification
M73	Parts catcher	OUT	
M74		IN	
M98	Sub-tape call from main tape		
M99	Main-tape call from sub-tape		

N.B. Two M functions cannot be commanded in a single block

APPENDIX 4

PROGRAM LISTING OF THE POSTFANUC PROGRAM

```
      TRACE 2
      SUBROUTINE RINIT
C THIS ROUTINE IS TO INITIALIZE THE ROLLER DATA FILE
C
      INTEGER NEL(100),ITYPE(100),IEL(100),IDIR(100)
      REAL RXA1(100,7),RXA(100,7),GRAD(100)
C
      COMMON /BK2/NEL,ITYPE,IDIR,RXA,RXA1,IEL,GRAD

      DO 100 I=1,100
      NEL(I)=0
      ITYPE(I)=0
      IDIR(I)=0
      IEL(I)=0
      GRAD(I)=0.0
      DO 105 J=1,7
      RXA1(I,J)=0.0
105   RXA(I,J)=0.0
100   CONTINUE
      RETURN
      END
      TRACE 2
      SUBROUTINE RDECODE
C THIS SUBROUTINE IS TO DECODE THE ROLL CONTOUR DATA FILE AND
C OUTPUT THE CORRESPONDING GEOMETRIC DESCRIPTIONS FOR THE
C FAPT PART PROGRAM
C
      INTEGER NEL(100),ITYPE(100),IDIR(100),IEL(100)
      REAL RXA1(100,7),RXA(100,7),GRAD(100)
C
      COMMON /BK1/PASHT,CTOC,NSTAGE
      COMMON /BK2/NEL,ITYPE,IDIR,RXA,RXA1,IEL,GRAD
C
      READ(1,10) ICON
10    FORMAT(I0)
      IF(ICON.EQ.1) WRITE(2,21)NSTAGE
      IF(ICON.EQ.2) WRITE(2,22)NSTAGE
      IF(ICON.EQ.3) WRITE(2,23)NSTAGE
      IF(ICON.EQ.4) WRITE(2,24)NSTAGE
21    FORMAT(///,1X,'$ STAGE ',I2,4X,'TOP ROLL')
22    FORMAT(///,1X,'$ STAGE ',I2,4X,'BOTTOM ROLL')
23    FORMAT(///,1X,'$ STAGE ',I2,4X,'LEFT SIDE ROLL')
24    FORMAT(///,1X,'$ STAGE ',I2,4X,'RIGHT SIDE ROLL')
C
      DO 100 I=1,100
      READ(1,30)NEL(I),ITYPE(I),(RXA(I,J),J=1,7),IDIR(I)
30    FORMAT(2I0,7F0.0,I0)
C
      IF (NEL(I).EQ.0) RETURN
      IF (ICON.NE.1) GOTO 101
      RXA(I,2)=CTOC-(PASHT+RXA(I,2))
      RXA(I,4)=CTOC-(PASHT+RXA(I,4))
```

```
      IF (ITYPE(I).EQ.2) RXA(I,6)=CTOC-(PASHT+RXA(I,6))
      GOTO 105
C
101   IF (ICON.NE.2) GOTO 102
      RXA(I,2)=RXA(I,2)+PASHT
      RXA(I,4)=RXA(I,4)+PASHT
      IF(ITYPE(I).EQ.2) RXA(I,6)=RXA(I,6)+PASHT
      GOTO 105
C
102   IF (ICON.EQ.3.OR.ICON.EQ.4) GOTO 103
      IF (I.EQ.1) WRITE(2,31)
31    FORMAT(1X,'$ INVALID INPUT OF ROLL CONTOUR TYPE')
      GOTO 100
C
103   DO 104 J=2,6,2
104   RXA(I,J)=ABS(RXA(I,J))
C
105   IF (ITYPE(I).NE.1) GOTO 106
      IF (RXA(I,1).EQ.RXA(I,3).AND.RXA(I,2).EQ.RXA(I,4)) GOTO 107
C
      IF(RXA(I,3).NE.RXA(I,1)) GOTO 115
      GRAD(I)=9999.999
      GOTO 116
115   GRAD(I)=(RXA(I,4)-RXA(I,2))/(RXA(I,3)-RXA(I,1))
116   IF (NEL(I).EQ.1) GOTO 108
      IF (ITYPE(I-1).EQ.2) GOTO 108
      IF (ABS(GRAD(I)-GRAD(I-1)).GE.0.001) GOTO 108
109   IEL(I)=IEL(I-1)
      GOTO 100
108   IEL(I)=NEL(I)
      DO 200 K=2,4,2
      RXA1(I,K-1)=RXA(I,K-1)
200   RXA1(I,K)=-2*RXA(I,K)
      WRITE(2,41)IEL(I),(RXA1(I,J),J=1,4)
41    FORMAT(1X,'S',I2,'=P(',F8.3,',',F8.3,'D),P(',
     %F8.3,',',F8.3,'D)')
      GOTO 100
C
106   IF(ITYPE(I).EQ.2) GOTO 110
      WRITE(2,32)NEL(I)
32    FORMAT(1X,'$ INVALID ELEMENT TYPE,ELEMENT NO.  ',I2)
      GOTO 100
C
110   IF(NEL(I).EQ.1) GOTO 111
      IF((RXA(I,5).EQ.RXA(I-1,5)).AND.
     %(RXA(I,6).EQ.RXA(I-1,6)).AND.
     %(RXA(I,7).EQ.RXA(I-1,7))) GOTO 109
C
111   IEL(I)=NEL(I)
      DO 250 K=5,7,2
250   RXA1(I,K)=RXA(I,K)
      RXA1(I,6)=-2*RXA(I,6)
      WRITE(2,42)IEL(I),(RXA1(I,J),J=5,7,1)
42    FORMAT(1X,'C',I2,'=P(',F8.3,',',F8.3,'D),',F8.3)
      GOTO 100
C
107   IF(NEL(I).EQ.1) GRAD(I)=9988.9898
      IF(NEL(I).EQ.1) GOTO 108
      IF(ITYPE(I-1).EQ.1) GOTO 109
      ITYPE(I)=ITYPE(I-1)
```

```
      DO 120 J=5,7,1
120   RXA(I,J)=RXA(I-1,J)
      GOTO 109
C
100   CONTINUE
      RETURN
      END
      TRACE 2
      SUBROUTINE CONFAPT
C THIS ROUTINE MANIPULATES THE DATA FROM THE ROLL CONTOUR
C DATA FILE AND ESTABLISHES THE GEOMETRIC DESCRIPTIONS FOR
C A COMPACT 2 PART PROGRAM
C
      COMMON /BK1/PASHT,CTOC,NSTAGE
C
      READ(1,10)NSTAGE,PASHT,CTOC,OSTOL,OSTOR
10    FORMAT(I0,4F0.0)
      READ(1,11)XDATUM,ORGXP,ORGYP,SCALE,ILEFT,IRIGHT
11    FORMAT(4F0.0,2I0)
C
      IF(ILEFT.EQ.0.AND.IRIGHT.EQ.0) J=2
      IF(ILEFT.EQ.1.AND.IRIGHT.EQ.1) J=4
      IF(ILEFT.EQ.1.AND.IRIGHT.EQ.0) J=3
      IF(ILEFT.EQ.0.AND.IRIGHT.EQ.1) J=3
C
      DO 100 I=1,J
      CALL RINIT
      CALL RDECODE
100   CONTINUE
      RETURN
      END
      TRACE 2
      MASTER POSTFANUC
C THIS IS TO OUTPUT A GEOMETRIC DESCRIPTION FILE IN FAPT
C
      INTEGER OUNIT
      READ(1,10) PIE,IUNIT,OUNIT,JTOTAL
10    FORMAT(F0.0,3I0)
      DO 100 I=1,JTOTAL
      CALL CONFAPT
100   CONTINUE
      STOP
      END
      FINISH
```

APPENDIX 5

CAD/CAM SYSTEM INFORMATION

1. LIST OF THE PROCESSING COMMAND FILES

The command files stored in the PERQ fixed disc memory, corresponding to the processing commands for executing different stages of the CAD/CAM system software.

2. FILES FOR THE CAD/CAM SOFTWARE SYSTEM

List of files containing all source programs, compiled files and run job files, with regard to different command files.

3. DATA INPUT FORMAT

Explanatory notes for input data items in the Finished Section Program, Flower Patter Program, Roll Design Program and Roll Machining Program.

PROCESSING COMMAND FILES FOR THE CAD/CAM SOFTWARE

Command File	Content
SECTION.Cmd	RUN :USER>R1PLOT
SECTIONX.Cmd	RUN :USER>NR1PLOT
FLOWER.Cmd	RUN :USER>R2PLOT
FLOWERX.Cmd	RUN :USER>NR2PLOT
TEMPLATE.Cmd	RUN :USER>R3PLOT
ROLL.Cmd	RUN :USER>R4PLOT
ROLLX.Cmd	RUN :USER>NR4PLOT
TAPE.Cmd	RUN :USER>CUTPLOT RUN :USER>MFPOST
TAPEX.Cmd	RUN :USER>NCUTPLOT RUN :USER>MFPOST
CHKTAPE.Cmd	RUN :USER>CHKTAP

FILES FOR THE CAD/CAM SYSTEM SOFTWARE

COMMAND FILE	RUN FILE	COMPILED OBJECT FILE	FORTRAN SOURCE FILE
SECTION.Cmd	R1PLOT.Run	R1plot.Seg R1aplot.Seg FramPrint.Seg	R1plot.For R1aplot.For FramPrint.For
SECTIONX.Cmd	NR1PLOT.Run	NR1plot.Seg NR1aplot.Seg NFramPrint.Seg	NR1plot.For NR1aplot.For NFramPrint.For
FLOWER.Cmd	R2PLOT.Run	R2plot.Seg Decofra.Seg FramPrint.Seg	R2plot.For Decofra.For FramPrint.For
FLOWERX.Cmd	NR2PLOT.Run	NR2plot.Seg NDecofra.Seg FramPrint.Seg	NR2plot.For NDecofra.For FramPrint.For
TEMPLATE.Cmd	R3PLOT.Run	R3plot.Seg R3aPlot.Seg R3bPlot.Seg	R3plot.For R3aPlot.For R3bPlot.For
ROLL.Cmd	R4PLOT.Run	R4plot.Seg R4b.Seg Roll6mx.Seg R3bplot.Seg R1surfmx.Seg R2surfmx.Seg R5m1plot.Seg R5m2plot.Seg R5m3plot.Seg	R4Plot.For R4b.For Roll6mx.For R3bplot.For R1surfmx.For R2surfmx.For R5m1plot.For R5m2plot.For R5m3plot.For
ROLLX.Cmd	NR4PLOT.Run	NR4plot.Seg R4b.Seg Roll6mx.Seg R3bplot.Seg R1surfmx.Seg R2surfmx.Seg R5m1plot.Seg R5m2plot.Seg R5m3plot.Seg	NR4Plot.For R4b.For Roll6mx.For R3bplot.For R1surfmx.For R2surfmx.For R5m1plot.For R5m2plot.For R5m3plot.For

COMMAND FILE	RUN FILE	COMPILED OBJECT FILE	FORTRAN SOURCE FILE
ROEDIT.Cmd	ROEDIT.Run	Roedit.Seg Roeda.Seg Roedb.Seg Roedc.Seg Roedd.Seg Roede.Seg (HP7221.Seg)	Roedit.For Roeda.For Roedb.For Roedc.For Roedd.For Roede.For
TAPE.Cmd	CUTPLOT.Run	Cutplot.Seg Cut1plot.Seg Cut2plot.Seg TLIB.Seg PlotHP.Seg (HP7221.Seg)	Cutplot.For Cut1plot.For Cut2plot.For TLIB.For PlotHP.For
	MFPOST.Run	MFpost.Seg MFapost.Seg	MFpost.For MFapost.For
TAPEX.Cmd	NCUTPLOT.Run	NCutplot.Seg Cut1plot.Seg NCut2plot.Seg TLIB.Seg PlotHP.Seg (HP7221.Seg)	NCutplot.For Cut1plot.For NCut2plot.For TLIB.For PlotHP.For
	MFPOST.Run	MFpost.Seg MFapost.Seg	MFpost.For MFapost.For
CHKTAPE.Cmd	CHKTAP.Run	Chktap.Seg	Chktap.For

DATA INPUT FORMAT
=================

During the execution of the CAD/CAM programs, the user will be led, by the question/answer type dialogue, to supply data input. It is, thus, not necessary to prepare data input according to the strict format. However, all data input should be encoded in accordance with the man-machine dialogue, and the input data sequences for differenct programs are given in the following :

A. FINISHED SECTION PROGRAM

INPUT DATA SEQUENCE

IUNIT = INPUT DIMENSION UNIT (1 = INCH & 2 = MM.)
OUNIT = OUTPUT DIMENSION UNIT (1 = INCH & 2 = MM.)
THICK = THICKNESS OF STRIP
ORIGIN = STARTING POINT FOR FIRST ELEMENT

* 1 = AT LEFT, MOVING RIGHT ONLY
* 2 = AT RIGHT, MOVING LEFT ONLY
* 3 = AT CENTRE, MOVING RIGHT, THEN LEFT(NON-SYMMETRICAL)
* 4 = AT CENTRE, MOVING LEFT,THEN RIGHT(NON-SYMMETRICAL)
* 5 = AT CENTRE, MOVING RIGHT ONLY(SYMMETRICAL)

NEXT ENTER DEFINITION STATEMENTS FOR ELEMENT DEFINITION SEQUENCE, EACH STATEMENT CONSISTS OF 4 DATA VALUES AS FOLLOWS :-

SEQUENCE NUMBER (N)	ELEMENT TYPE (TYPE)	LENGTH OR RADIUS	ANGLE OF BENDING
(INTEGER)	(INTEGER)	(REAL)	(REAL)
START FROM 1, INCREMENT BY 1 UP TO 50, TO TERMINATE ENTER 0.	1 = LINEAR 2 = CIRCULAR THE REST ARE ILLEGAL.	POSITIVE LENGTH FOR TYPE 1, INSIDE RADIUS FOR TYPE 2 (MAY BE 0).	FOR TYPE 2 ELEMENTS ONLY, ZERO FOR TYPE 1 NON-ZERO FOR TYPE 2 *

*THE EXCEPTION BEING, WHEN N=1 AND TYPE=1,
ANGLE OF BENDING IS TAKEN AS THE ANGLE OF INCLINATION
BETWEEN THE AXIS OF THE FIRST LINEAR ELEMENT
AND THE HORIZONTAL AXIS. (RANGE = -90.0 TO +90.0, IN DEGREES)

NOTE THAT WHEN ORIGIN IS 3 OR 4 (DOUBLE SIDED DEFINITION),
2 SETS OF DEFINITION SEQUENCE ARE REQUIRED, ONLY 1 SET
IS REQUIRED WHEN ORIGIN IS 1,2 OR 5.
FIRST ELEMENT AND LAST ELEMENT OF THE SEQUENCE MUST BE LINEAR.
MAXIMUM NO. OF ELEMENTS IN EACH SEQUENCE MUST BE LESS THAN 50.

NEXT ENTER DATA VALUES FOR PAPERSIZE AND SCALE AS FOLLOWS :-

```
PAPERSIZE        SCALE
---------        -----
(INTEGER)        (REAL)
```

PAPERSIZE= 3 (FOR A3 SIZE) OR 4 (FOR A4 SIZE),
SCALE= ANY POSITIVE VALUE OF MAGNITUDE LESS THAN THE GIVEN LIMITS

NEXT ENTER DATA FOR DIMENSIONING OPTION, 1 IF YES OR 0 IF NO.

NEXT ENTER DATA FOR TITLE-BLOCK INPUT SELECTION OPTION :-
1 IF TITLE-BLOCK INFORMATION WILL BE SUPPLIED SUBSEQUENTLY, OR
0 IF NO TITLE-BLOCK INFORMATION WILL BE SUPPLIED
(IF 1 REFER TO SUBROUTINE DTVAL FOR FURTHER INFORMATION REGARDING
TITLE-BLOCK INPUTS)

B. FLOWER PATTERN PROGRAM

INPUT DATA SEQUENCE

---

1. JRUN = 1 (IF FLOWER PATTERNS ONLY), OR
   2 (IF ROLLER-PLOTTINGS ONLY), OR
   3 (IF FLOWER-PATTERNS AND ROLLER-PLOTTINGS).

2. NSTAGE = TOTAL NUMBER OF STAGES
   (THE PERMISSIBLE RANGE IS FROM 1 TO 50)

3. JBEND = SELECTION OF CIRCULAR-ELEMENT BENDING DEFINITION OPTION:-
   0 (IF SIMPLE ELEMENT DEFINITION), OR
   1 (IF COMPOSITE PERCENTAGE ELEMENT DEFINITION).

4A. ISEQ = BENDING STAGE SEQUENCE NUMBER (LARGEST OF WHICH = NSTAGE).
IELML = LEFT-HAND ELEMENT (CIRCULAR) TO BE BENT
XANGL = CUMULATIVE ANGLE OF BENDING FOR THE LEFT-HAND ELEMENT.
IELMR = RIGHT-HAND ELEMENT (CIRCULAR) TO BE BENT.
XANGR = CUMULATIVE ANGLE OF BENDING FOR THE RIGHT-HAND ELEMENT.

4B. ** THIS IS THE COMPOSITE PERCENTAGE ELEMENT DEFINITION OPTION **

ISEQ = (AS IN NOTE 4A)
IELML = (AS IN NOTE 4A)
XANGL = (AS IN NOTE 4A)
IPCL1 = PERCENTAGE LENGTH OF LEADING LINEAR PART.
IPCL2 = PERCENTAGE LENGTH OF CIRCULAR PART.
IPCL3 = PERCENTAGE LENGTH OF TRAILING LINEAR PART.
IELMR = (AS IN NOTE 4A)
XANGR = (AS IN NOTE 4A)
IPCR1 = PERCENTAGE LENGTH OF LEADING LINEAR PART.
IPCR2 = PERCENTAGE LENGTH OF CIRCULAR PART.
IPCR3 = PERCENTAGE LENGTH OF TRAILING LINEAR PART.

(NOTE THAT THE IPCL'S AND THE IPCR'S SHOULD BE INTEGER VALUES IN THE RANGE 0 TO 100 AND THE SUM SHOULD BE 100 EXACTLY IN EACH CASE).

N.B. :- TO TERMINATE THE DEFINITION SEQUENCE FOR EITHER 4A OR 4B, JUST ENTER 0 FOR ISEQ AND DUMMY VALUES FOR THE REST.

5. ** THIS IS THE RADII-SHARPENING OPTION **

JSHARP = 1 (IF STAGE NO. ARE TO BE ENTERED INDIVIDUALLY), OR
-1 (IF ALL STAGES EXCEPT LAST STAGE REQUIRE SHARPENING OF RADII), OR
0 (IF RADII-SHARPENING IS NOT REQUIRED AT ALL).

N.B. :- IF JSHARP IS 0 OR -1, NO FURTHER INPUT DATA IS REQUIRED FOR THIS OPTION, PROCEED TO THE NEXT OPTION INPUT.

ISQSH = STAGE NO. WITH RADII-SHARPENING ACTIVE.

6. ** THIS IS THE FIXED PERCENTAGE COMPOSITE LENGTH OPTION **

```
ISQCP = ENTRY NO. WITH FIXED PERCENTAGE COMPOSITE LENGTHS
IECL1 = L.H.S. ELEMENT NO. WITH SUCH COMPOSITION.
IECL2 = STAGE NO. WHEN IT STARTS TO BE ACTIVE.
IECL3 = STAGE NO. WHEN IT CEASES TO BE ACTIVE.
IECR1 = R.H.S. ELEMENT NO. WITH SUCH COMPOSITION.
IECR2 = STAGE NO. WHEN IT STARTS TO BE ACTIVE.
IECR3 = STAGE NO. WHEN IT CEASES TO BE ACTIVE.

N.B. :- IF ONLY ELEMENT ON ONE-SIDE IS DESIRED, THEN ENTER 0 FOR
        ELEMENT NO. ON THE IRRELEVANT SIDE.
        TO TERMINATE THIS SEQUENCE, ENTER 0 FOR ALL VALUES IN A LINE.
```

C. ROLL DESIGN PROGRAM

INPUT DATA SEQUENCE

R1. IPASHT = +1 (IF CONSISTENT PASS-HEIGHT & C-TO-C DISTANCE),OR
-1 (IF INCONSISTENT PASS-HEIGHTS & C-TO-C DISTANCES)

R2. PASHT = PASS-HEIGHT (OR RADIUS OF THE BOTTOM-ROLL)
CTOC = CENTRE-TO-CENTRE DISTANCE BETWEEN TOP & BOTTOM ROLLS

N.B.:- GIVE ONLY ONE VALUE EACH OF IPASHT=+1 ,OTHERWISE
GIVE THE CORRECT NUMBER (= NSTAGE) OF VALUES EACH IF IPASHT=-1;
BOTH PASHT & CTOC VALUES MUST BE POSITIVE.

R3. ITOL =+1 (IF CONSISTENT LH & RH TOLERANCES),OR
-1 (IF INCONSISTENT LH & RH TOLERANCES).

R4. TOLLH = LEFT-HAND TOLERANCE BETWEEN ROLLER AND LAST ELEMENT
TOLLR = RIGHT-HAND TOLERANCE BETWEEN ROLLER AND LAST ELEMENT.

N.B.:- GIVE ONLY ONE VALUE EACH IF ITOL=+1, OTHERWISE
GIVE THE CORRECT NUMBER (= NSTAGE) OF VALUES EACH
IF ITOL=-1; BOTH TOLLH AND TOLRH MUST BE POSITIVE.

R5. JTEMP = 1 IF COMPONENT DRAWING IN HIDDEN-LINE FORM
IS REQUIRED, OTHERWISE 0

RSCALE = DESIRED SCALE FOR THE ROLLER DRAWINGS (AUTOMATICALLY
REDUCED IF IT EXCEEDS THE MAXIMUM PERMISSIBLE SCALE)

RGAP = THE GAP DIMENSION BETWEEN TOP AND BOTTOM ROLLERS

R6. JSTAGE(1)=-1 (IF ALL STAGES REQUIRED FOR ROLLERS),OR
JSTAGE(1)= 0 (IF NO STAGES REQUIRED FOR ROLLERS),OR
JSTAGE(1)=ANY POSITIVE VALUE(INTEGER LESS THAN NSTAGE) (IF
SELECTED STAGES REQUIRED FOR ROLLERS).

N.B.:- IF JSTAGE(1) IS -1 OR 0, NO FURTHER INPUT DATA IS REQUIRED;
IF JSTAGE(1) IS POSITIVE,THEN SUPPLY OTHER DESIRED
BENDING STAGE NO. IN ASCENDING ORDER, ENTER 0 TO TERMINATE.
THE TOTAL NUMBER OF JSTAGE DATA MUST NOT EXCEED
NSTAGE VALUE.

```
***********************************************************************

    SPECIAL OPTIONS

***********************************************************************

    PINCH-DIFFERENCE OR DRIVE/CLEARANCE SURFACE OPTION:-
    ===================================================

 P1. JPINCH = 0 (IF NO EXTERNAL PINCH DIFFERENCE DIMENSION
                 DEFINITION IS SUPPLIED), OR
              1 (IF ONLY ONE PINCH DIFFERENCE DIMENSION DEFINITION
                 IS SUPPLIED FOR ALL BENDING STAGES), OR
             -1 (IF PINCH DIFFERENCE DIMENSION DEFINITION IS SUPPLIED
                 INDIVIDUALLY FOR EACH BENDING STAGE).

              2 (IF SINGLE THICKNESS OPTION WITH ONE COMMON CLEARANCE
                 VALUE FOR ALL STAGES SELECTED),OR

             -2 (IF SINGLE THICKNESS OPTION WITH INDIVIDUAL CLEARANCE
                 VALUE FOR EACH STAGE SELECTED)

    N.B. :- IF JPINCH IS 0,1 OR -1 THEN PROCEED TO SECTION P2.
            IF JPINCH IS 2 OR -2 THEN PROCEED TO SECTION P2A.


 P2. PDIM1 = THICKNESS BETWEEN THE DRIVE SURFACES.
     PDIM2 = THICKNESS BETWEEN THE CLEARANCE SURFACES.

    N.B. :- IF JPINCH IS 0, DATA FOR PDIM1 AND PDIM2 ARE NOT REQUIRED;
            IF JPINCH IS 1, ONLY 1 SET IS REQUIRED; AND
            IF JPINCH IS -1, 1 SET FOR EACH STAGE SHOULD BE SUPPLIED.

 P3. ISQP = CURRENT BENDING STAGE SEQUENCE NO.
     IEPL = L.H.S. ELEMENT DEFINING THE DRIVE SURFACE CONTOUR.
     IEPR = R.H.S. ELEMENT DEFINING THE DRIVE SURFACE CONTOUR.

    N.B. :- TO TERMINATE DEFINITION STATEMENT SEQUENCE, ENTER 0 FOR ISQP,
            AND DUMMY VALUES FOR THE OTHER ITEMS.
            TWO STATEMENTS REQUIRED FOR EACH BENDING STAGE
            TO DEFINE THE BEGINNING AND THE END OF DRIVE-SURFACE.
            TOTAL NO. OF DEFINITION STATEMENT SHOULD BE 50 OR LESS.
            IF ONLY EITHER L.H.S. OR R.H.S. IS TO BE DEFINED THEN THE
            NON-APPLICABLE SIDE SHOULD HAVE ELEMENT 0 ENTERED.

 P2A. **  THIS IS THE SINGLE THICKNESS OPTION FOR H&E **
      CLEAR = CLEARANCE VALUE TO BE ADDED ONTO THE THICKNESS WITH
              8 PER CENT OF THICKNESS ALREADY ADDED .

    N.B. :- IF JPINCH IS 2, THEN SUPPLY ONLY ONE VALUE;
            IF JPINCH IS -2, THEN SUPPLY THE CORRECT NUMBER (=NSTAGE)
            OF VALUES FOR EACH INDIVIDUAL STAGE.
```

SIDE-CONTOUR OPTION:-
===================

S1. ISROL = 1 (IF SIDE-ROLL OPTION WILL BE USED), OR
0 (IF SIDE-ROLL OPTION WILL NOT BE USED)

N.B. IF ISROL IS 0, THEN NO FURTHER DATA IS REQUIRED FOR OTHER SIDE-ROLL INPUT AND EXTENSION-CONTOUR INPUT.

S2. IOTOS = 1 (IF CONSISTENT ORIGIN TO SIDE-ROLL AXES) OR
-1 (IF INCONSISTENT ORIGIN TO SIDE-ROLL AXES DISTANCES)

S3. OTOSL = DISTANCE BETWEEN ORIGIN AND LEFT SIDE-ROLL CENTRE
OTOSR = DISTANCE BETWEEN ORIGIN AND RIGHT SIDE-ROLL CENTRE

(SEE DIAGRAM).

N.B. :- GIVE ONLY 1 SET OF VALUES IF IOTOS IS 1; IF IOTOS IS -1 THEN GIVE CORRECT NUMBER (=NSTAGE) OF SETS OF VALUES FOR EVERY STAGE; BOTH OTOSL AND OTOSR MUST BE POSITIVE.

S4. ISQS = CURRENT BENDING STAGE SEQUENCE NO.
IESL1 = L.H.S. ELEMENT CONSTITUTING L.H.S. SIDE-ROLL CONTOUR.
IESL2 = WHICH FACE OF THE L.H.S. ELEMENT IS DESIRED
IESR1 = R.H.S. ELEMENT CONSTITUTING R.H.S. SIDE-ROLL CONTOUR.
IESR2 = WHICH FACE OF THE R.H.S. ELEMENT IS DESIRED

(SEE DIAGRAM FOR DETAIL OF ELEMENT FACE DEFINITION).

N.B. :- IF SIDE-ROLLS ARE NOT REQUIRED FOR A PARTICULAR BENDING STAGE, THEN SKIP AND GO ON TO DEFINE THE NEXT STAGE WHICH HAS SIDE-ROLLS.
THE SIDE-ROLL CONTOUR IS DEFINED BY 2 BOUNDARY ELEMENT FACES, NAMELY THE STARTING ELEMENT FACE AND THE ENDING ELEMENT FACE. THEY MUST BE ENTERED IN THAT SEQUENCE IN 2 CONSECUTIVE DEFINITION STATEMENTS.
THE DEFINITION CONVENTION IS FROM BOTTOM UPWARDS, HENCE THE COMPLETE SIDE-ROLL CONTOUR WILL CONSIST OF THE BOTTOM LEADING PART, FOLLOWED BY THE PART IN CONTACT WITH THE STRIP AND THEN FOLLOWED BY THE TRAILING TOP PART.
NO MORE THAN 2 DEFINITION STATEMENTS SHOULD BE ENTERED FOR EACH SIDE-ROLL.
TO TERMINATE DEFINITION STATEMENT SEQUENCE, ENTER 0 FOR ISQS,AND DUMMY VALUES FOR THE OTHER ITEMS.
IF ONLY EITHER L.H.S. OR R.H.S. ELEMENT IS TO BE DEFINED IN THE CURRENT STATEMENT, THEN ENTER 0 FOR THE ELEMENT ON THE NON-APPLICABLE SIDE.
TOTAL NO. OF DEFINITION STATEMENTS SHOULD BE 50 OR LESS.

S5. ** THIS SECTION ON EXTENSION-CONTOUR OPTION CAN BE USED
ONLY IF SIDE-ROLL OPTION IS USED, ELSE IT IS IRRELEVANT **

IEXTN = 1 (IF EXTENSION-CONTOUR OPTION WILL BE USED), OR
0 (IF EXTENSION-CONTOUR OPTION WILL NOT BE USED).

N.B. :- IF IEXTN IS 0, THEN NO FURTHER DATA INPUT FOR THIS
OPTION IS REQUIRED.

S6. ISC1 = SIDE-CONTOUR NO.
ISC2 = SIDE-CONTOUR TYPE (1 FOR SIDE-ROLL, 2 FOR SPIGOT TYPE A,
3 FOR SPIGOT TYPE B.)

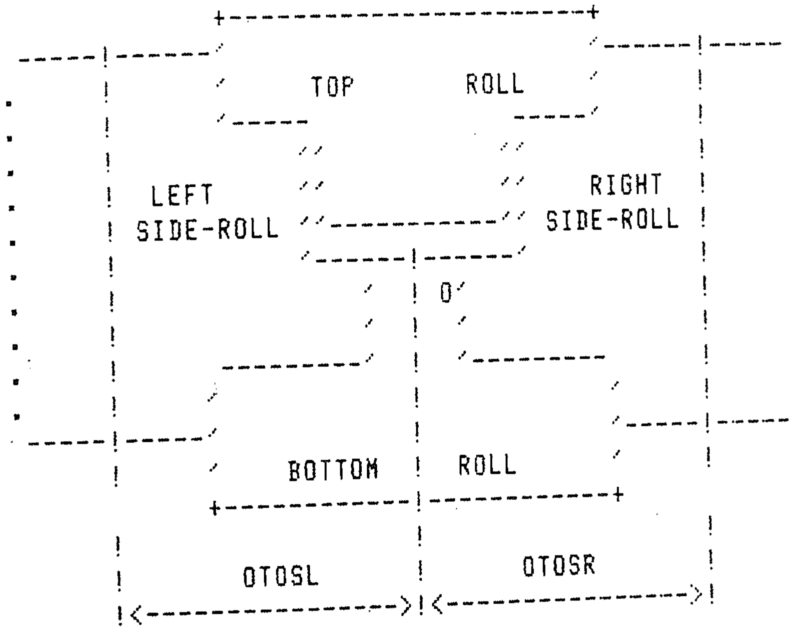
SCDIM1 = DIMENSION DEFINITION FOR SIDE-CONTOUR PART 1
SCDIM2 = DIMENSION DEFINITION FOR SIDE-CONTOUR PART 2
SCDIM3 = DIMENSION DEFINITION FOR SIDE-CONTOUR PART 3
SCDIM4 = DIMENSION DEFINITION FOR SIDE-CONTOUR PART 4

(SEE DIAGRAM FOR DETAILS OF EACH DEFINITON).

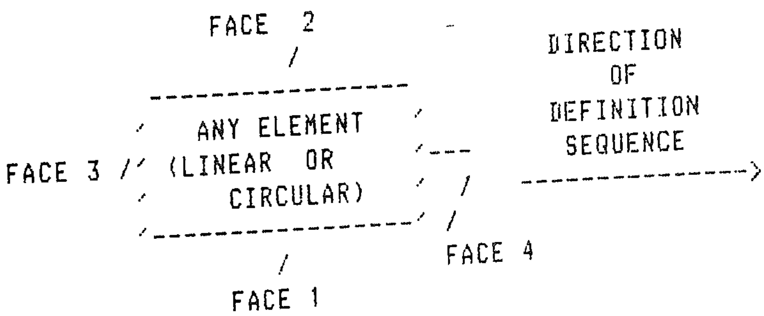
S7. ISQD = BENDING STAGE NO. FOR WHICH SIDE-CONTOUR DEFINITION
IS INTENDED
IELD1 = TYPE OF SIDE-CONTOUR ON L.H.S.
IELD2 = SIDE-CONTOUR NO. WHICH HAS BEEN DEFINED PREVIOUSLY AND
IS TO BE SELECTED FOR THIS STAGE NO. ON L.H.S.
IERD1 = TYPE OF SIDE-CONTOUR ON R.H.S.
IERD2 = SIDE-CONTOUR NO. WHICH HAS BEEN DEFINED PREVIOUSLY AND
IS TO BE SELECTED FOR THIS STAGE NO. ON R.H.S.
N.B. :- THE SIDE-CONTOUR TYPE MUST MATCH WITH THE DEFINED TYPE
DESCRIBED IN SECTION S2.
IF THE SELECTED CONTOUR TYPE IS 1 (I.E. SIDE-ROLL),
THEN 2 DEFINITION STATEMENTS FOR BOTTOM LEADING PART AND
TOP TRAILING PART RESPECTIVELY MUST BE ENTERED IN
SUCCESSION WHEREAS IF THE SELECTED CONTOUR TYPE IS 2 OR 3
(I.E. SPIGOTS),THEN ONLY 1 DEFINITION STATEMENT IS REQUIRED.

DIAGRAMS FOR ROLL INPUT DATA NOTES IN ROLL DESIGN
****

(1) DISTANCES BETWEEN ORIGIN AND SIDE-ROLL AXES



(2) ELEMENT FACES FOR DEFINING SIDE-ROLL CONTOURS



WHERE FACE 1 IS BOTTOM FACE,
FACE 2 IS TOP FACE,
FACE 3 IS LEADING FACE AND
FACE 4 IS TRAILING FACE.

(3) PARTS FOR DEFINING EXTENSION-CONTOURS
=====================================

TYPE 1 :- SIDE-ROLL EXTENSION-CONTOURS

```
                                                  !
                                                  !  PART 4
                                         PART 3   !
(TOP-ROLL SIDE)                         ------------
                                        !
                                PART 2  !
                                        !
                                       **
                                     **
                            PART 1 **  (INCLINED AT +30 DEGREES)


                            PART 1 **  (INCLINED AT -30 DEGREES)
                                     **
                                       **
                                        !
                                PART 2  !
                                        !
(BOTTOM-ROLL SIDE)                      ------------
                                          PART 3  !
                                                  !  PART 4
                                                  !
```

TYPE 2 :- SPIGOT TYPE A

```
                              PART 2
                            ----------
                          **          !
                 PART 1  **           !  PART 3
                       **             !
                     **               !
                                      ----------
                                        PART 4
```

TYPE 3 :- SPIGOT TYPE B

```
                                        PART 4
                                      ----------
                     **               !
                       **             !
                 PART 1  **           !  PART 3
                           **         !
                            ----------
                              PART 2
```

(N.B. :- THE IRRELEVANT PARTS OF THE EXTENSION-CONTOURS MAY HAVE
THEIR LENGTH MADE ZERO)

D. ROLL MACHINING PROGRAM

INPUT DATA FORMAT (CUTTING PATH)

C1. NSTAGE = STAGE NO. OF THE ROLL
ICON = ROLLER NO. (1 FOR TOP ROLL, 2 FOR BOTTOM ROLL, 3 FOR LEFT SIDE ROLL, AND 4 FOR RIGHT SIDE ROLL)

C2. ICYCLE = TYPE OF CUTTINT CYCLE CHOSEN (1 FOR ROUGHING, 2 FOR GROOVING, 3 FOR FINISH PROFILING, AND 4 FOR POCKETING)
ISEL = THE NO. OF THE STARTING ELEMENT OF THE MACHINING AREA
IEEL = ELEMENT NO. OF THE ENDING ELEMENT OF THE MACHINING AREA
TOL = MATERIAL TOLERANCE LEFT (MM)

N.B. :- TERMINATE INPUT TO THIS PROGRAM BY TYPING 4 0'S
IF ICYCLE = 1 OR 2, THEN PROCEED TO SECTION C3.

C2A. (FOR ICYCLE = 3 OR 4)
ZP = THE Z-COORDINATE OF THE CUTTING START POINT ON THE DESIGNATED CUT-IN LINE
XP = THE X-COORDINATE OF THE CUTTING START POINT ON THE DESIGNATED CUT-IN LINE
ANG = ANGLE OF THE CUT-IN LINE (+VE FOR CCW)

C3. NTOOL = THE TOOL NO. (TURRET POSITION). IF THE TOOL IS NOT CURRENTLY IN THE TOOL LIBRARY, ENTER NTOOL = 0, AND ENTER DETAILS OF THE TOOL IN SUBGROUP C4, FOR NTOOL = 1 TO 12, SKIP C.4
DCUT = DEPTH OF CUT (MM), FOR ICYCLE =2 OR 3, ENTER DCUT=0.0
SPEED = CUTTING SPEED IN M/MIN
FEED = FEEDRATE IN MM/REV

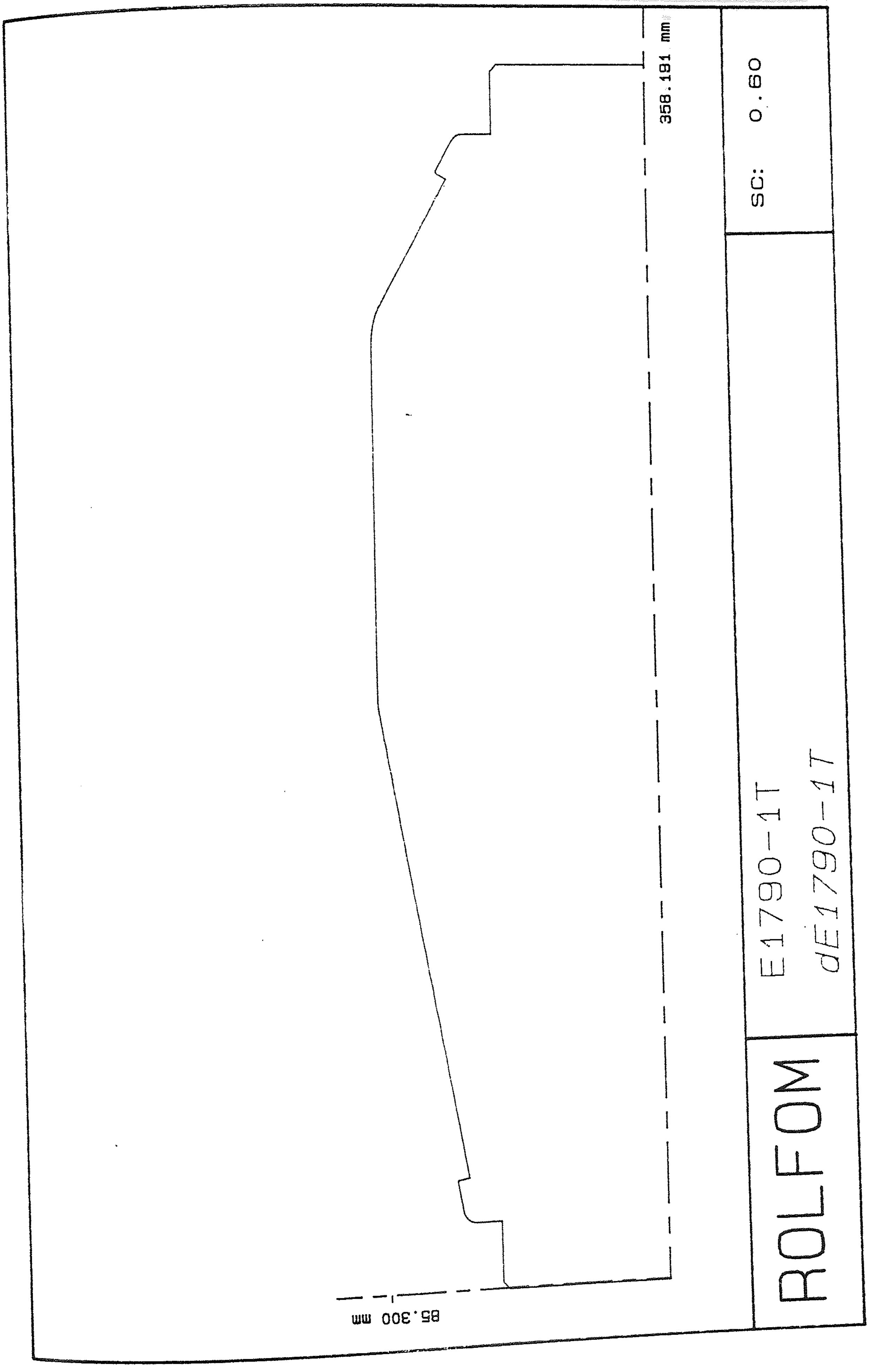
N.B. :- IF NTOOL = 0, PROCEED TO SECTION C4, ELSE GO TO C2.

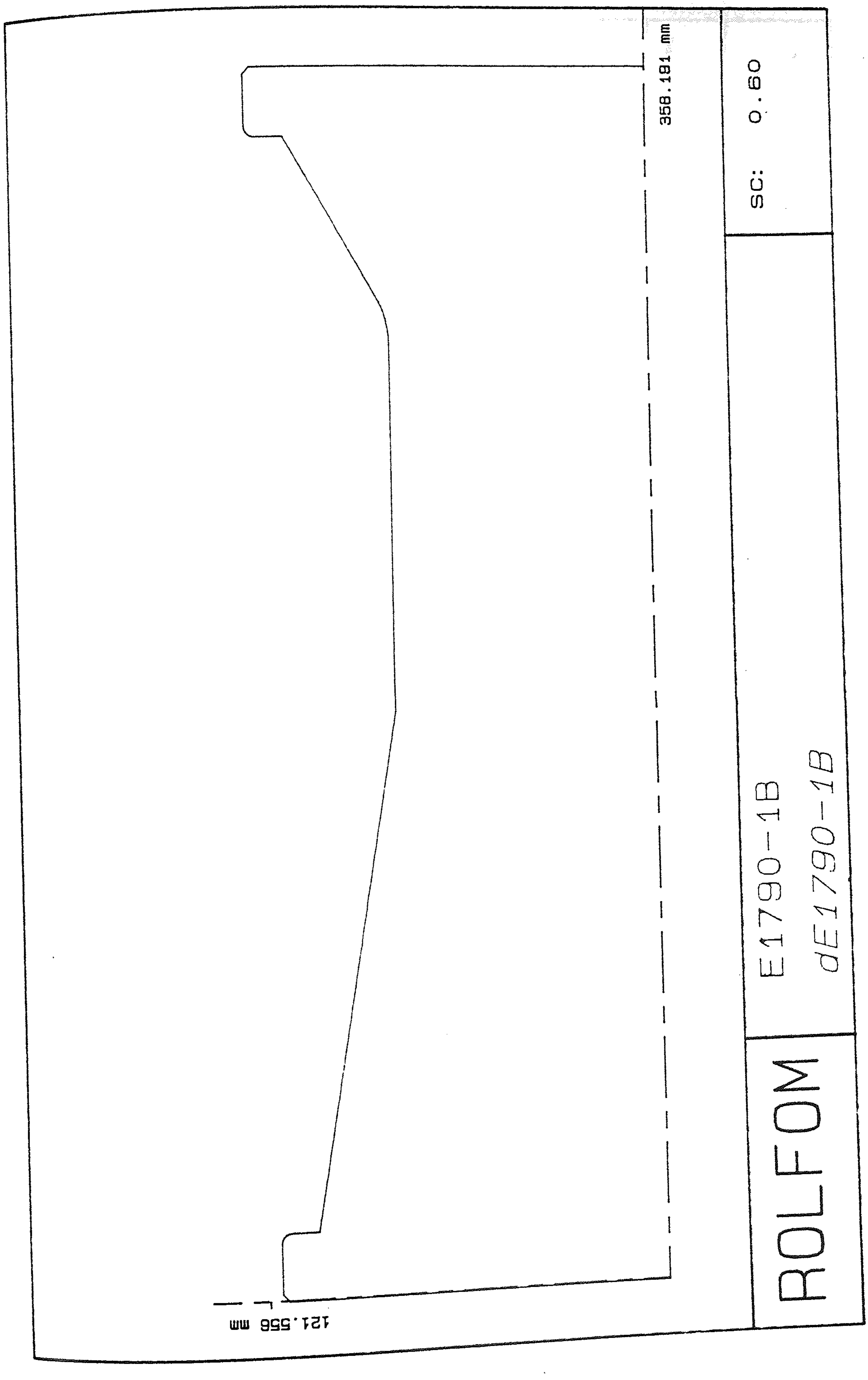
C4. (FOR DEFINING TOOL GEOMETRY OF NON-STANDARD CUTTING TOOL)
NTOOL = TURRET POSITION (1-12)
RAD = TOOL TIP CIRCLE RADIUS
RZ = Z-COORDINATE OF THE TOOL TIP CIRCLE CENTRE
RX = X-COORDINATE OF THE TOOL TIP CIRCLE CENTRE
WID = WIDTH OF PARTING-OFF TOOL, ENTER 0.0 FOR OTHER TOOLS
ZREF = Z-COORDINATE OF THE TOOL OFFSET POSITION
XREF = X-COORDINATE OF THE TOOL OFFSET POSITION

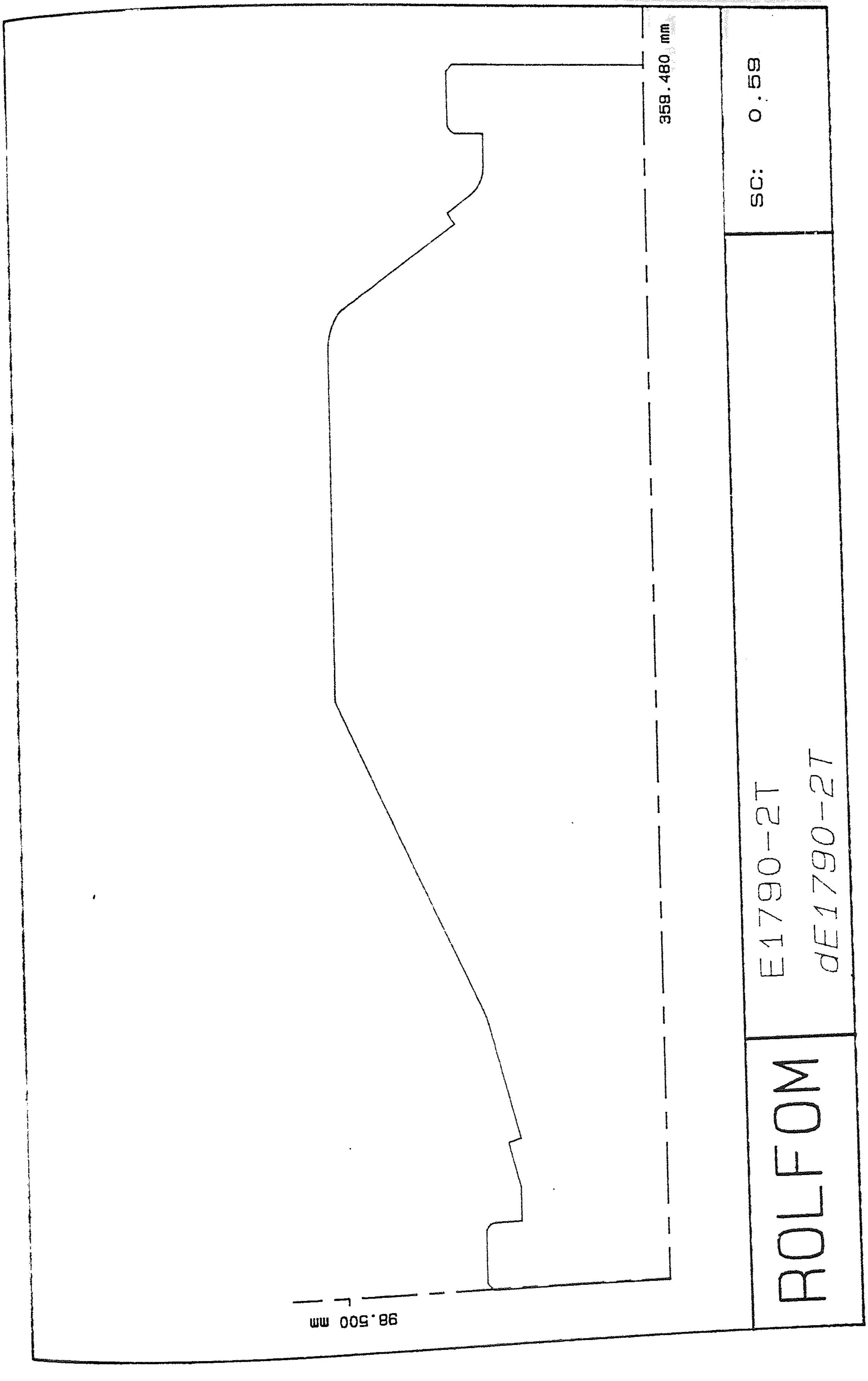
APPENDIX 6

A SET OF ROLL PROFILE DRAWINGS GENERATED DURING THE JOB RUN

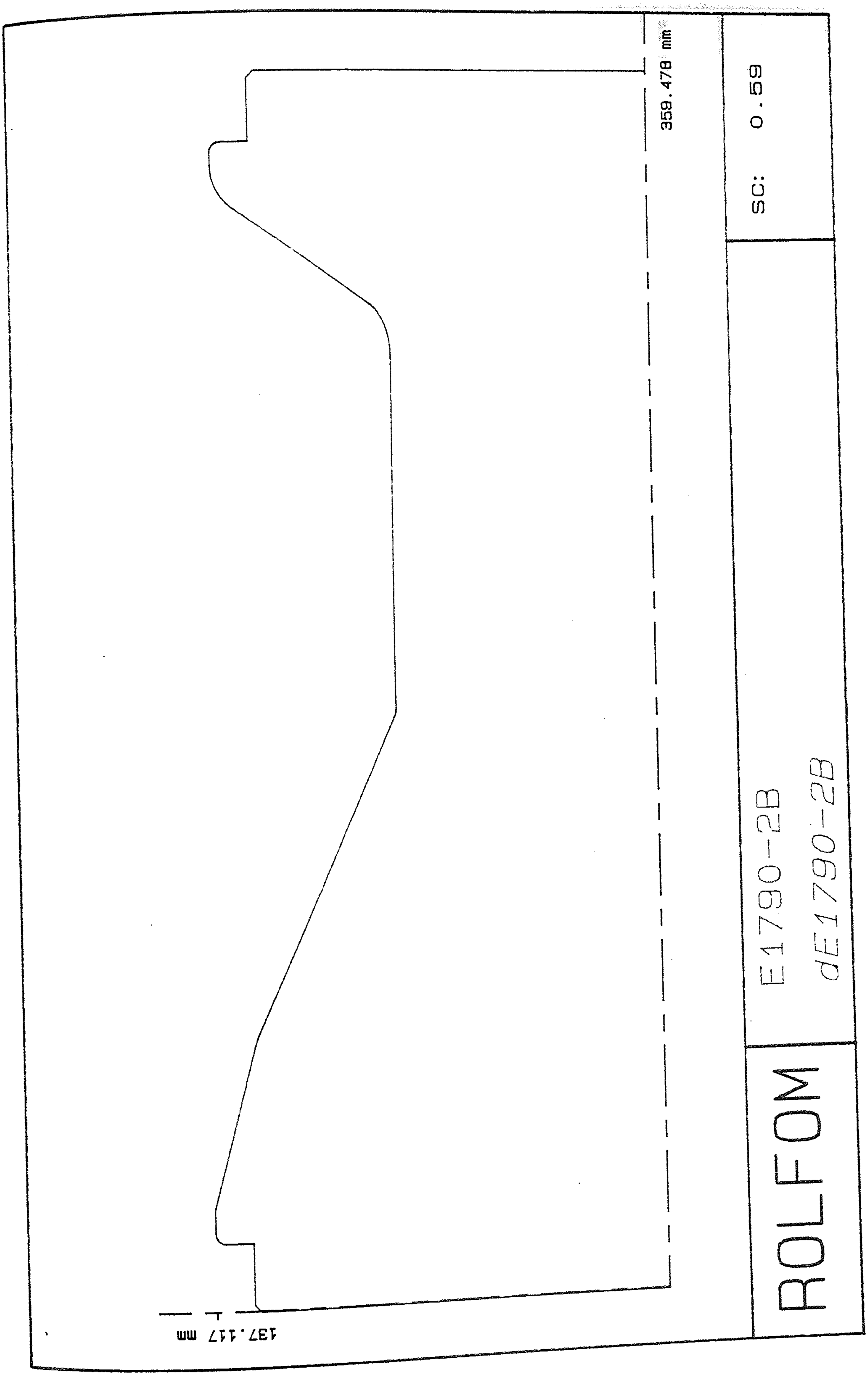
(NB : Figures 2.3 and 2.5 are the Finished Section Drawing and the Flower Pattern Drawing for this job)


358.181 mm
SC: 0.60
E1790-1T
dE1790-1T
ROLFOM
85.300 mm

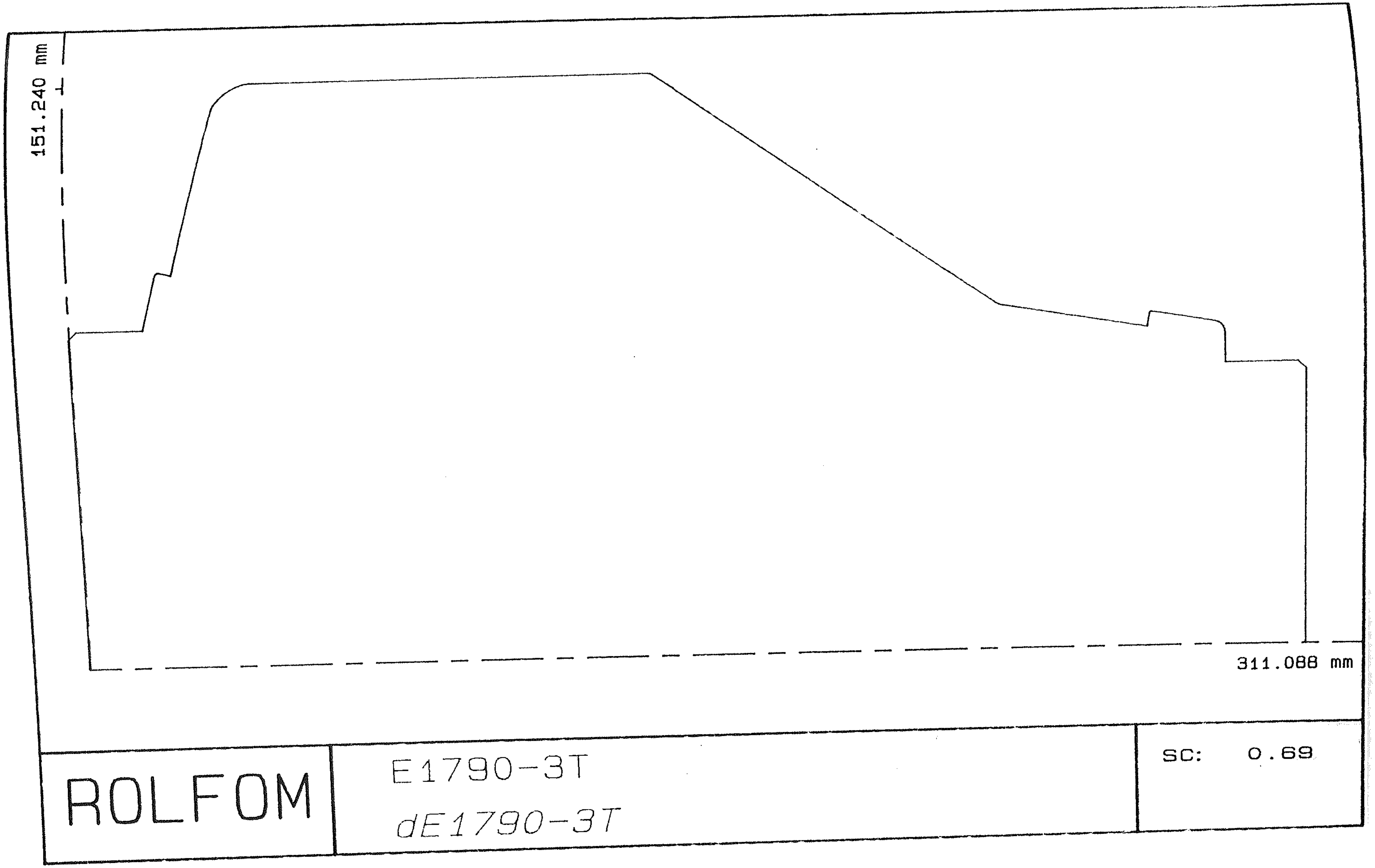

358.181 mm
121.556 mm
SC: 0.60
E1790-1B
dE1790-1B
ROLFOM



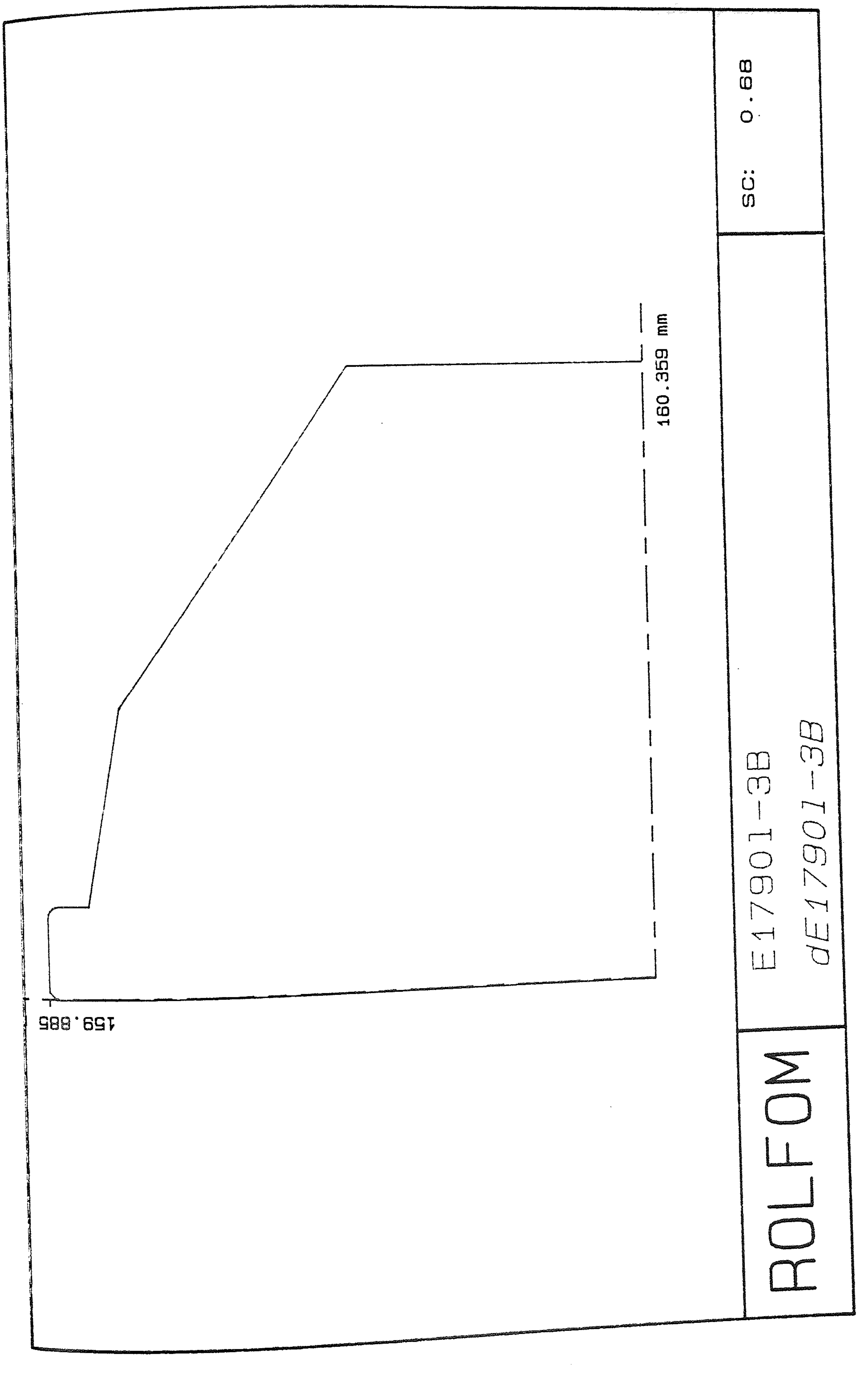
359.480 mm
SC: 0.59
E1790-2T
dE1790-2T
ROLFOM
98.500 mm

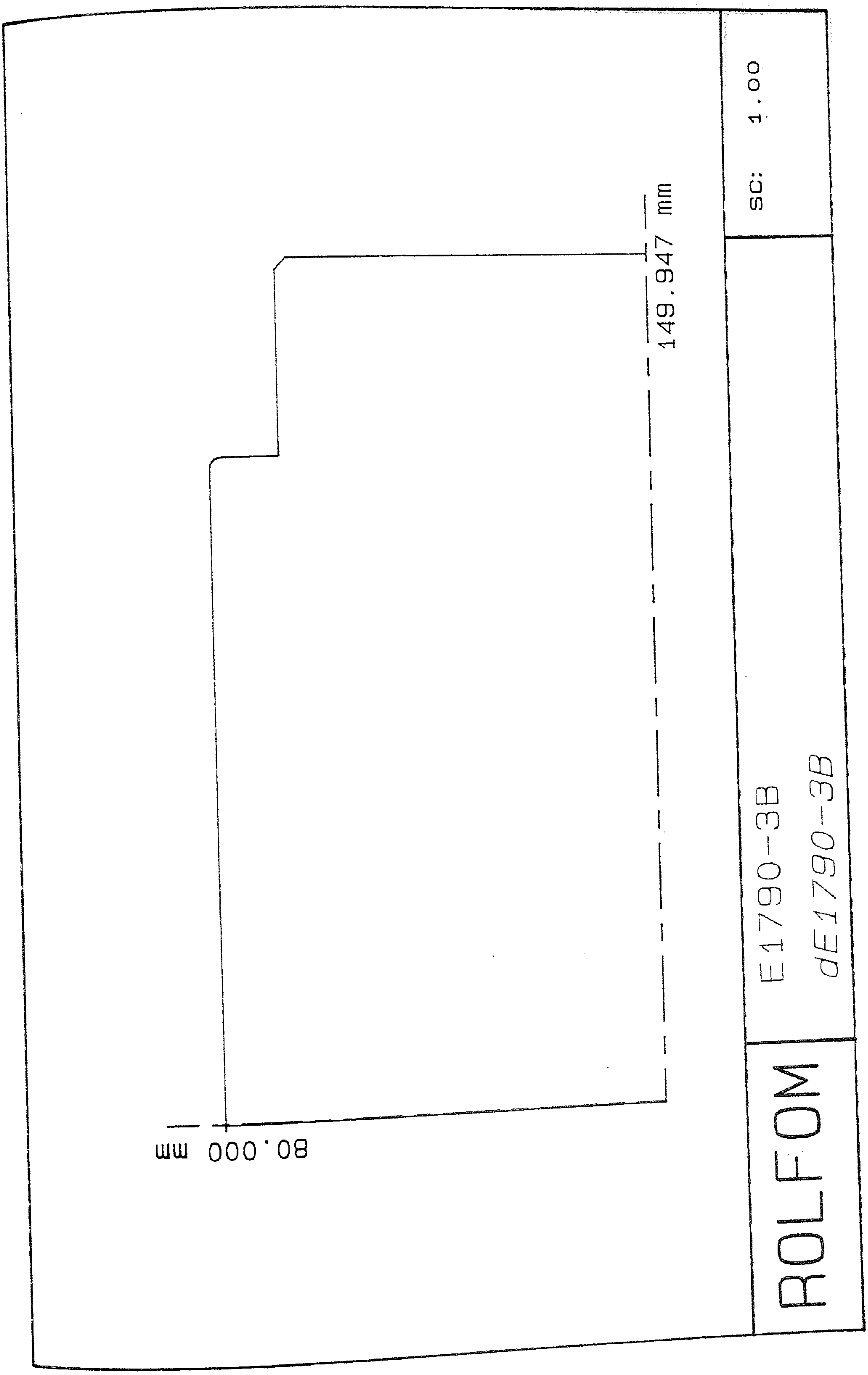


359.478 mm
SC: 0.59
E1790-2B
dE1790-2B
ROLFOM
137.117 mm

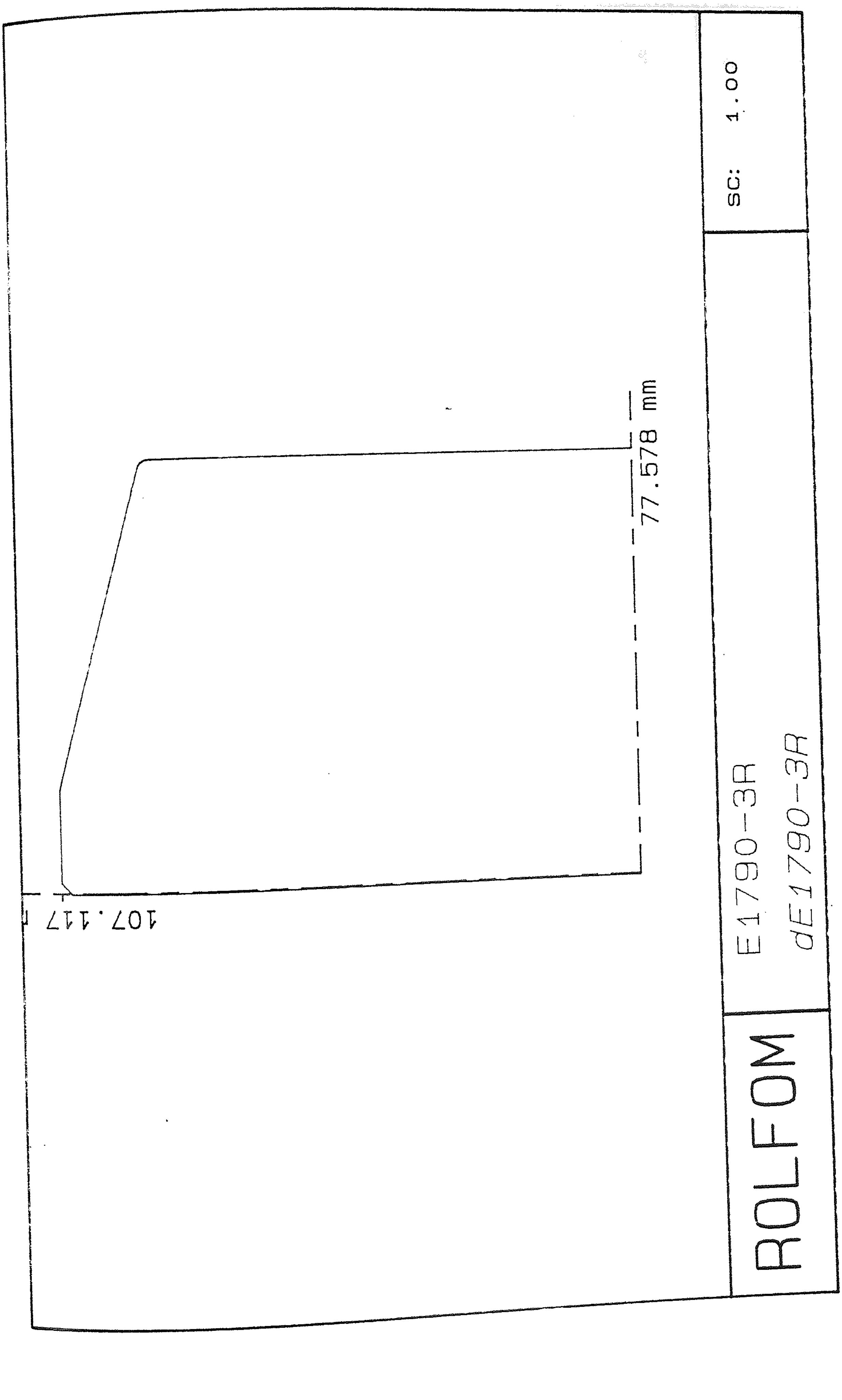


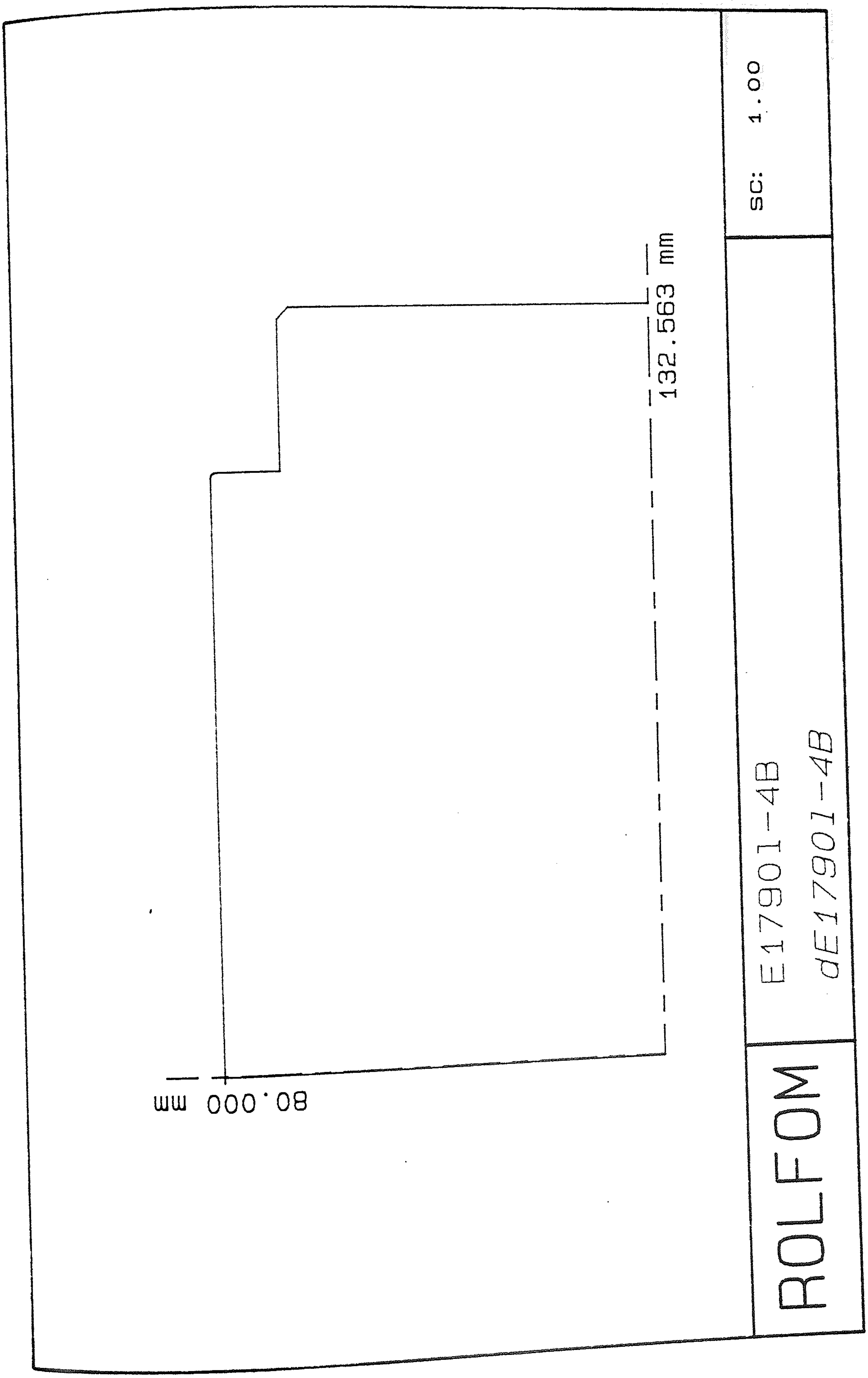
151.240 mm
311.088 mm
ROLFOM
E1790-3T
dE1790-3T
SC: 0.69

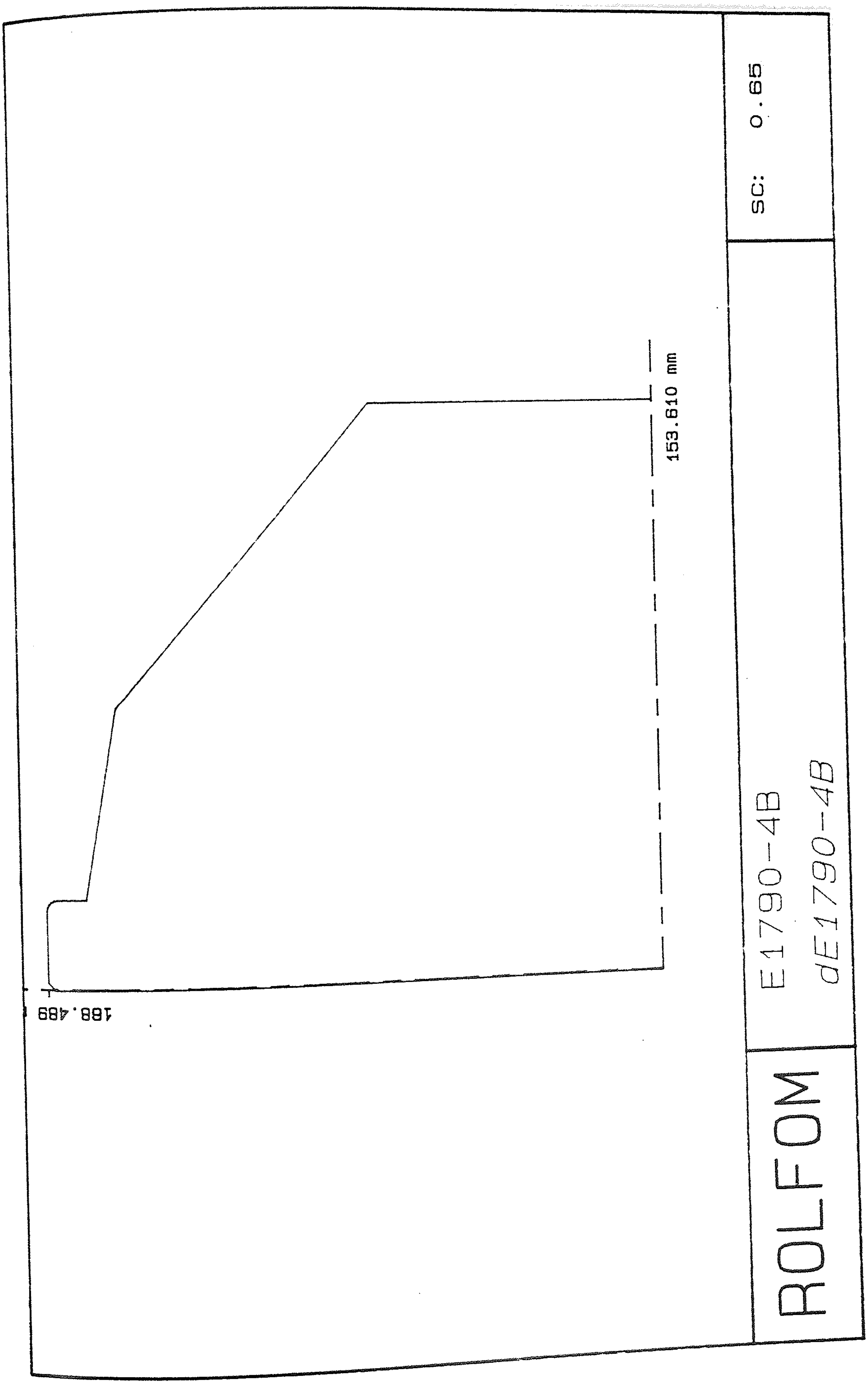

ROLFOM
E17901-3B
dE17901-3B
SC: 0.68
160.359 mm
159.885



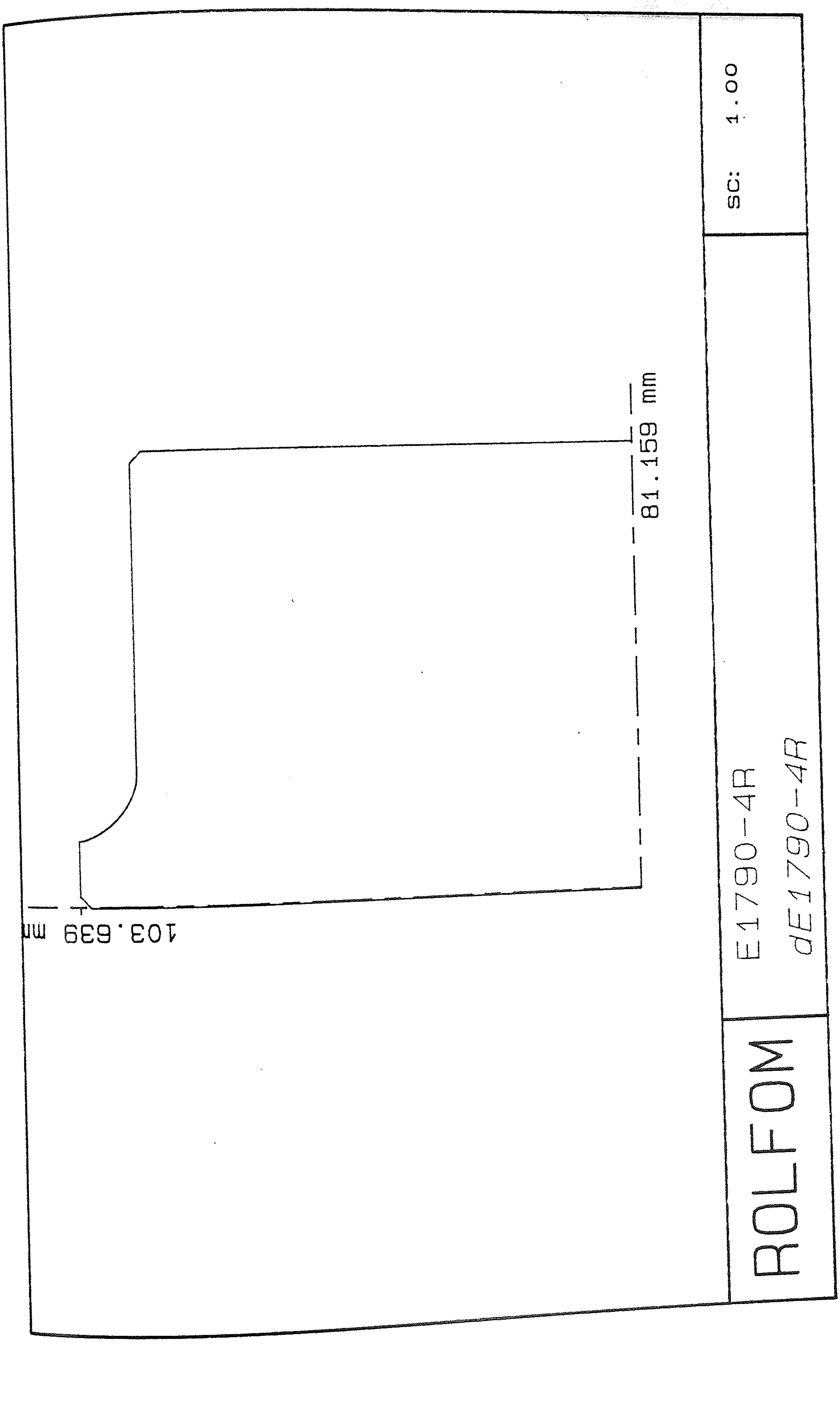
149.947 mm
80.000 mm
sc: 1.00
ROLFOM
E1790-3B
dE1790-3B

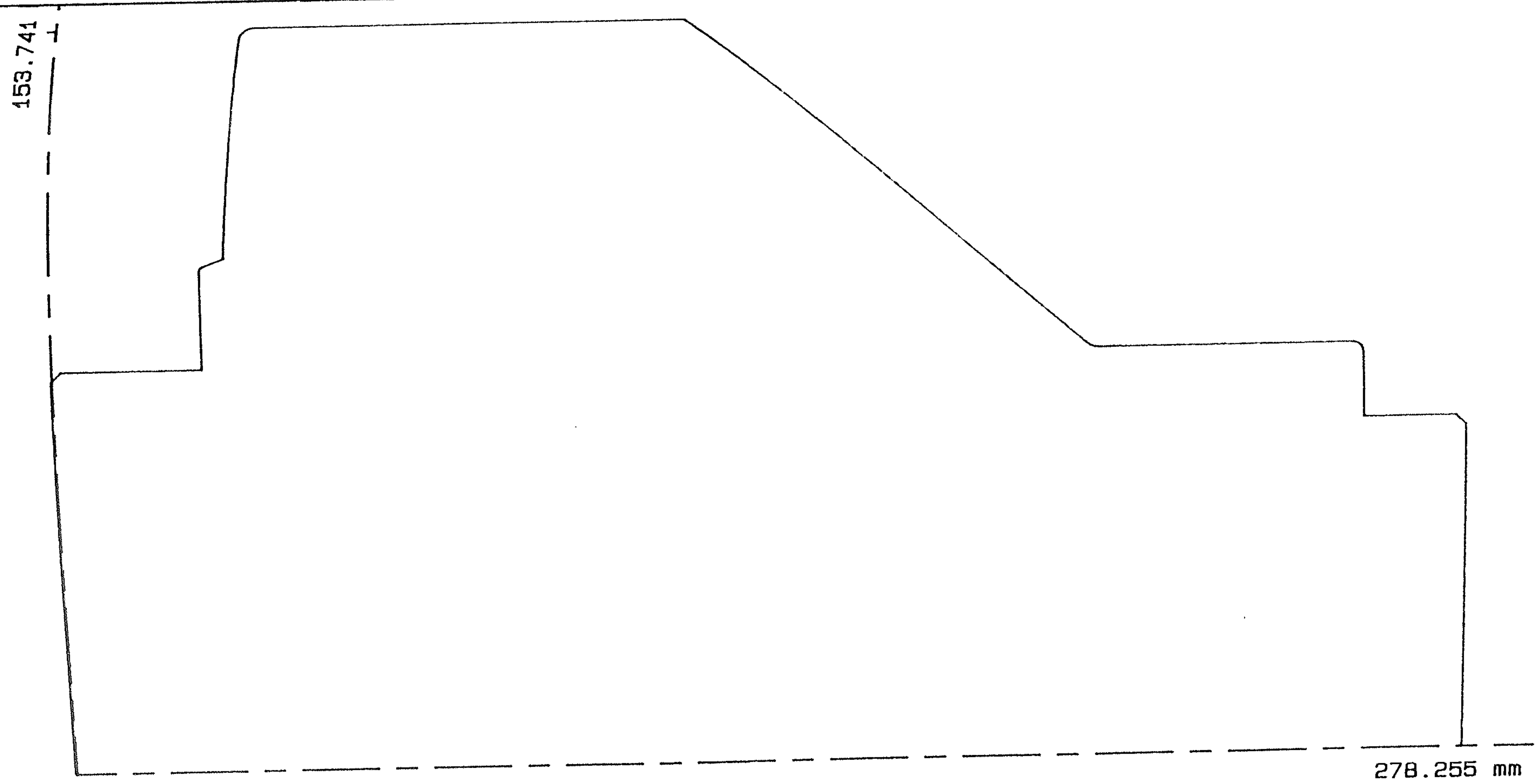

ROLFOM
E1790-3R
dE1790-3R
SC: 1.00
77.578 mm
107.117


132.563 mm
80.000 mm
SC: 1.00
E17901-4B
dE17901-4B
ROLFOM

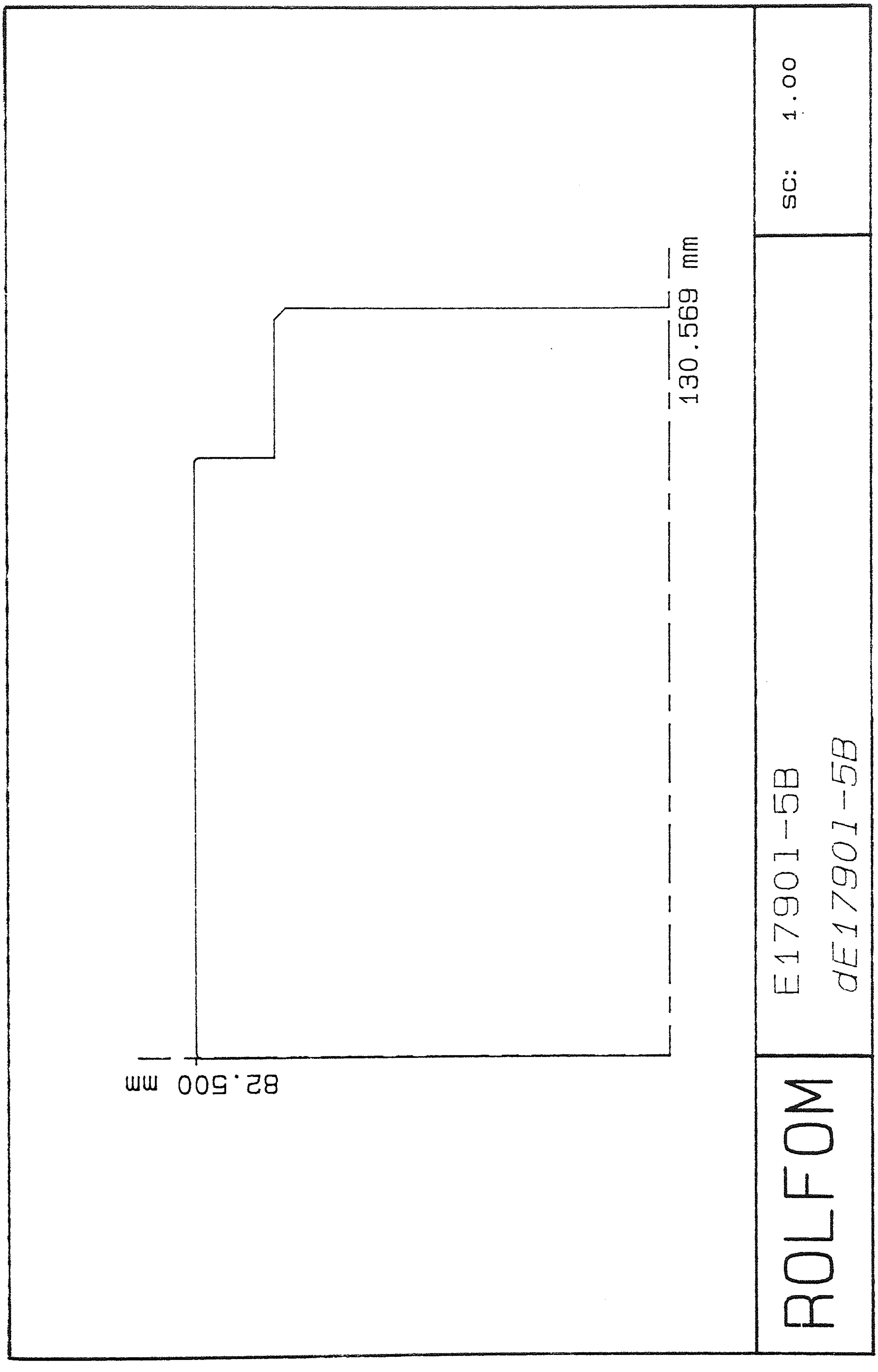


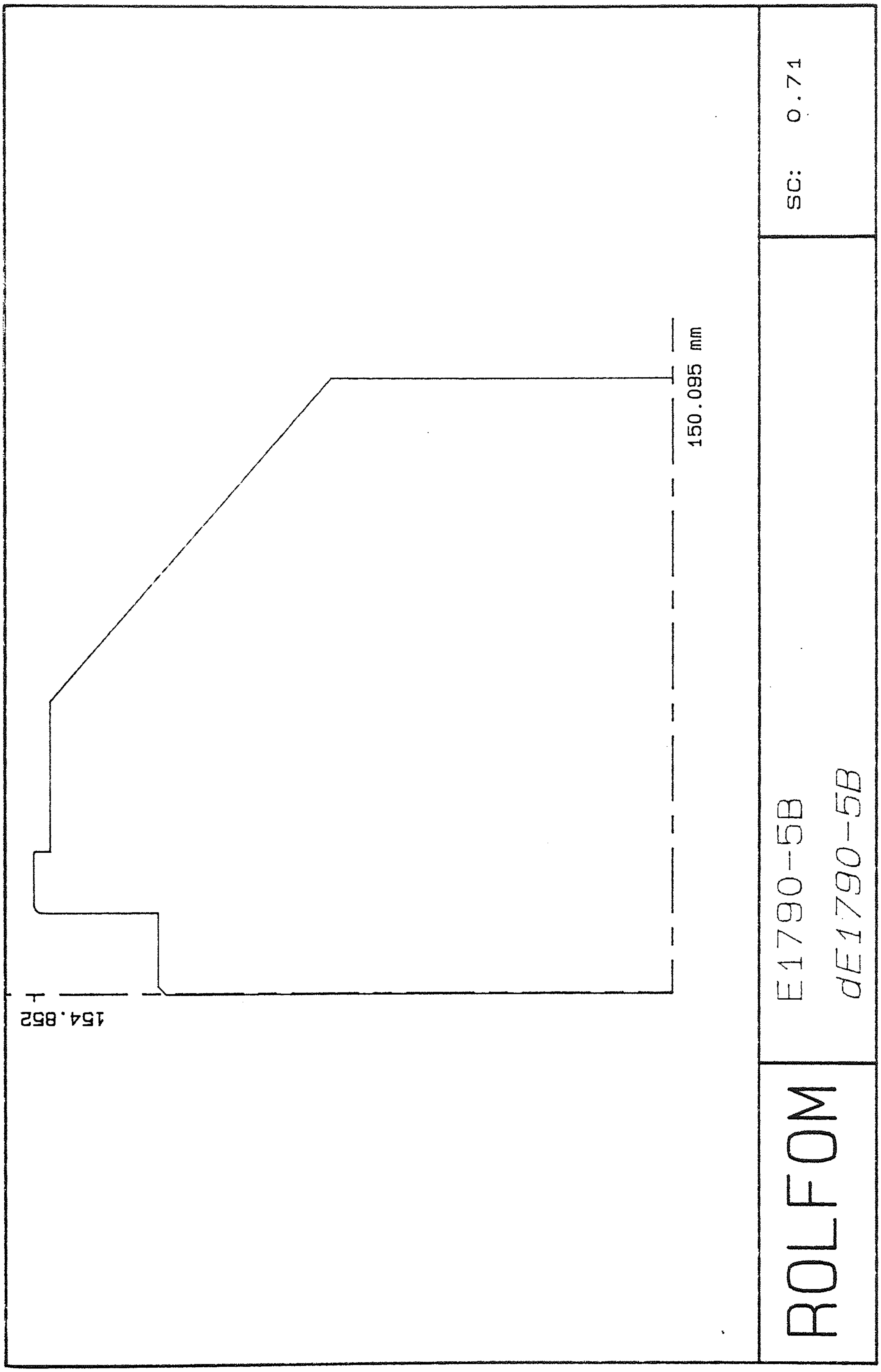
ROLFOM
E1790-4B
dE1790-4B
SC: 0.65
153.610 mm
188.489

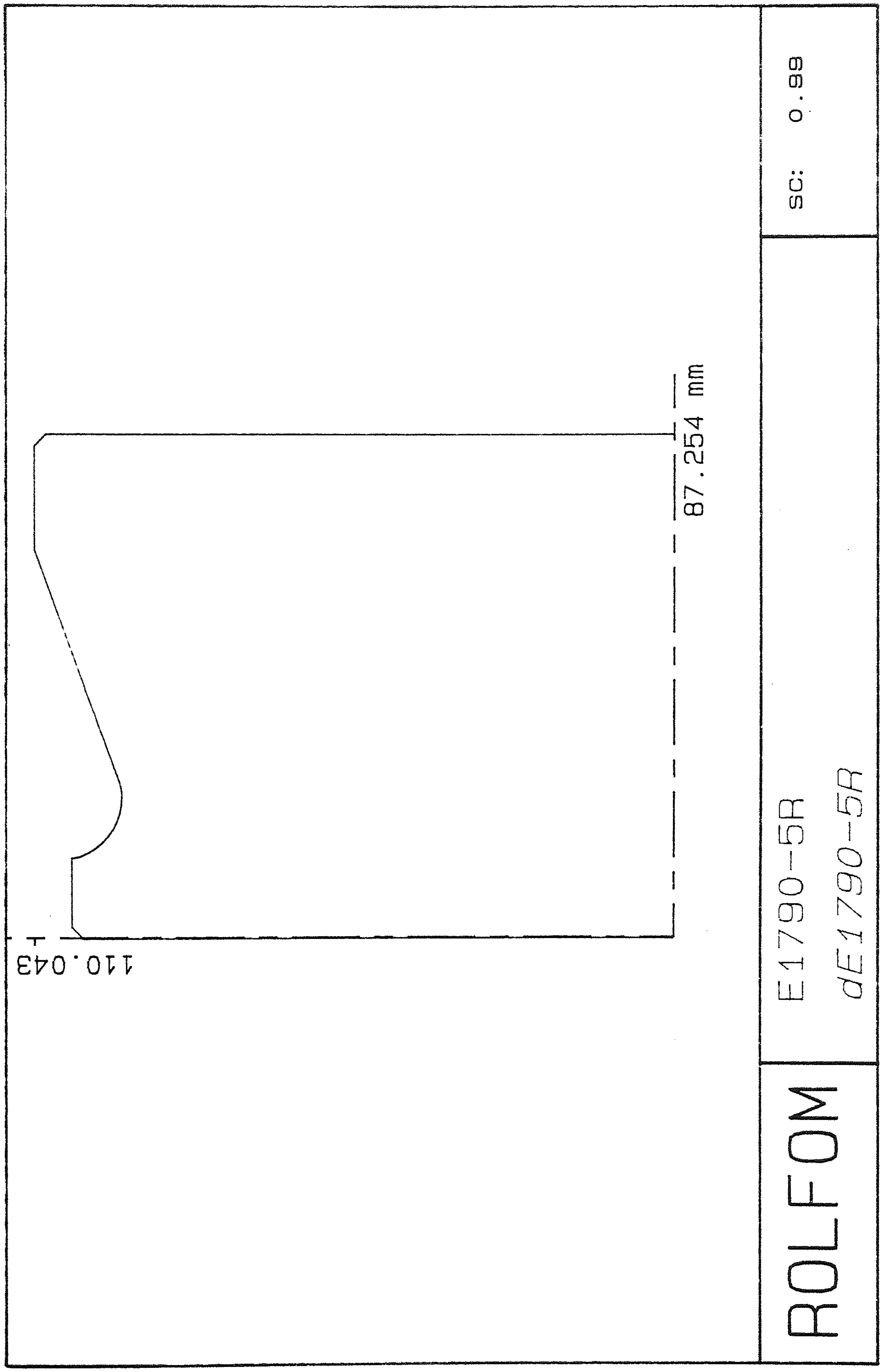

81.159 mm
103.639 mm
ROLFOM
E1790-4R
dE1790-4R
SC: 1.00



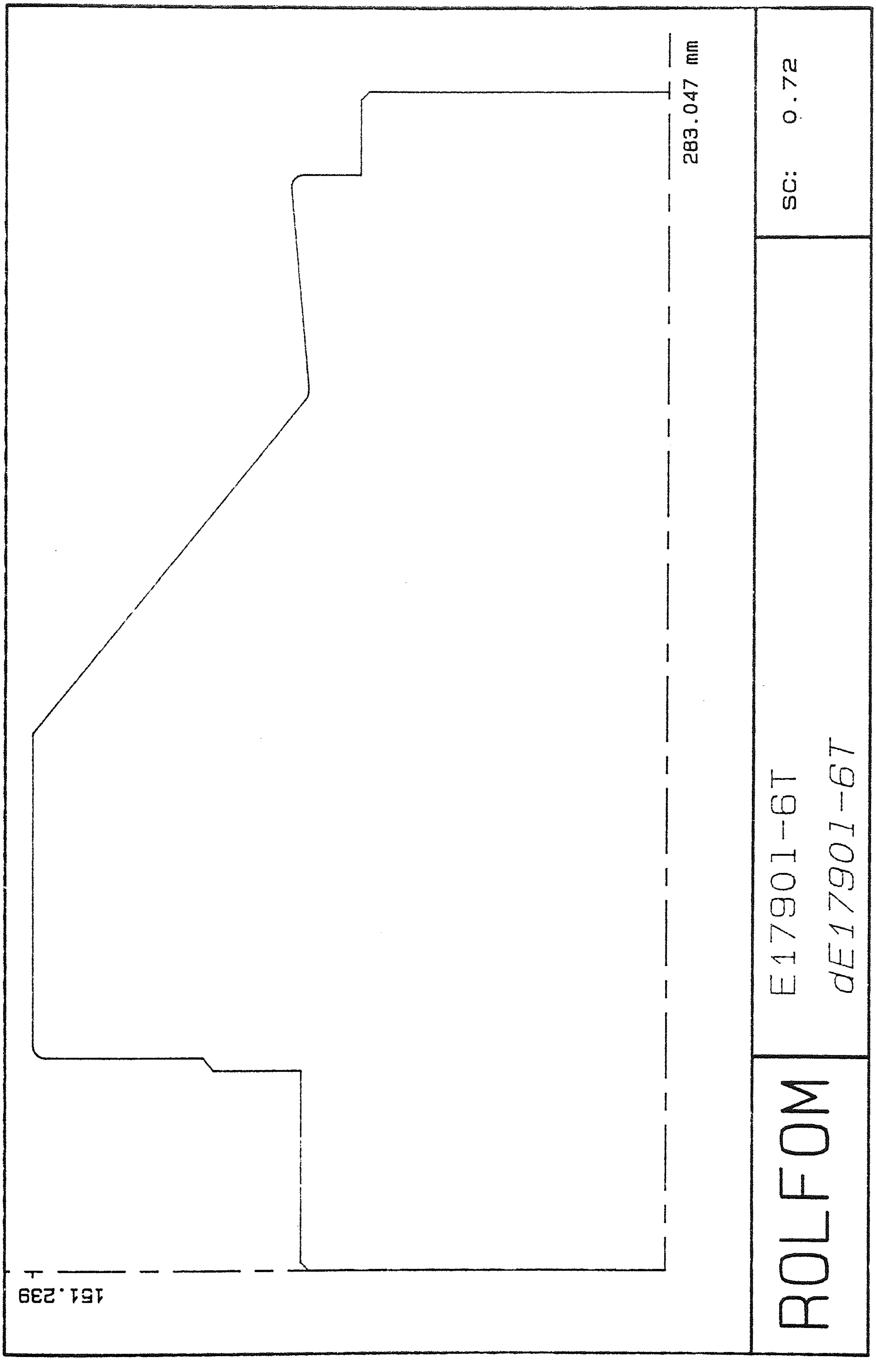
ROLFOM	E1790-5T *dE1790-5T*	SC: 0.71

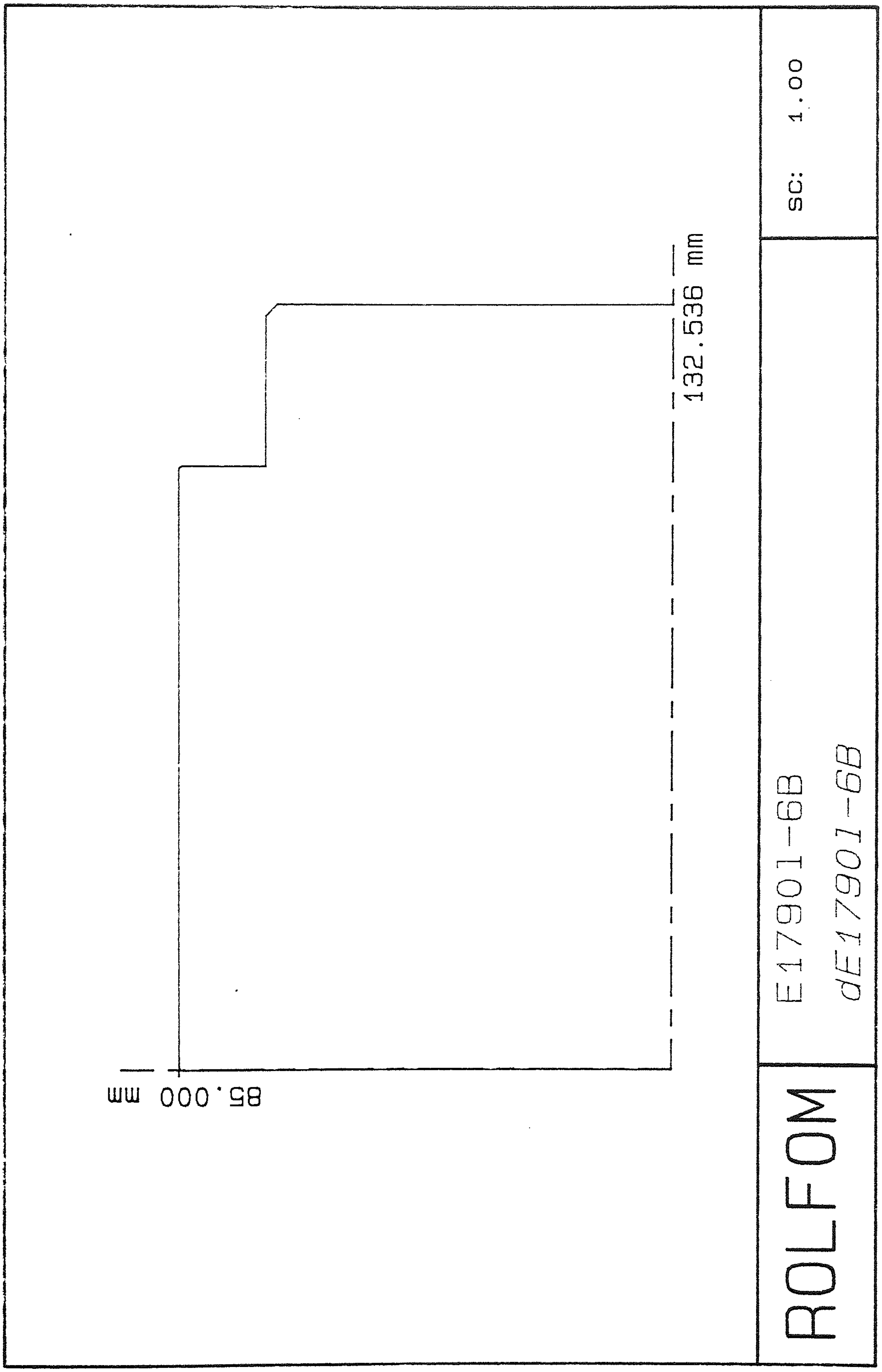

130.569 mm
82.500 mm
sc: 1.00
E17901-5B
dE17901-5B
ROLFOM

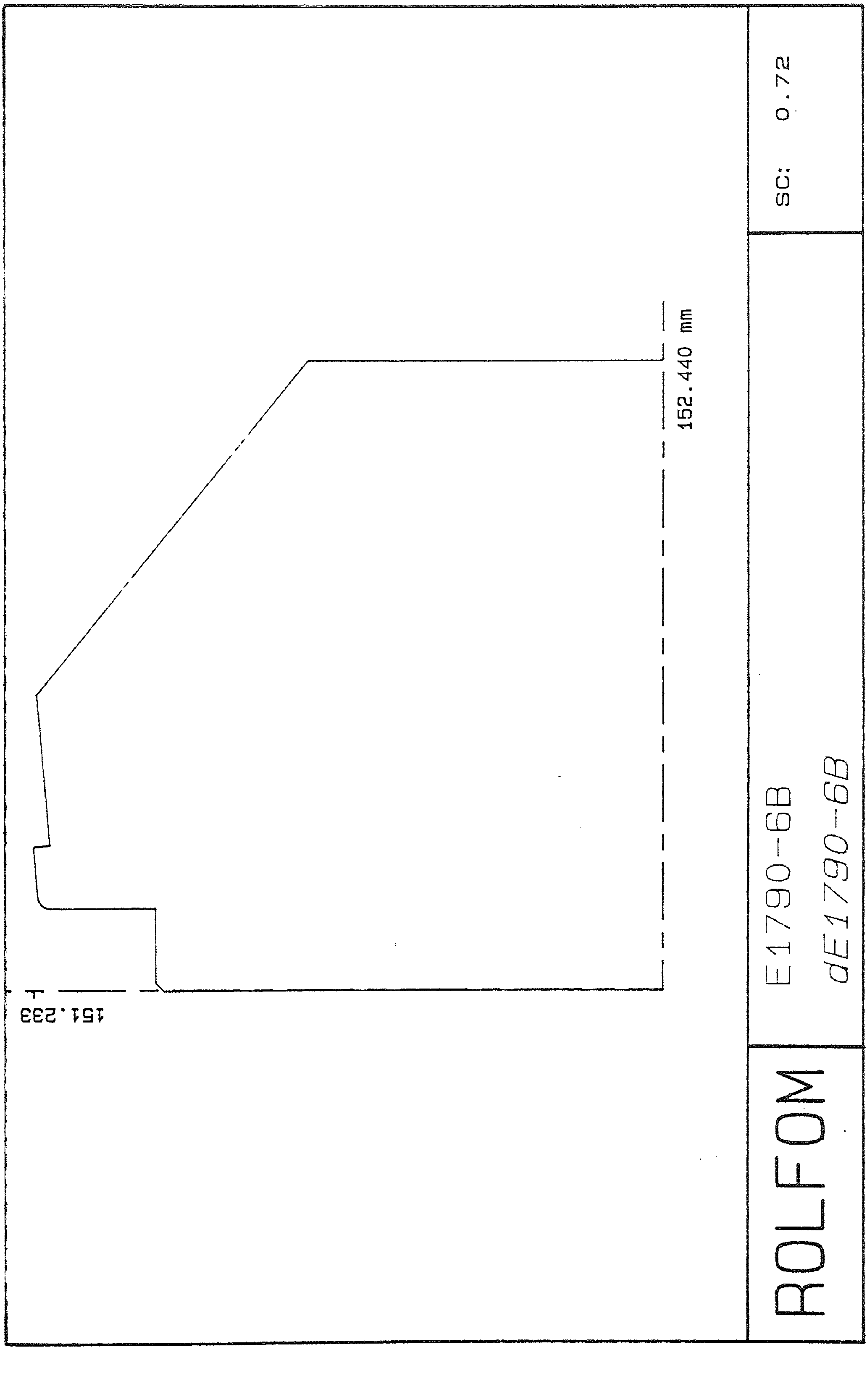

ROLFOM
E1790-5B
dE1790-5B
SC: 0.71
150.095 mm
154.852

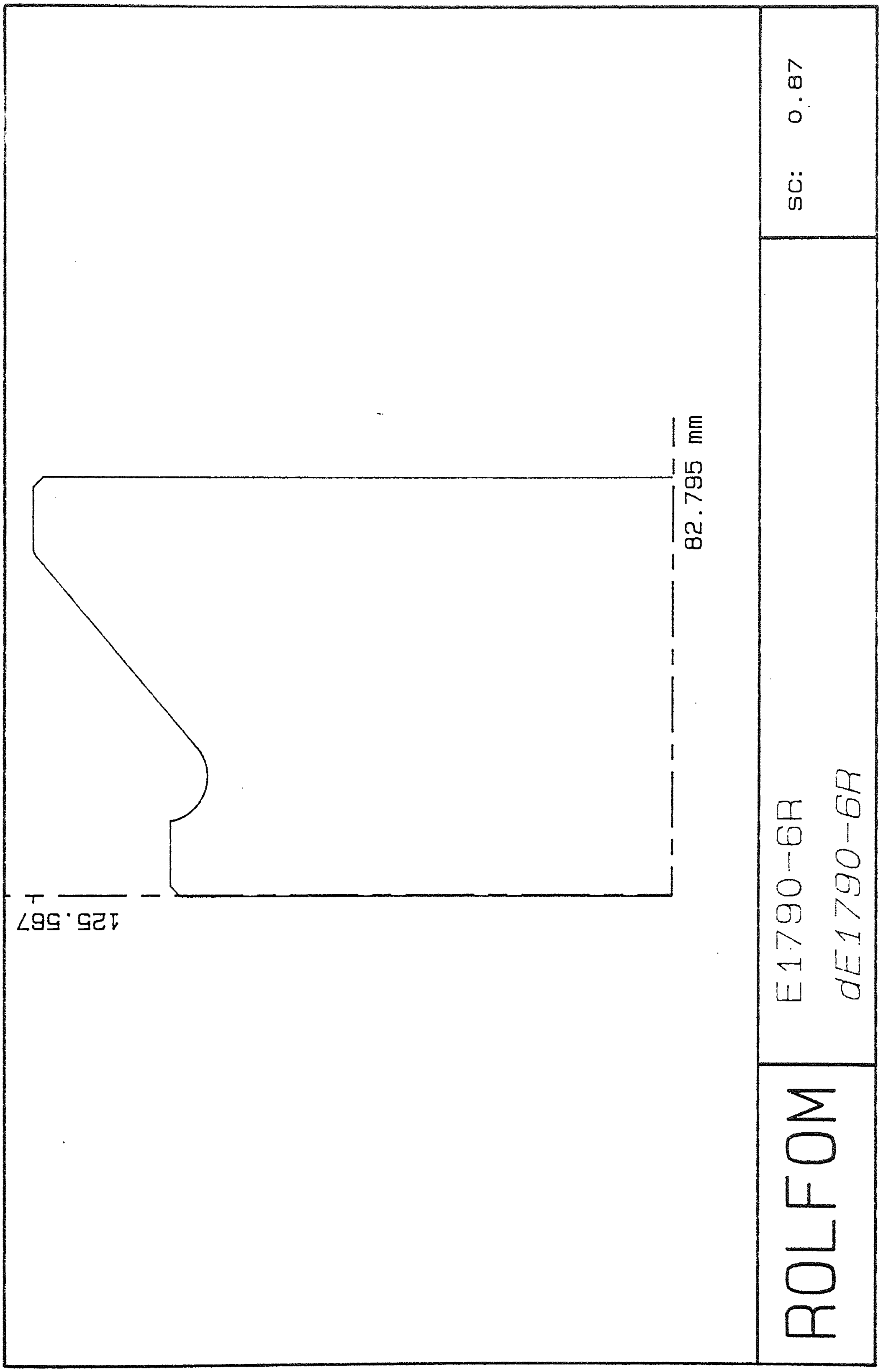


ROLFOM
E1790-5R
dE1790-5R
SC: 0.99
87.254 mm
110.043

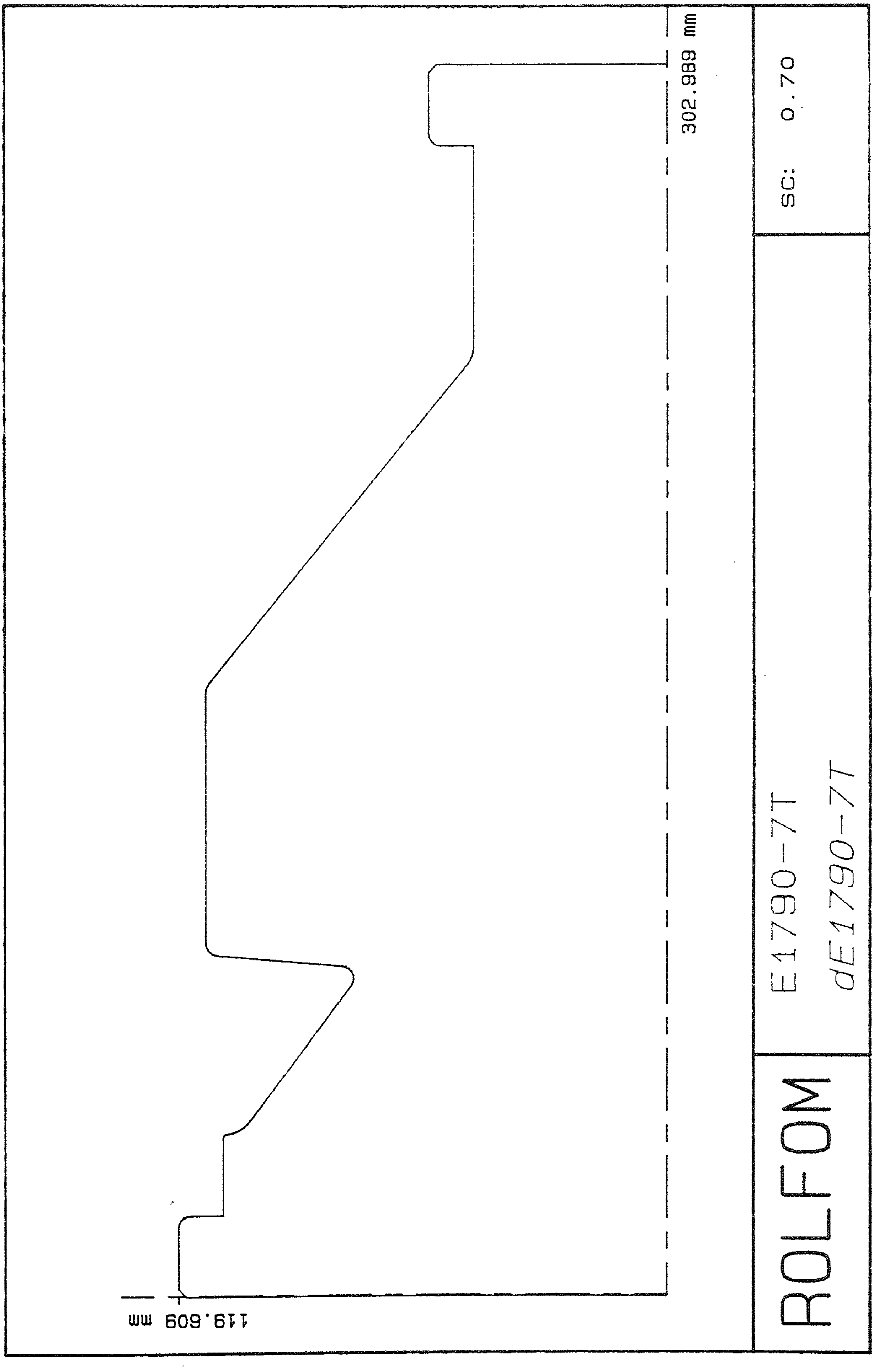

283.047 mm
SC: 0.72
E17901-6T
dE17901-6T
ROLFOM
151.239

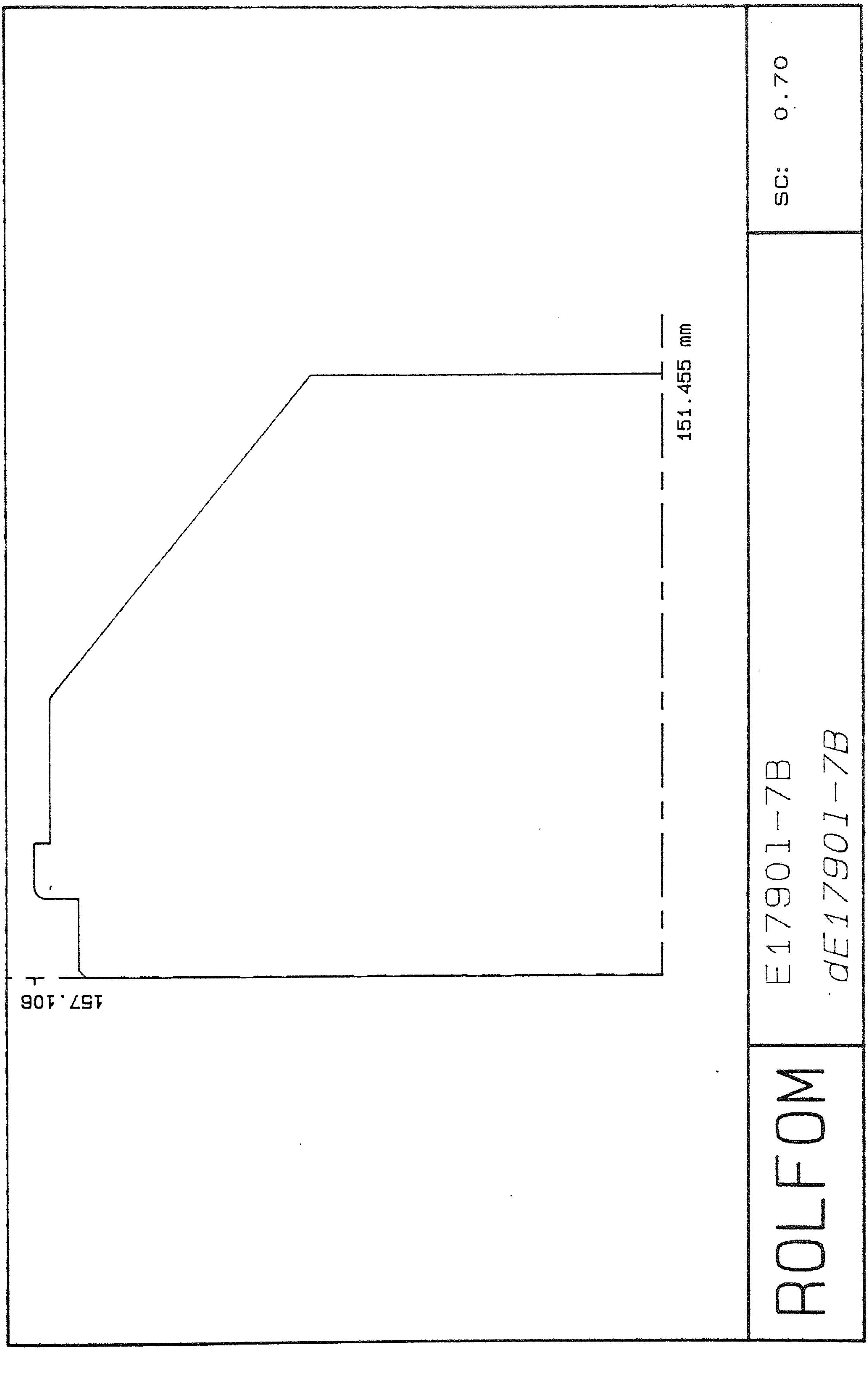

ROLFOM
E17901-6B
dE17901-6B
sc: 1.00
132.536 mm
85.000 mm

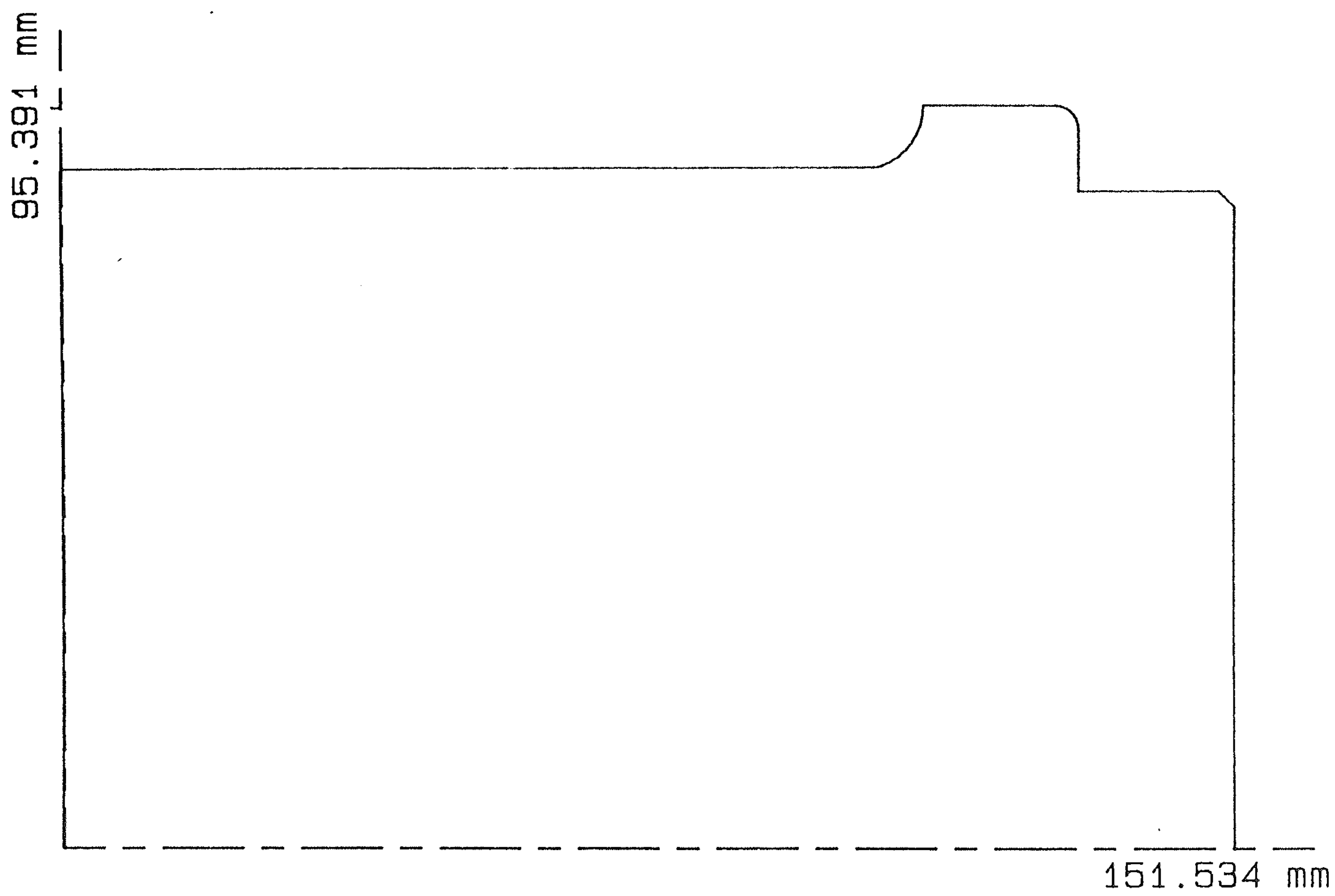

ROLFOM
E1790-6B
dE1790-6B
SC: 0.72
152.440 mm
151.233



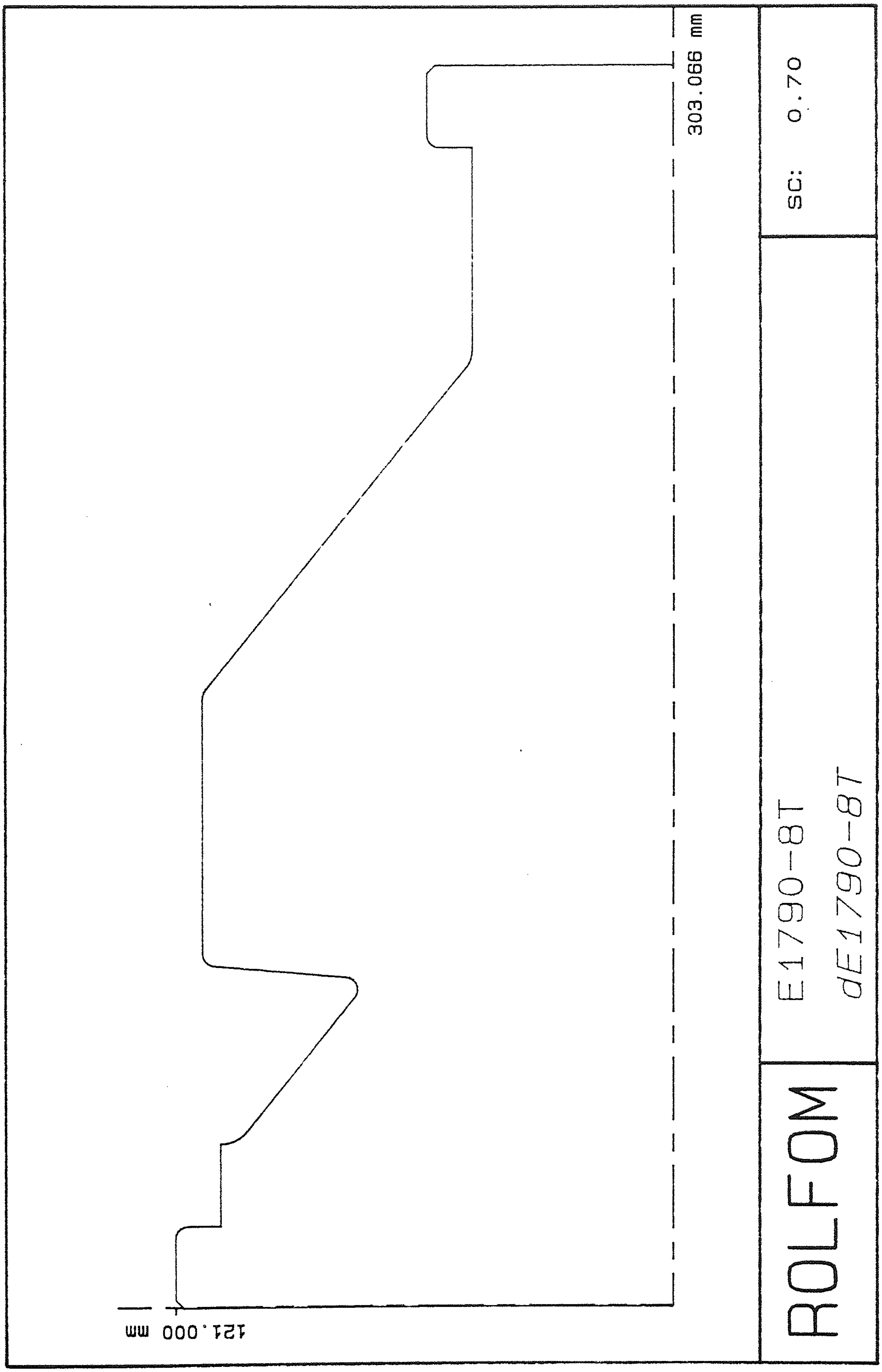
ROLFOM
E1790-6R
dE1790-6R
SC: 0.87
82.795 mm
125.567

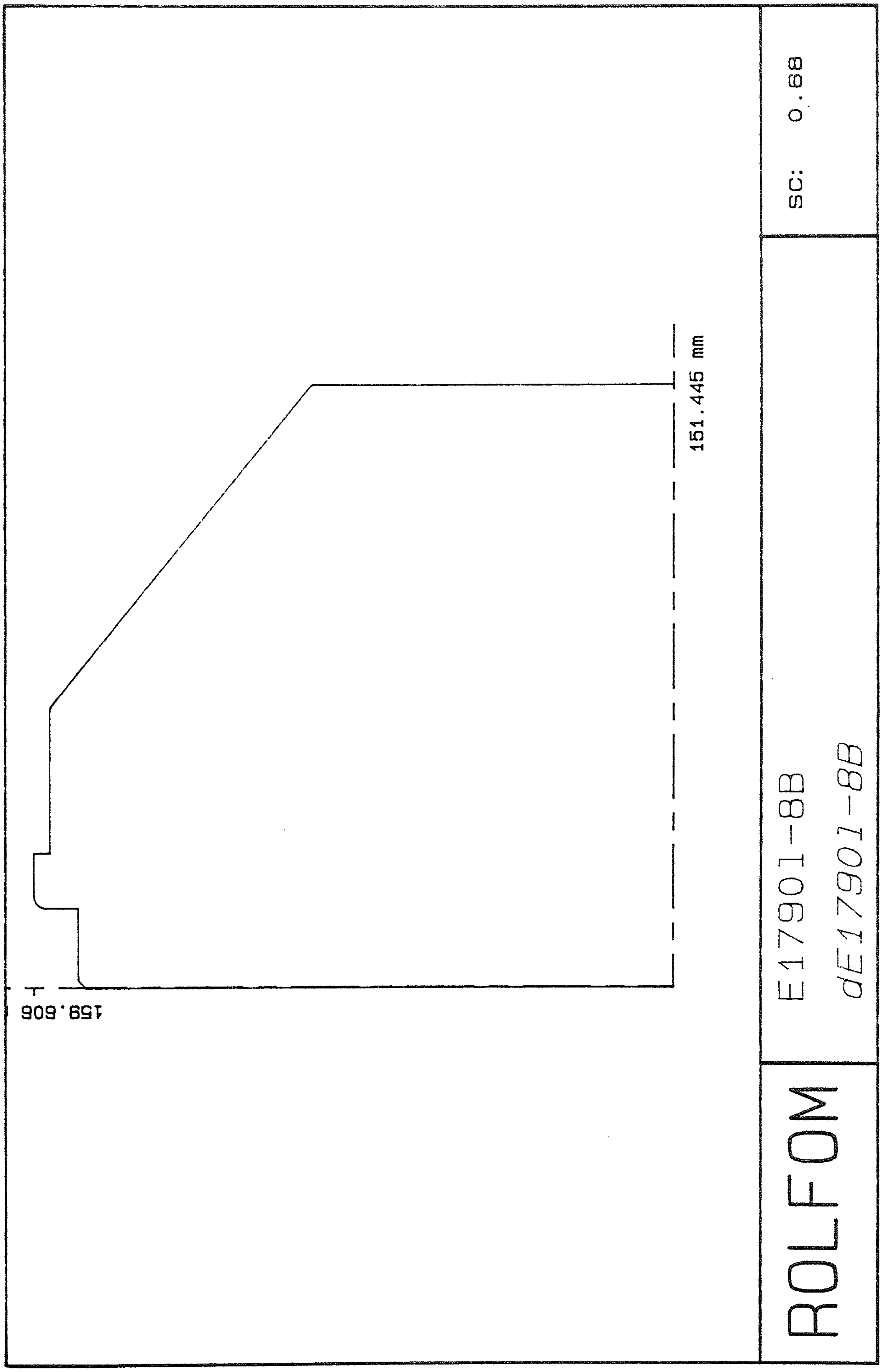

302.989 mm
119.609 mm
SC: 0.70
E1790-7T
dE1790-7T
ROLFOM

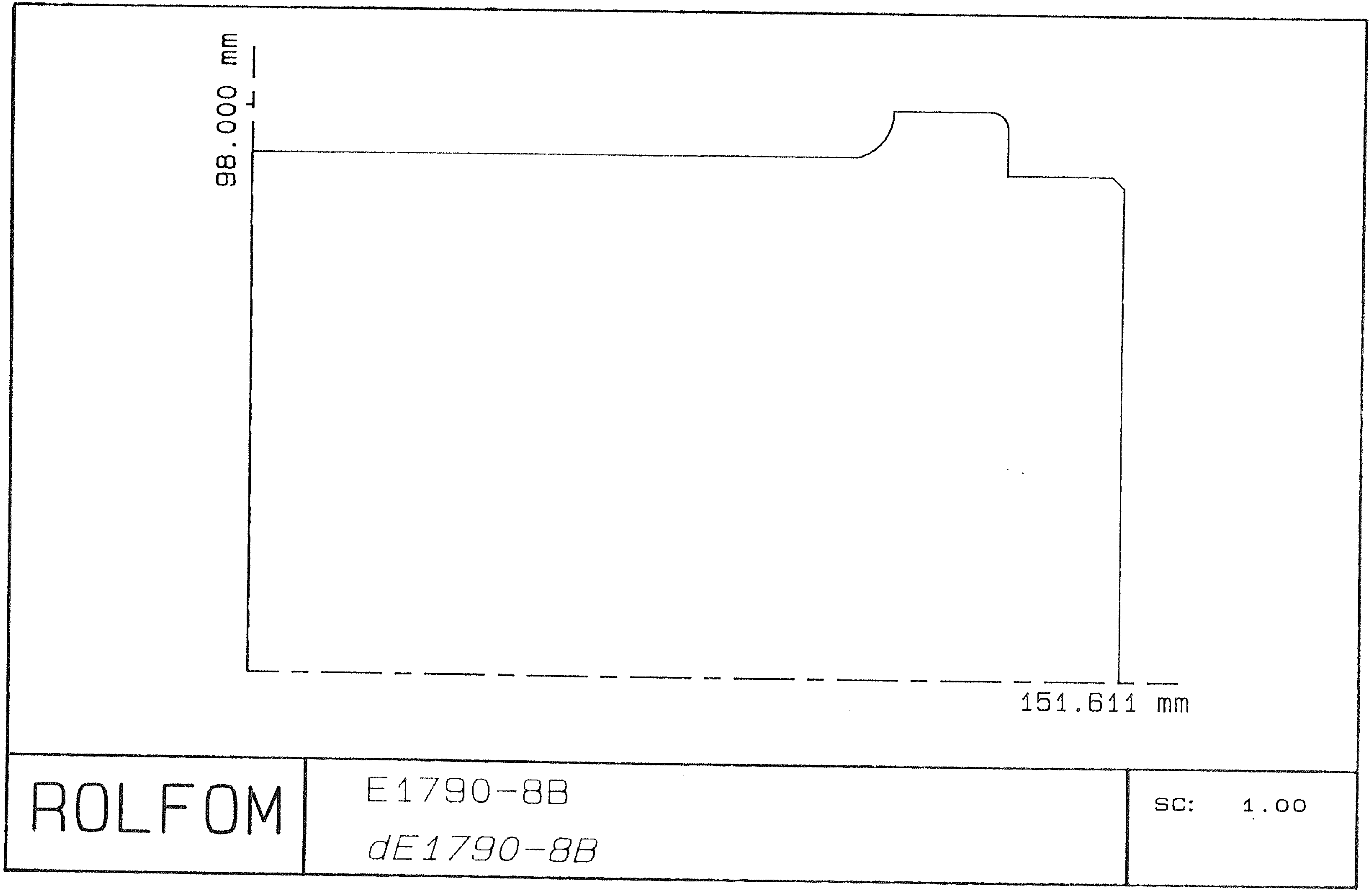

151.455 mm
157.106
ROLFOM
E17901-7B
dE17901-7B
SC: 0.70

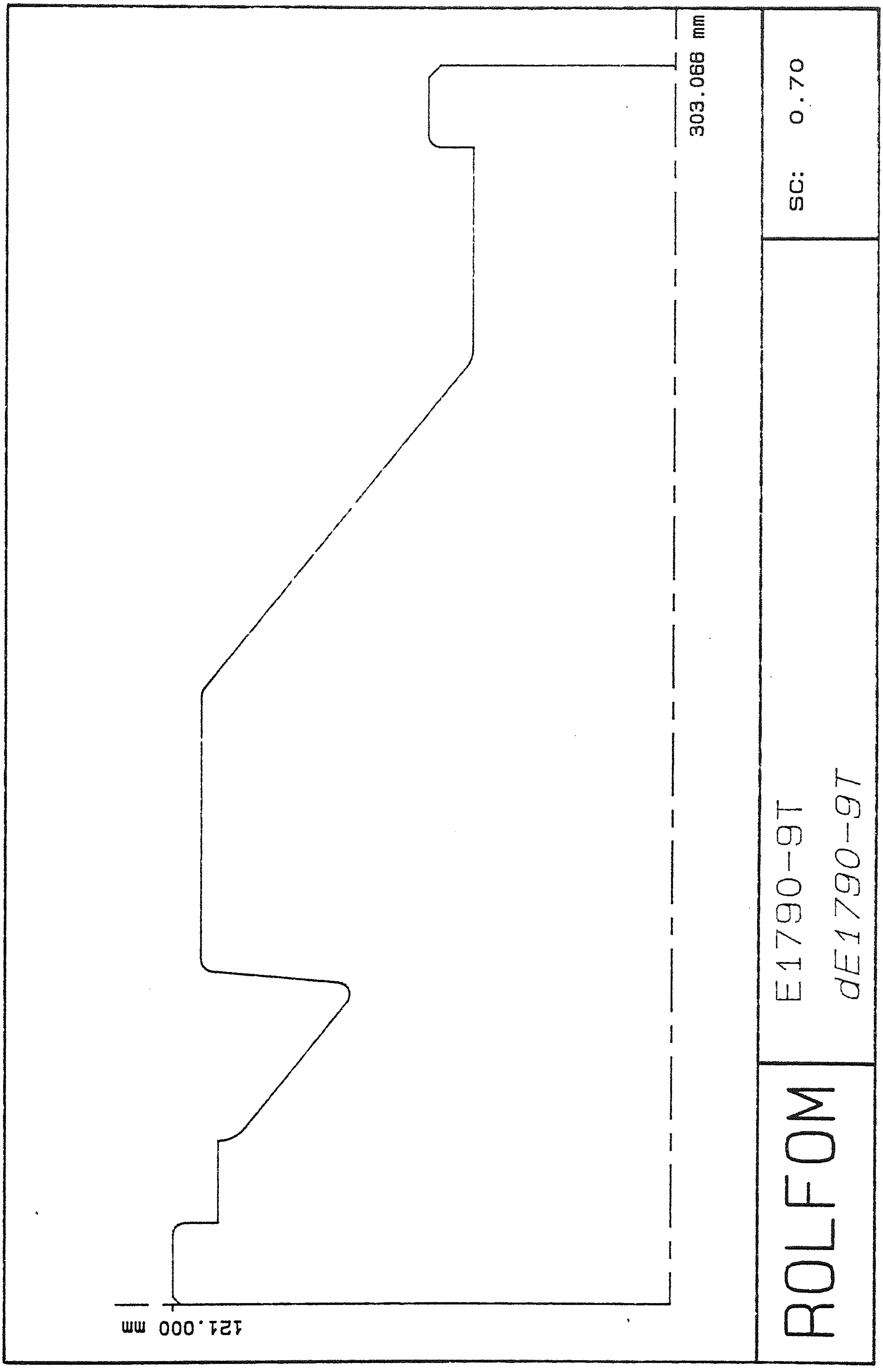


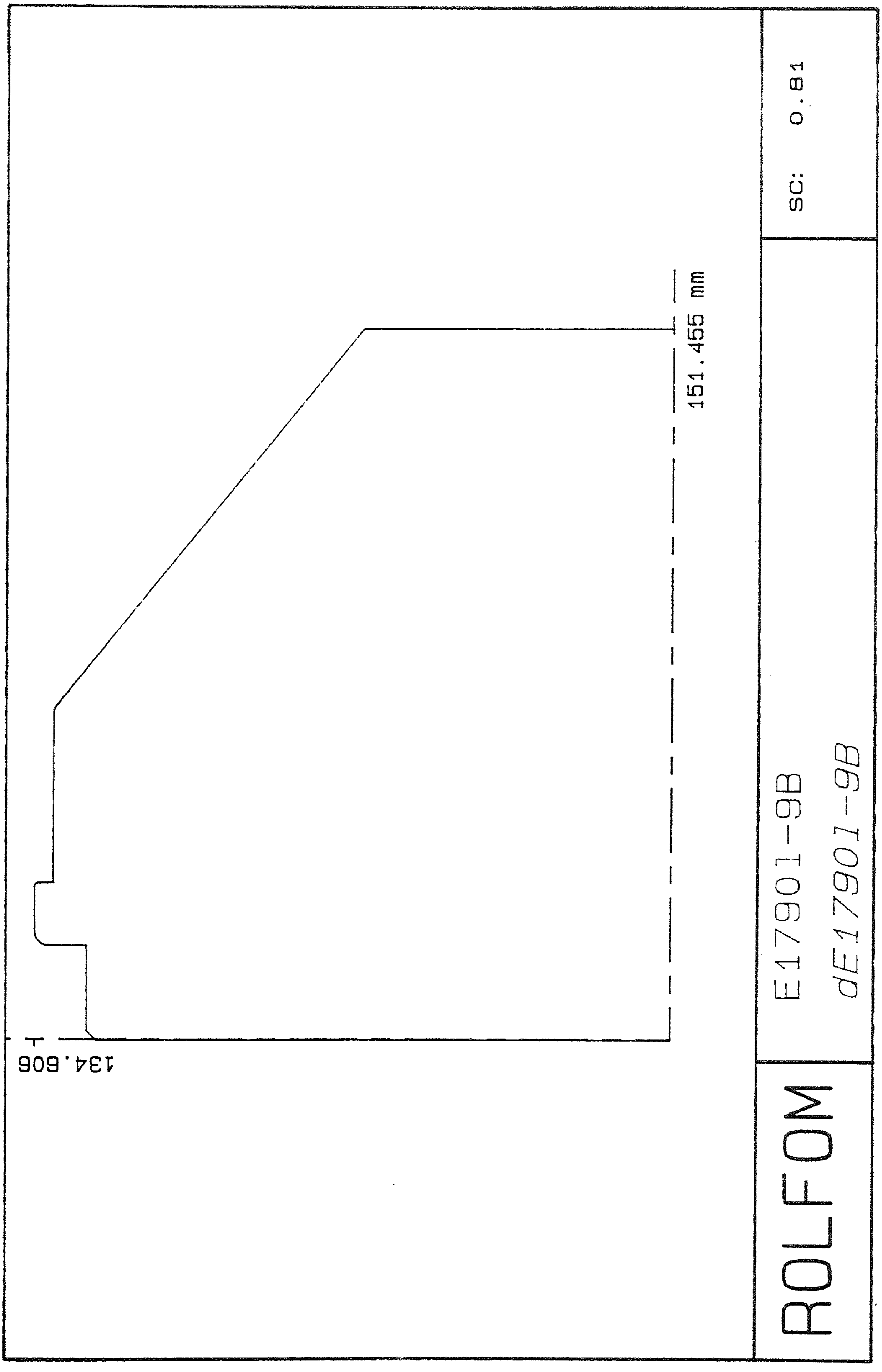
ROLFOM	E1790-7B *dE1790-7B*	SC: 1.00

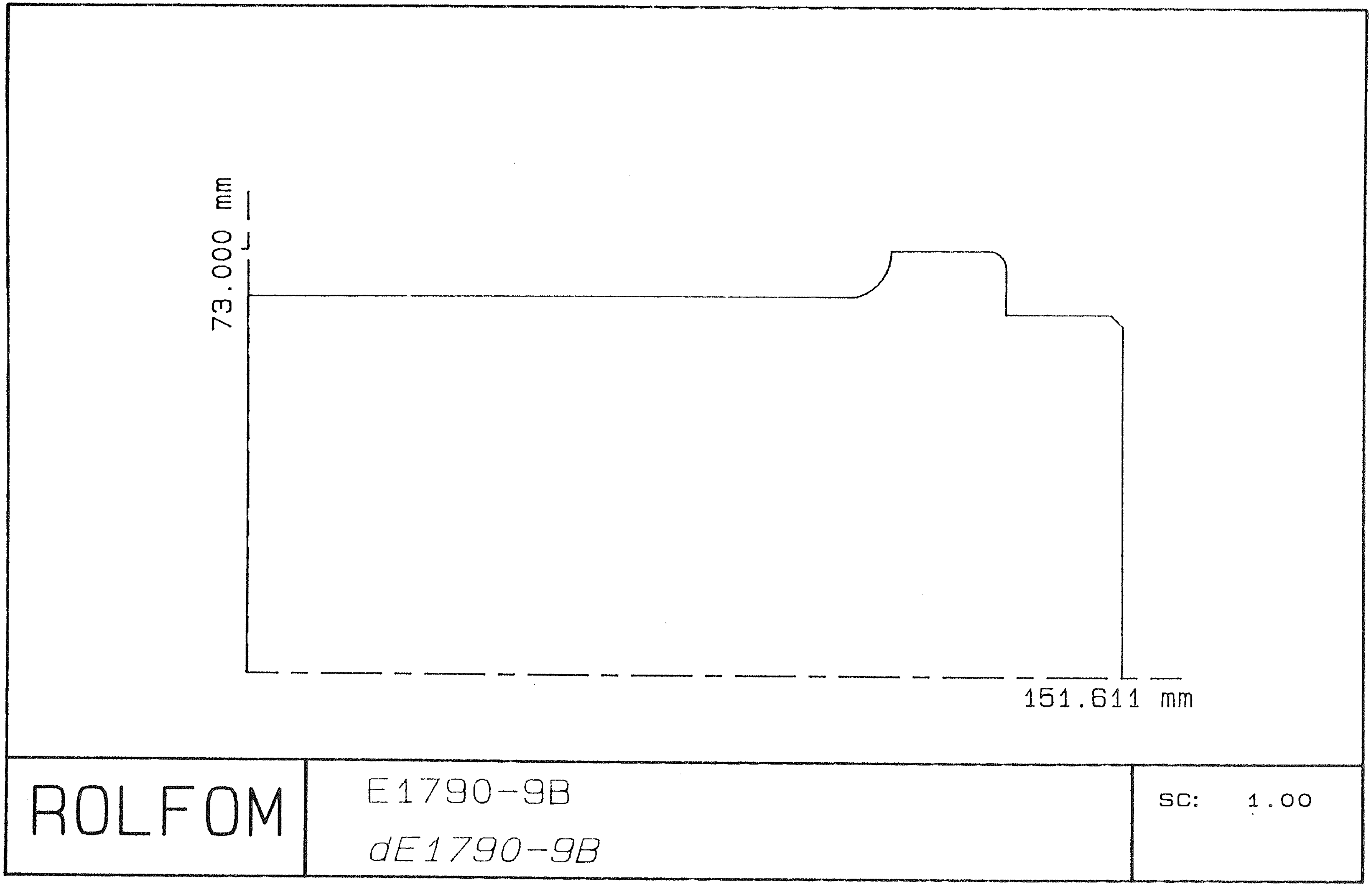

303.066 mm
121.000 mm
SC: 0.70
ROLFOM
E1790-8T
dE1790-8T


ROLFOM
E17901-8B
dE17901-8B
SC: 0.68
151.445 mm
159.606


98.000 mm
151.611 mm
ROLFOM
E1790-8B
dE1790-8B
SC: 1.00

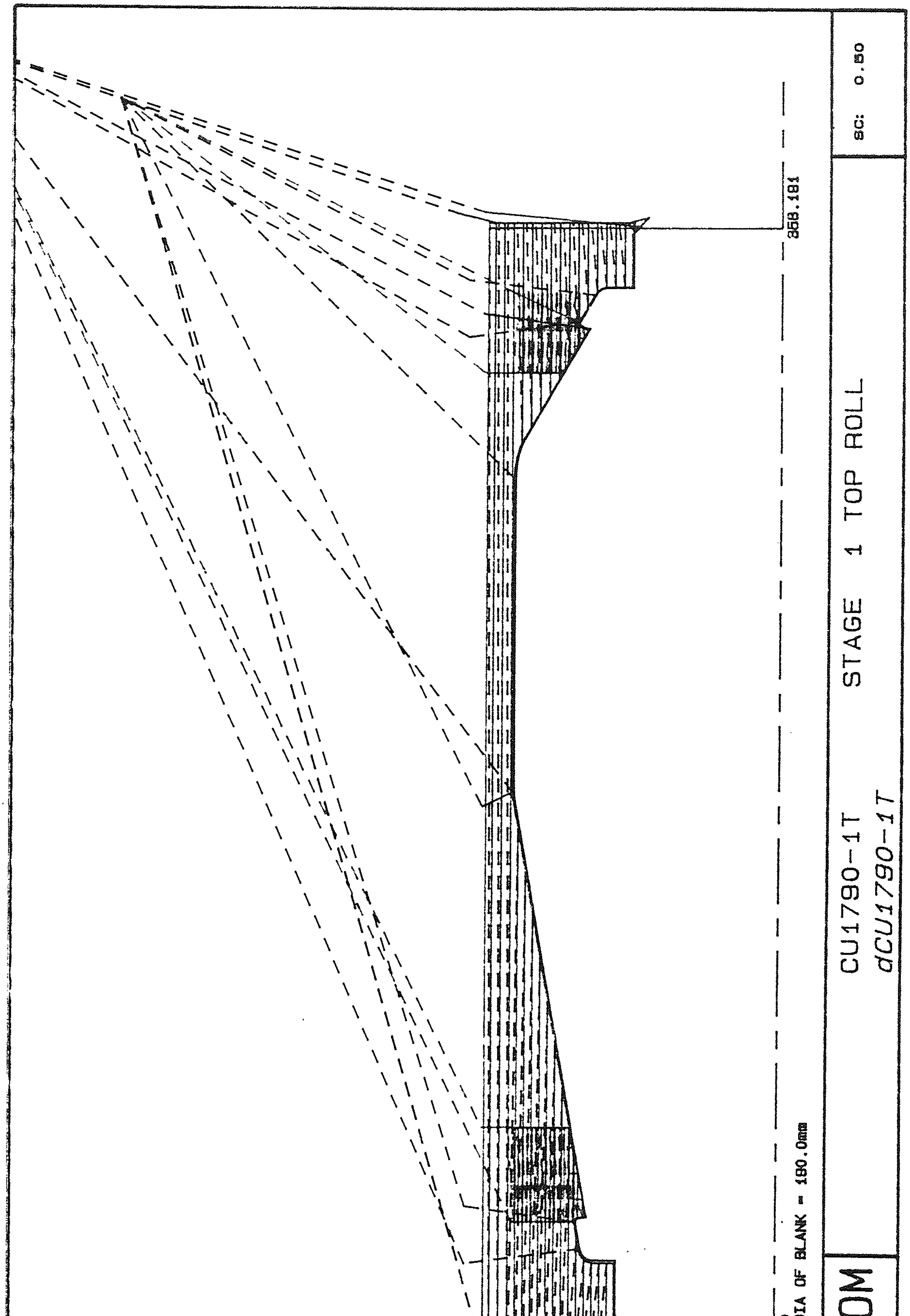

303.068 mm
121.000 mm
SC: 0.70
E1790-9T
dE1790-9T
ROLFOM

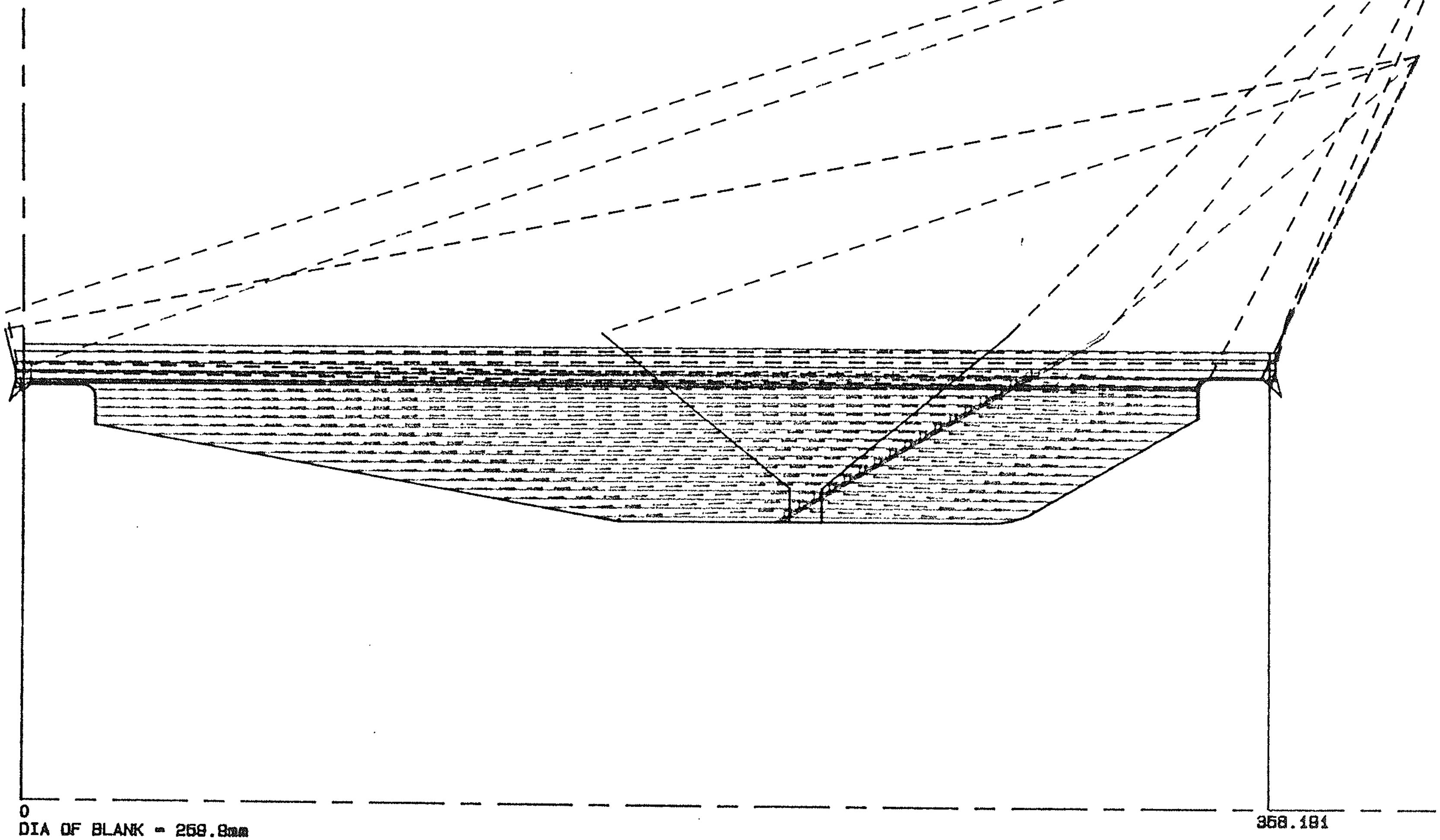

151.455 mm
134.606
ROLFOM
E17901-9B
dE17901-9B
SC: 0.81


73.000 mm
151.611 mm
ROLFOM
E1790-9B
dE1790-9B
SC: 1.00

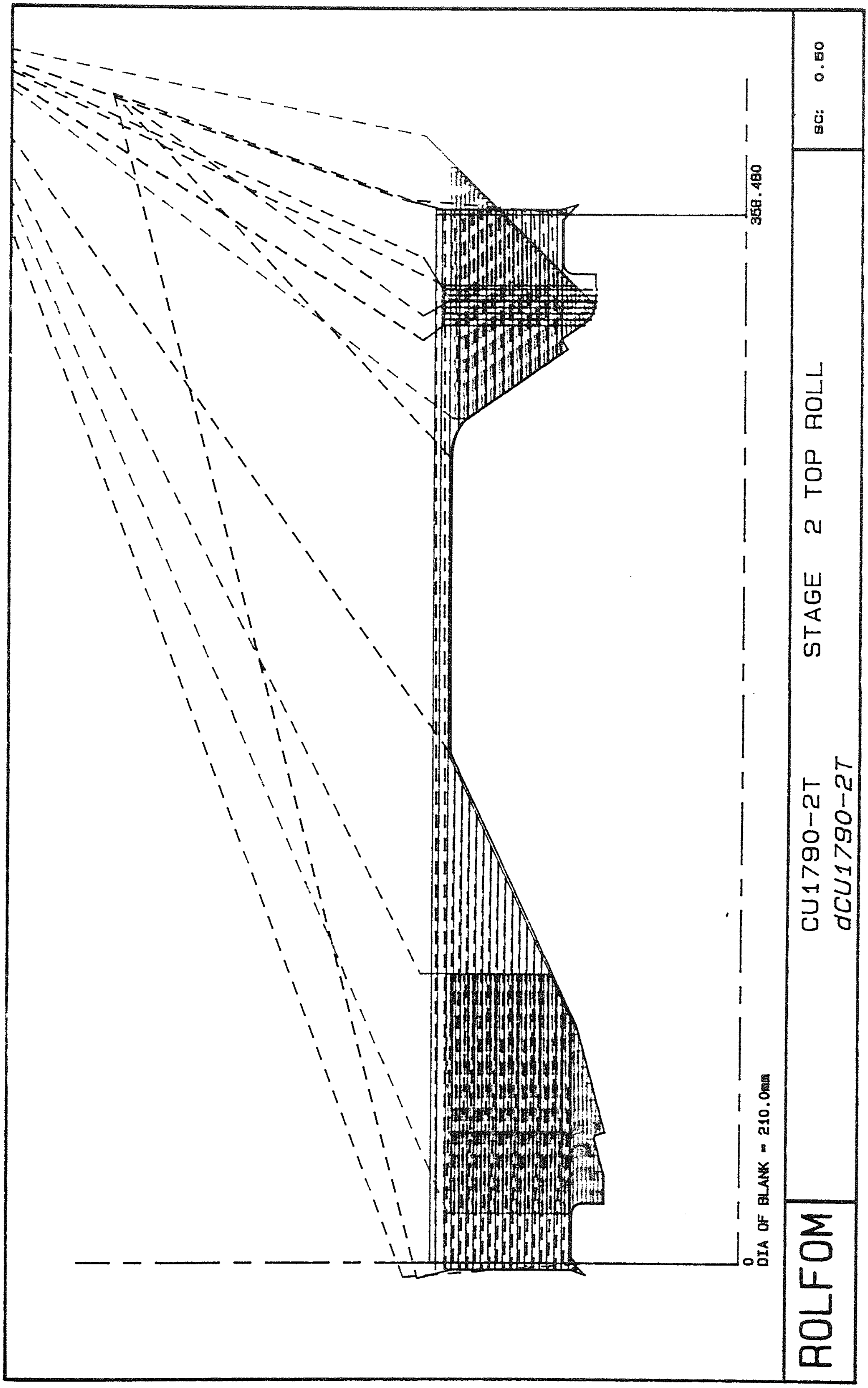
APPENDIX 7

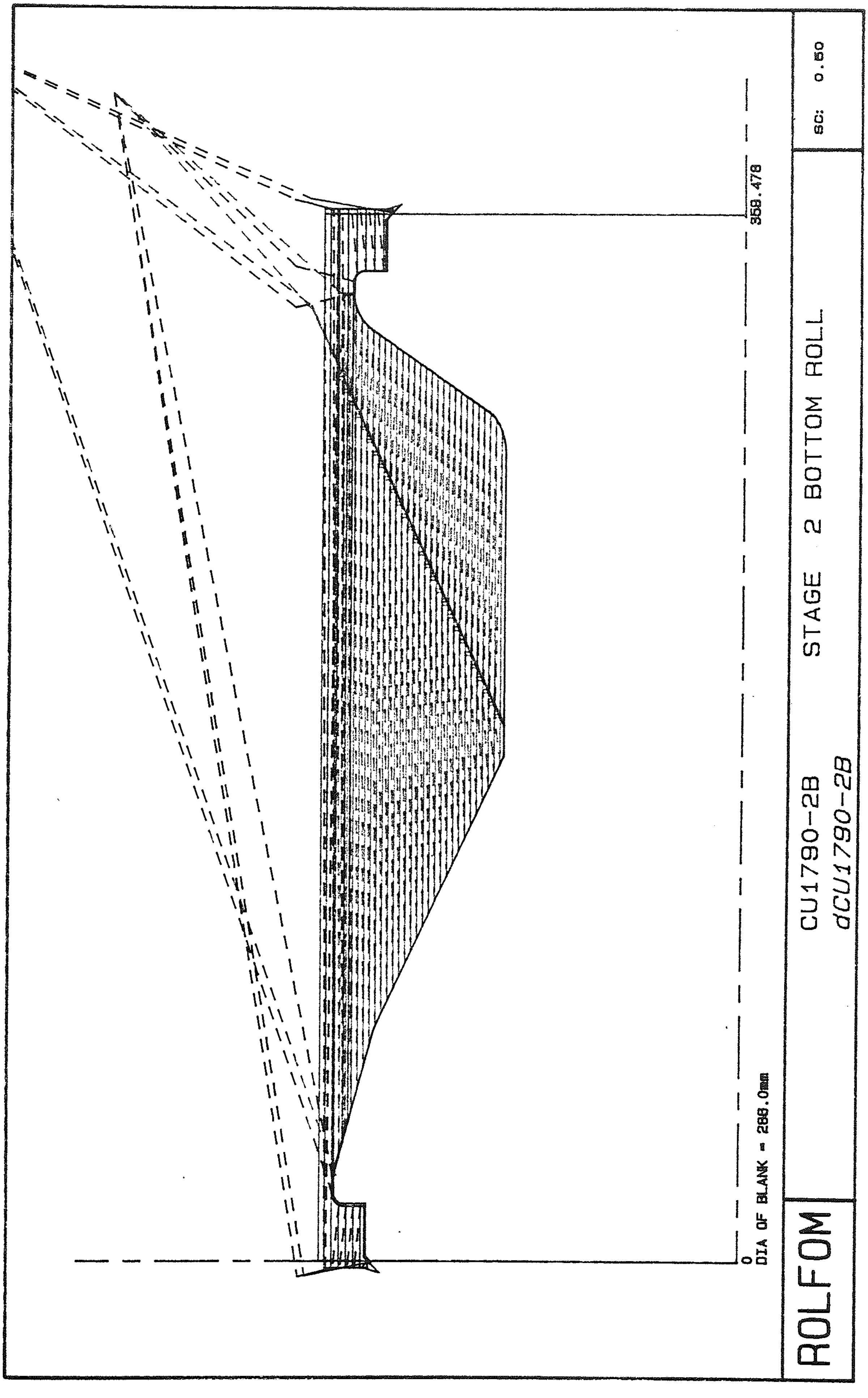
THE SET OF CUTTER PATH DRAWINGS FOR THE ROLLS SHOWN IN APPENDIX 6

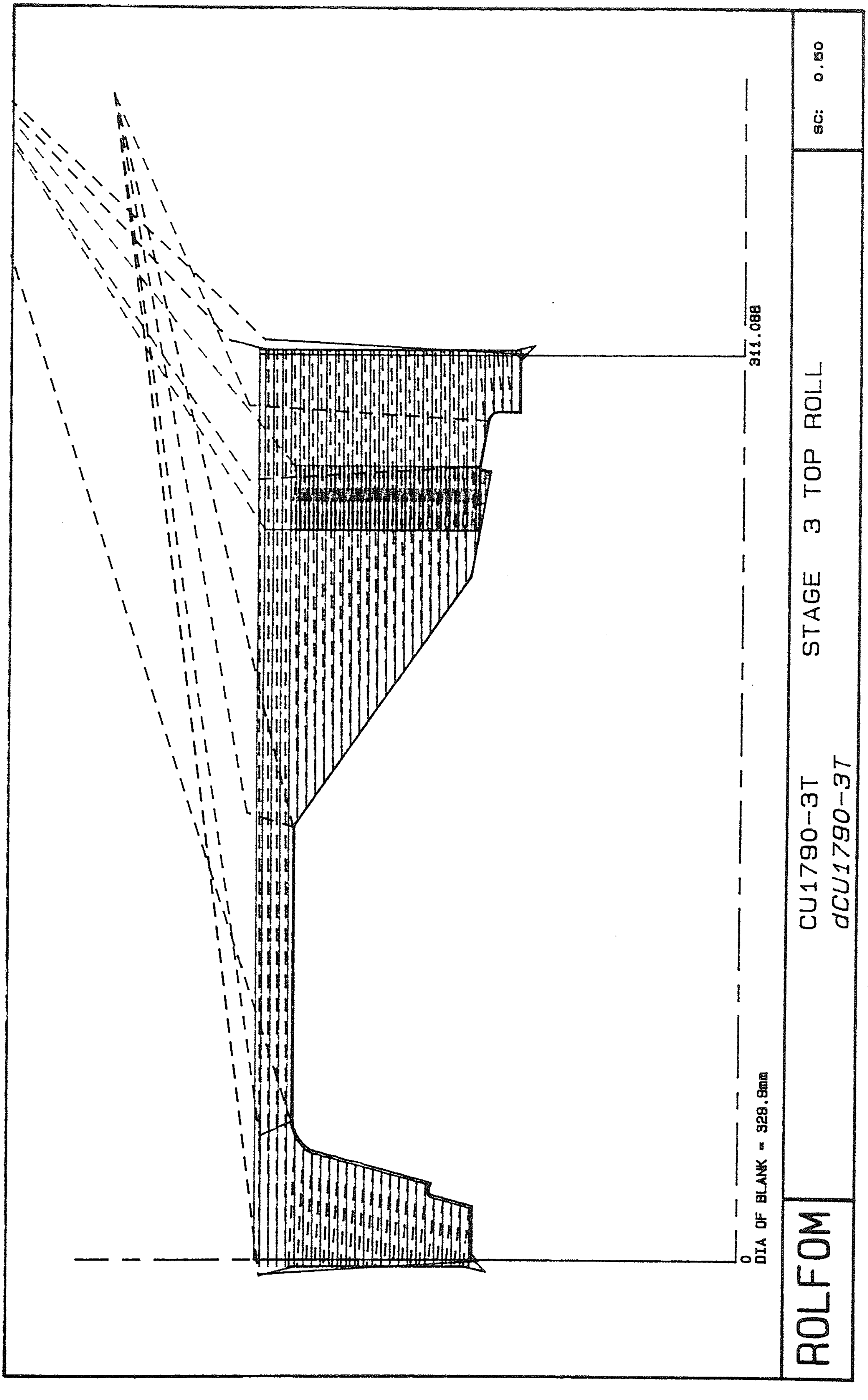

358.181
0
DIA OF BLANK = 180.0mm
ROLFOM
CU1790-1T
dCU1790-1T
STAGE 1 TOP ROLL
SC: 0.50



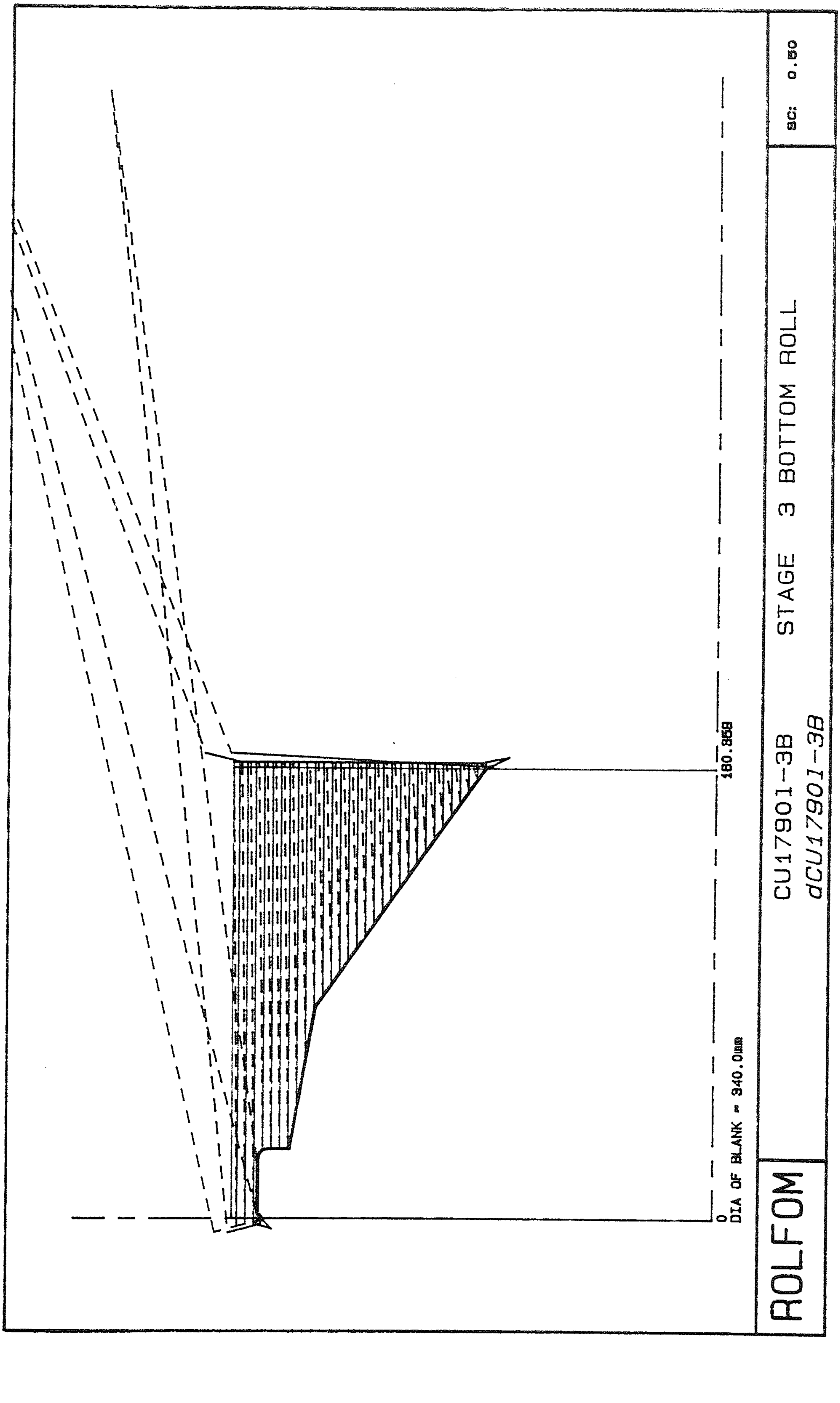
ROLFOM	CU1790-1B STAGE 1 BOTTOM ROLL *dCU1790-1B*	SC: 0.50

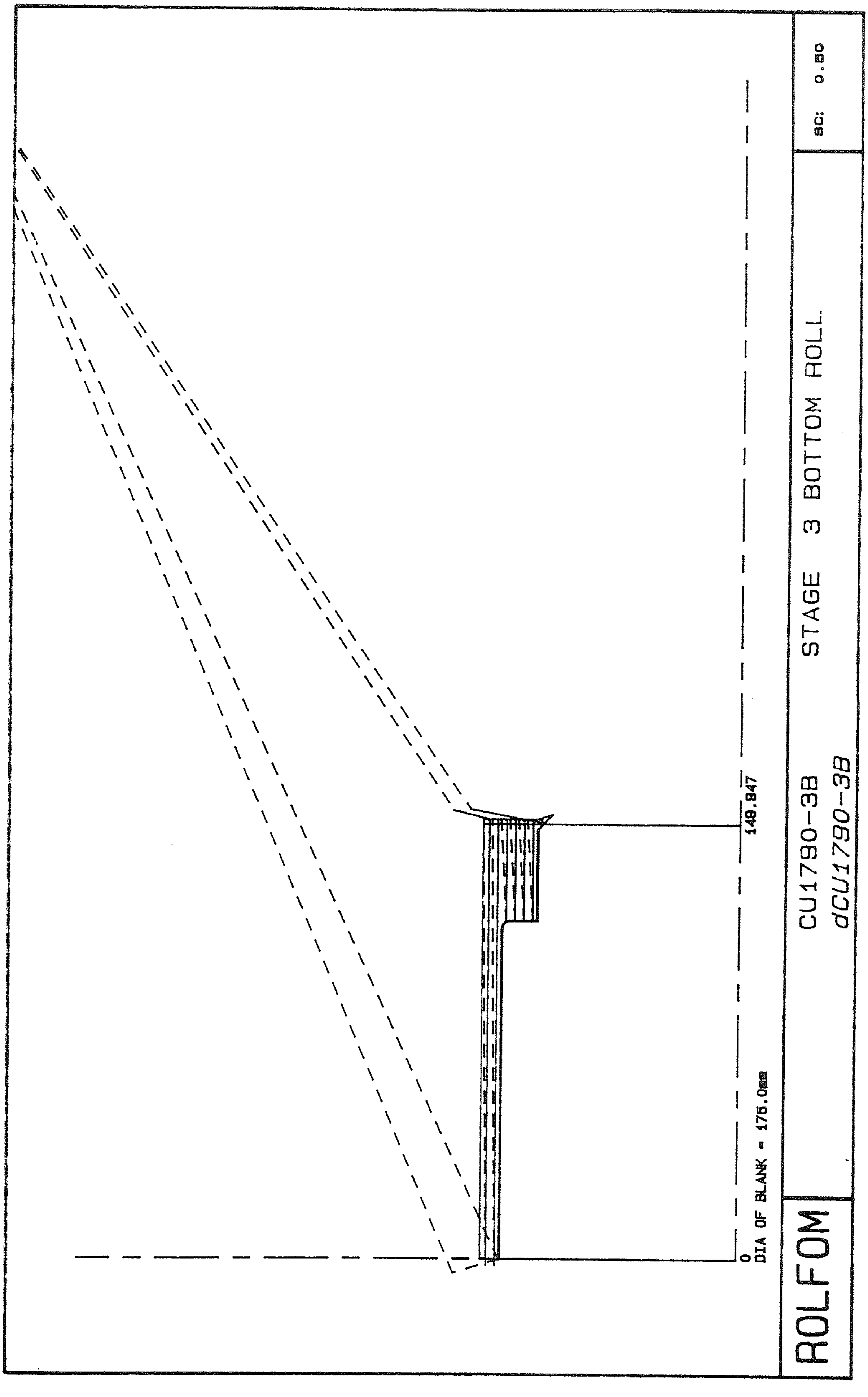

ROLFOM
CU1790-2T
dCU1790-2T
STAGE 2 TOP ROLL
SC: 0.50
358.480
0
DIA OF BLANK = 210.0mm

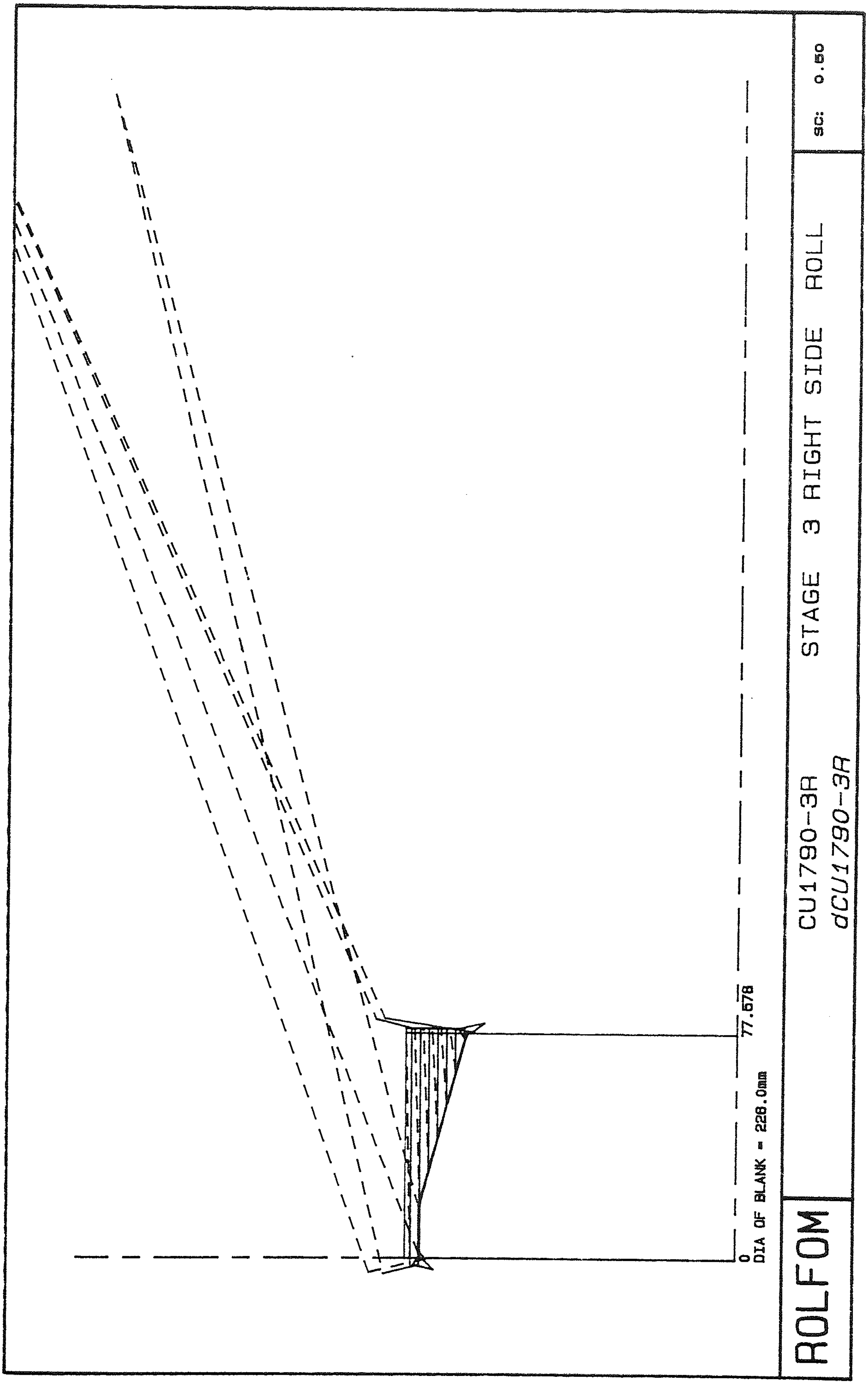

ROLFOM
CU1790-2B
dCU1790-2B
STAGE 2 BOTTOM ROLL
SC: 0.50
358.478
0
DIA OF BLANK = 266.0mm



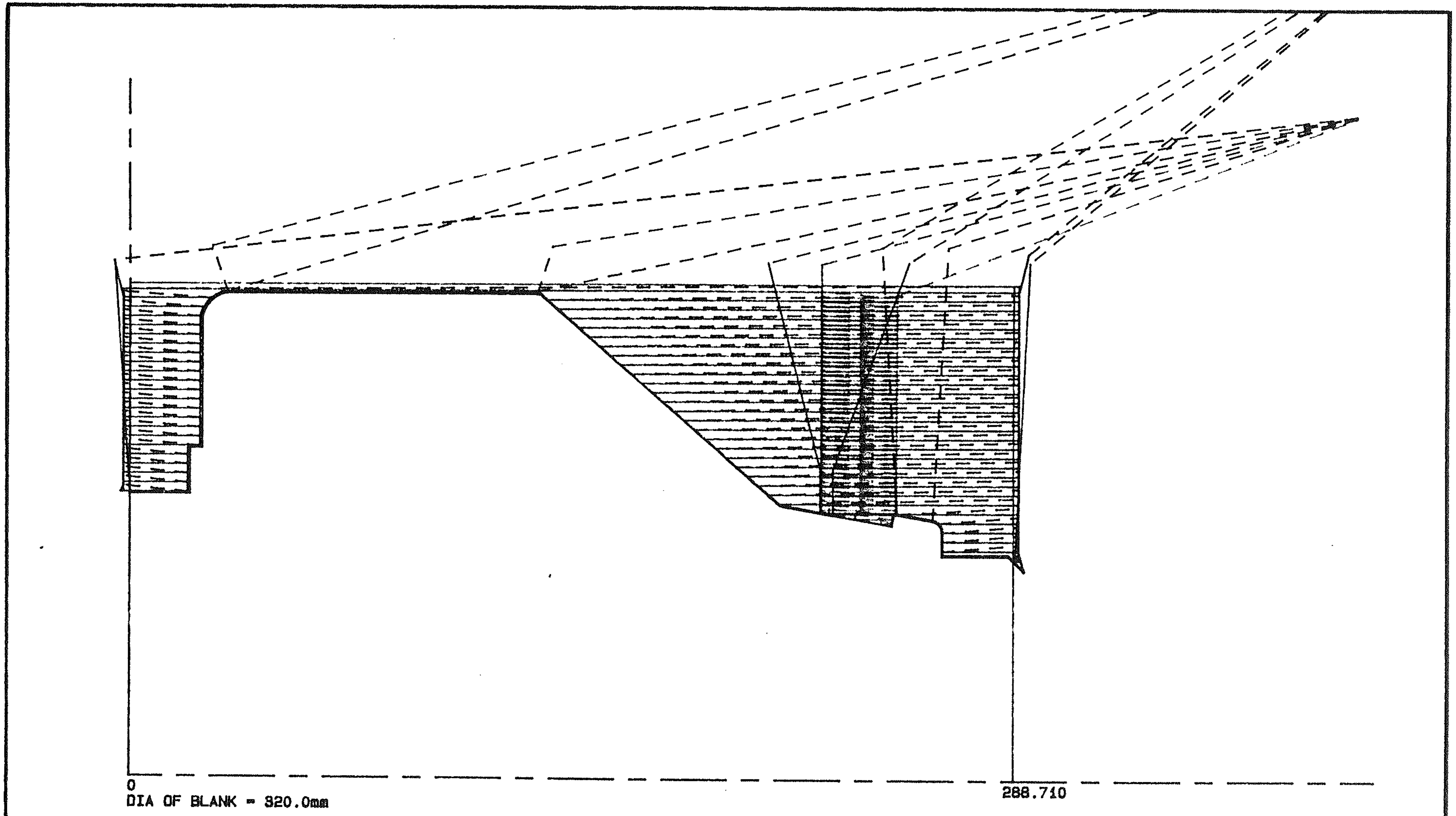
ROLFOM
CU1790-3T
dCU1790-3T
STAGE 3 TOP ROLL
SC: 0.50
0
DIA OF BLANK = 328.8mm
311.088


ROLFOM
CU17901-3B
dCU17901-3B
STAGE 3 BOTTOM ROLL
SC: 0.50
0
DIA OF BLANK = 340.0mm
180.858

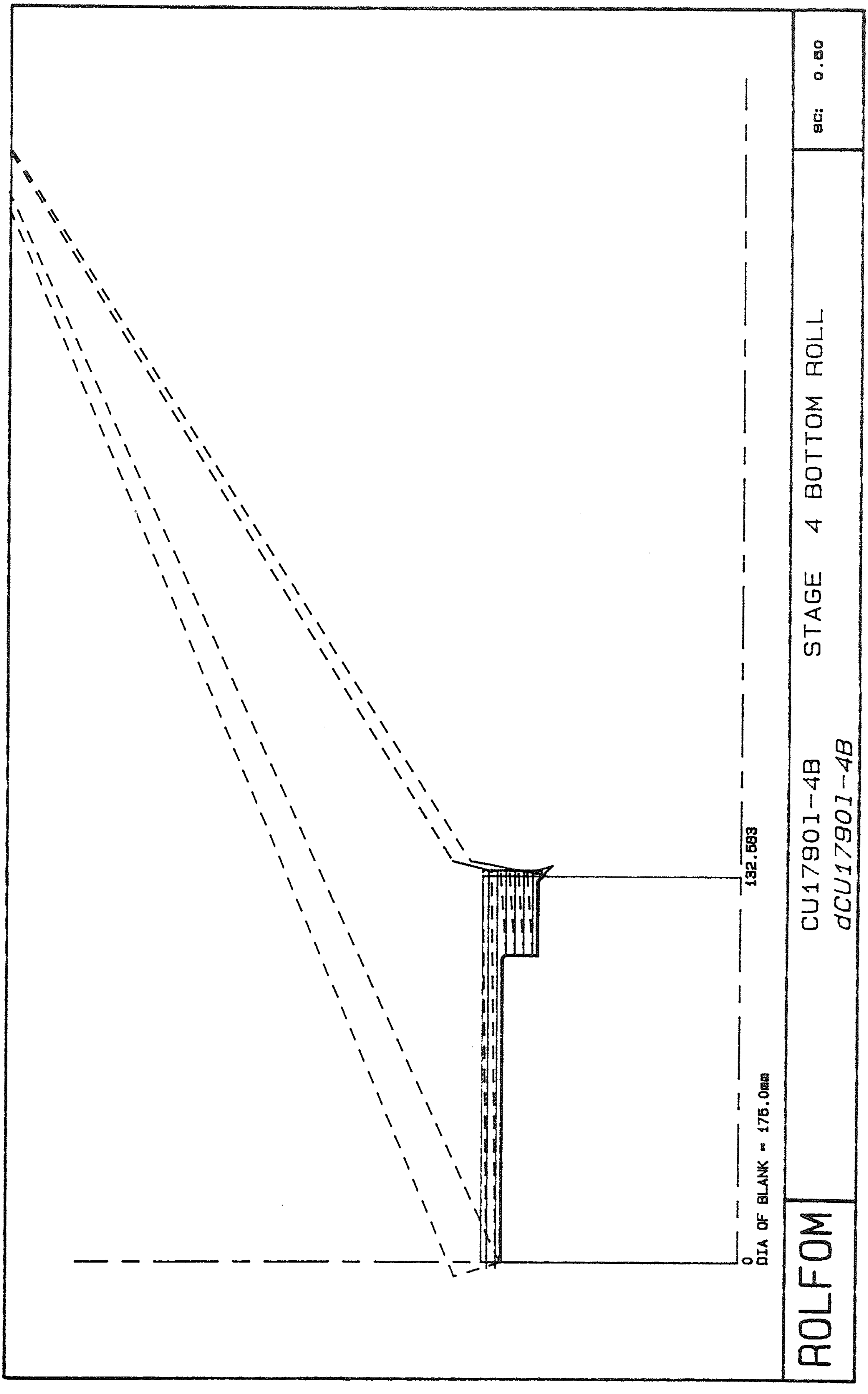

ROLFOM
CU1790-3B
dCU1790-3B
STAGE 3 BOTTOM ROLL.
SC: 0.50
149.847
0
DIA OF BLANK = 175.0mm



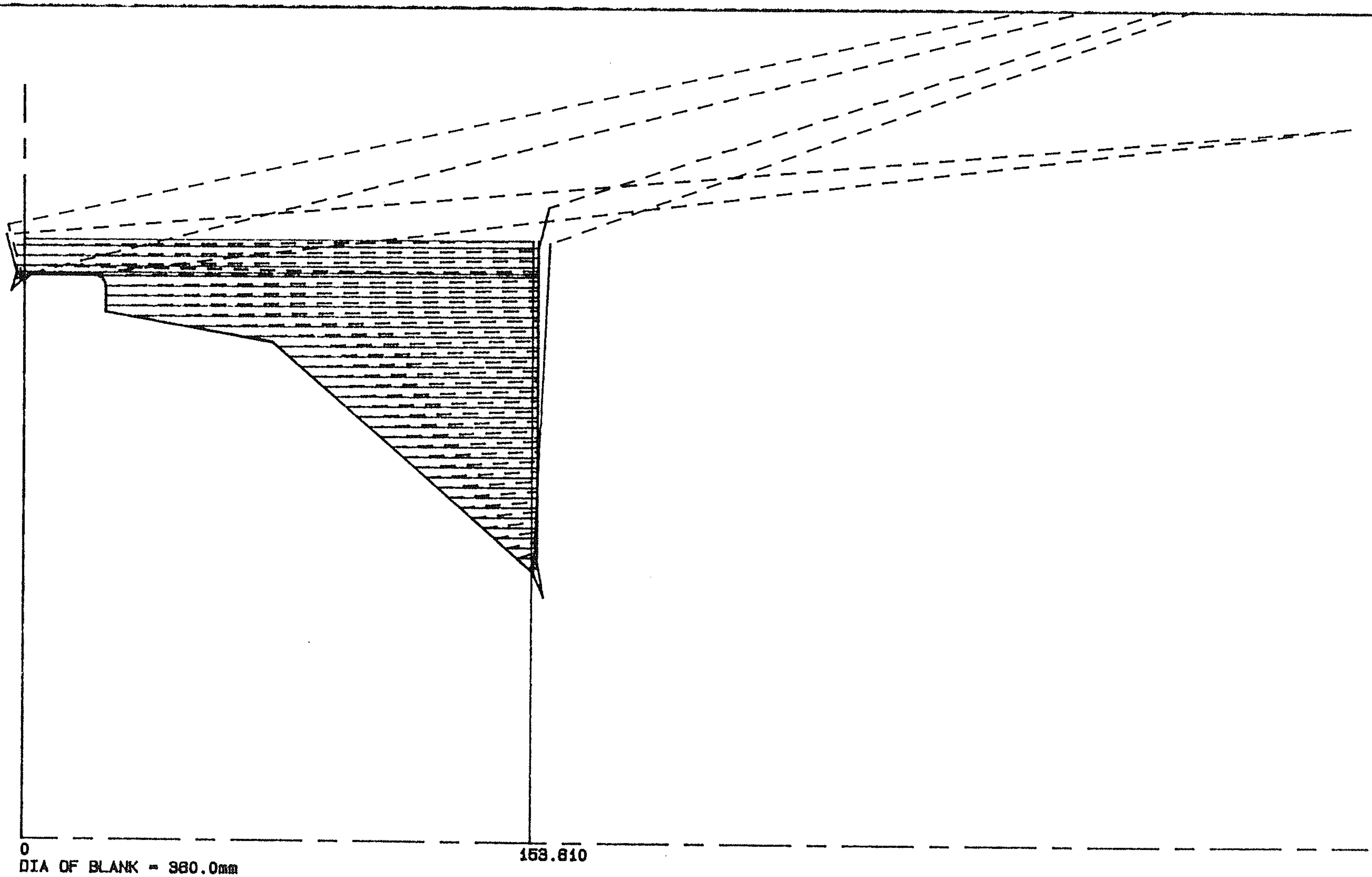
ROLFOM
CU1790-3R
dCU1790-3R
STAGE 3 RIGHT SIDE ROLL
SC: 0.50
DIA OF BLANK = 226.0mm
0
77.578



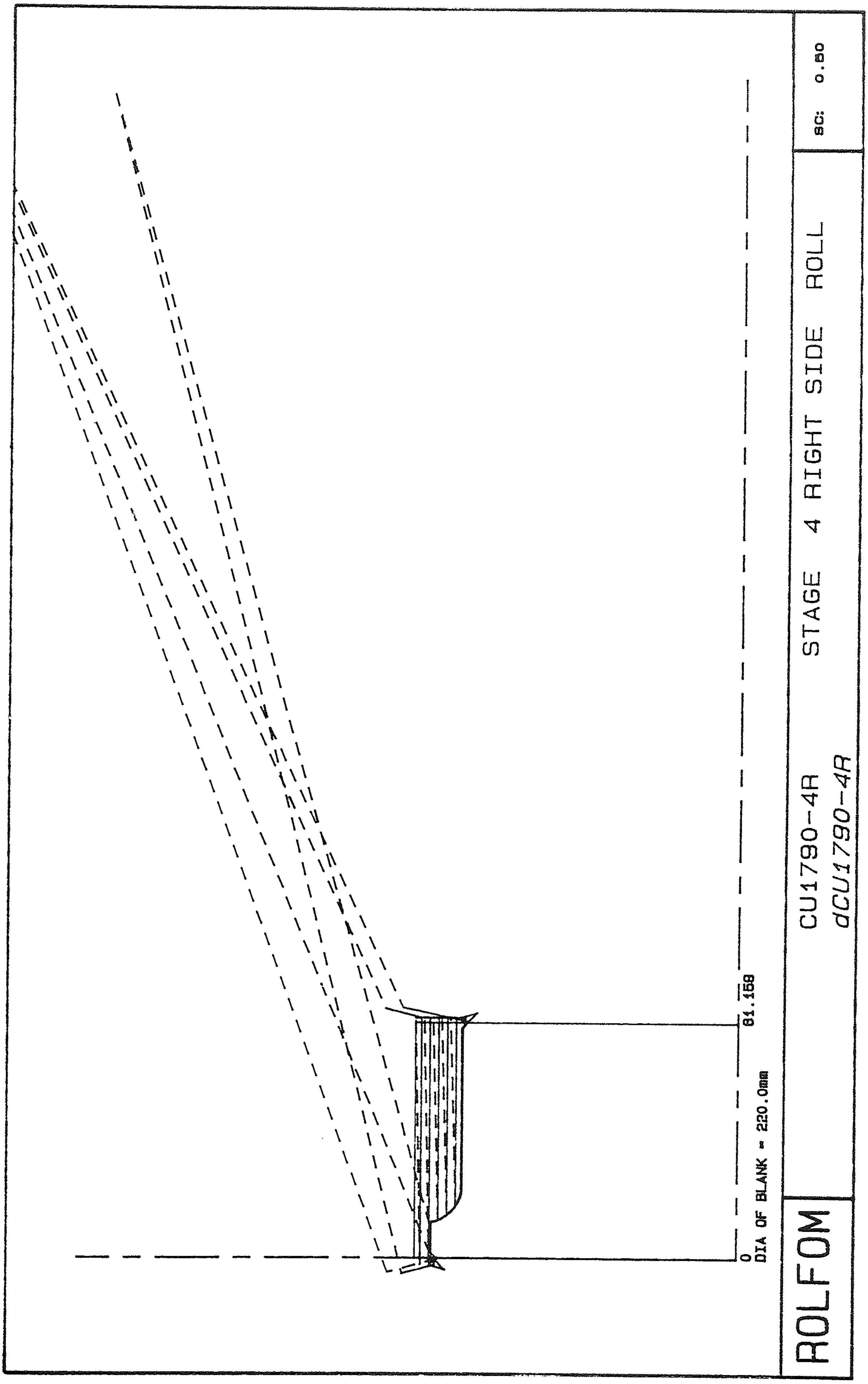
ROLFOM	CU17901-4T STAGE 4 TOP ROLL *dCU17901-4T*	SC: 0.50

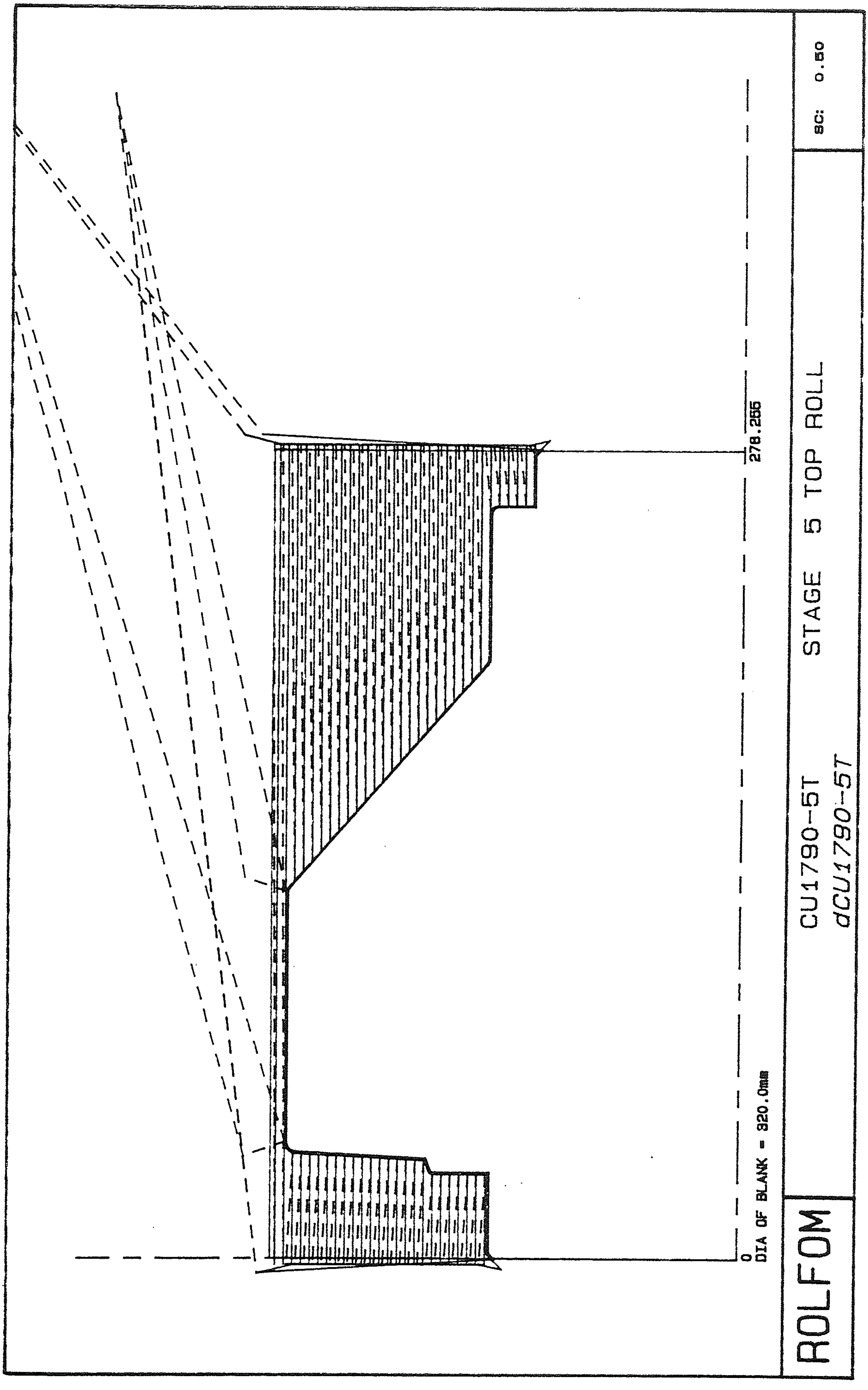


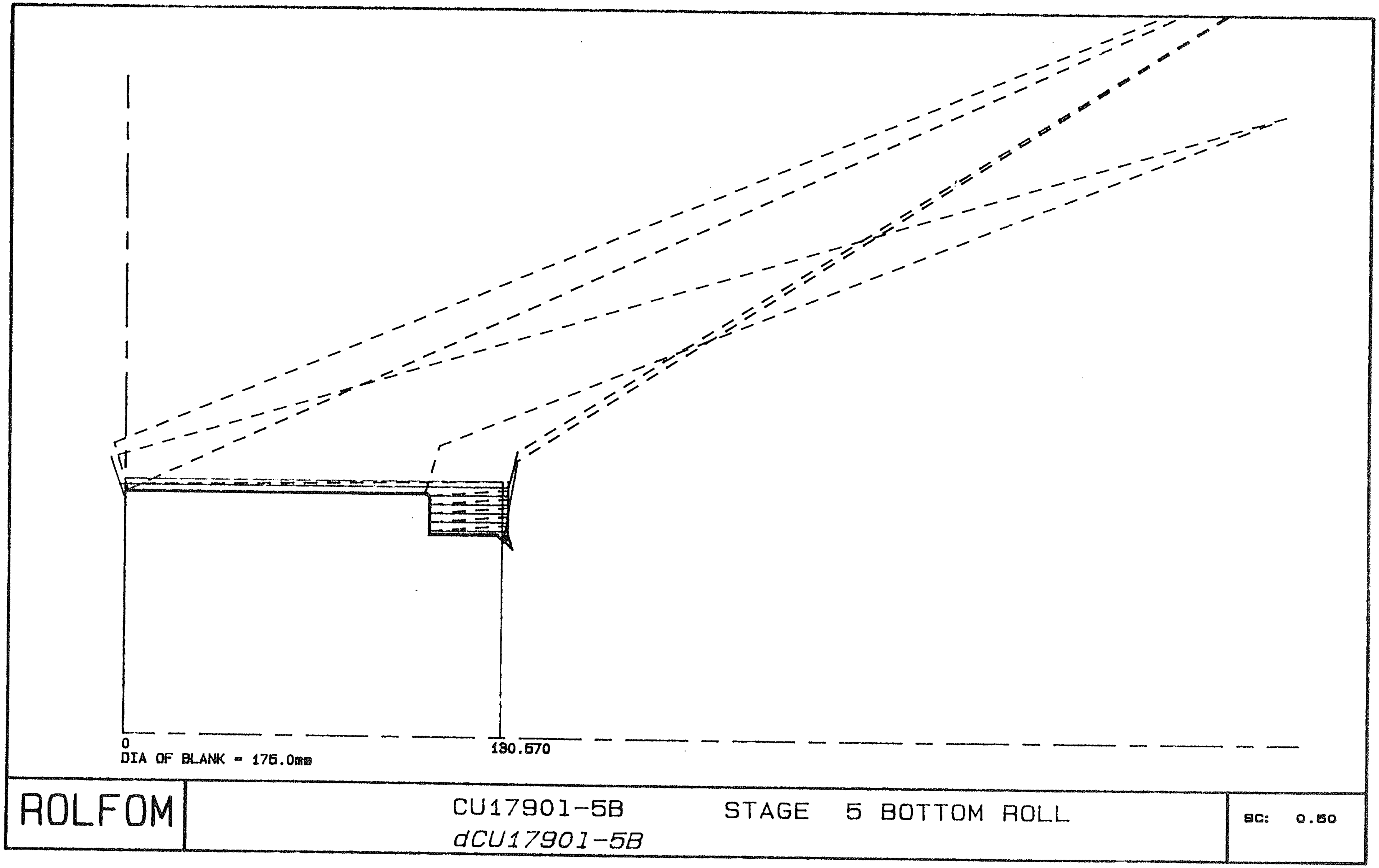
ROLFOM
CU17901-4B
dCU17901-4B
STAGE 4 BOTTOM ROLL
SC: 0.50
132.583
0
DIA OF BLANK = 175.0mm



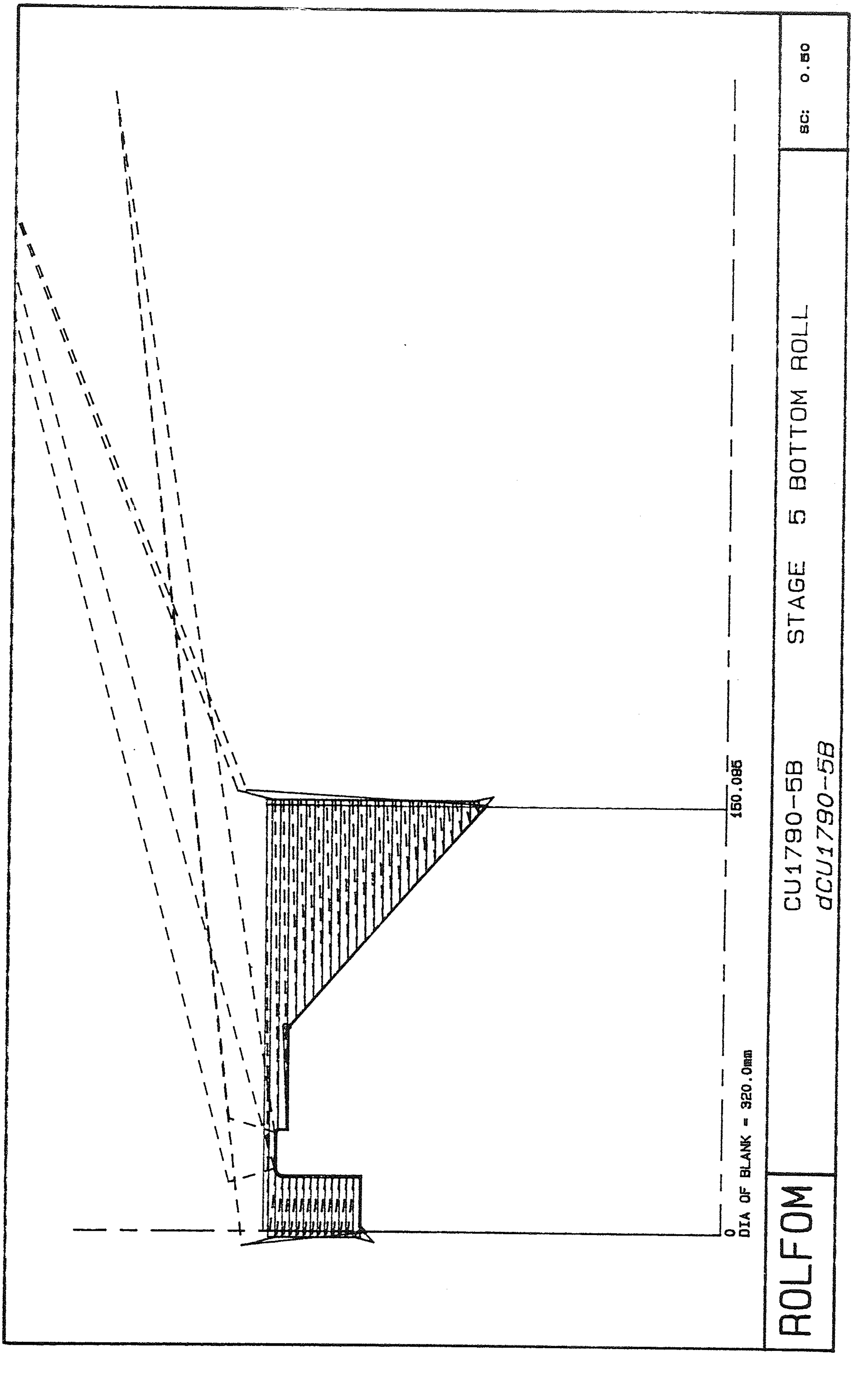
ROLFOM	CU1790-4B STAGE 4 BOTTOM ROLL *dCU1790-4B*	SC: 0.50

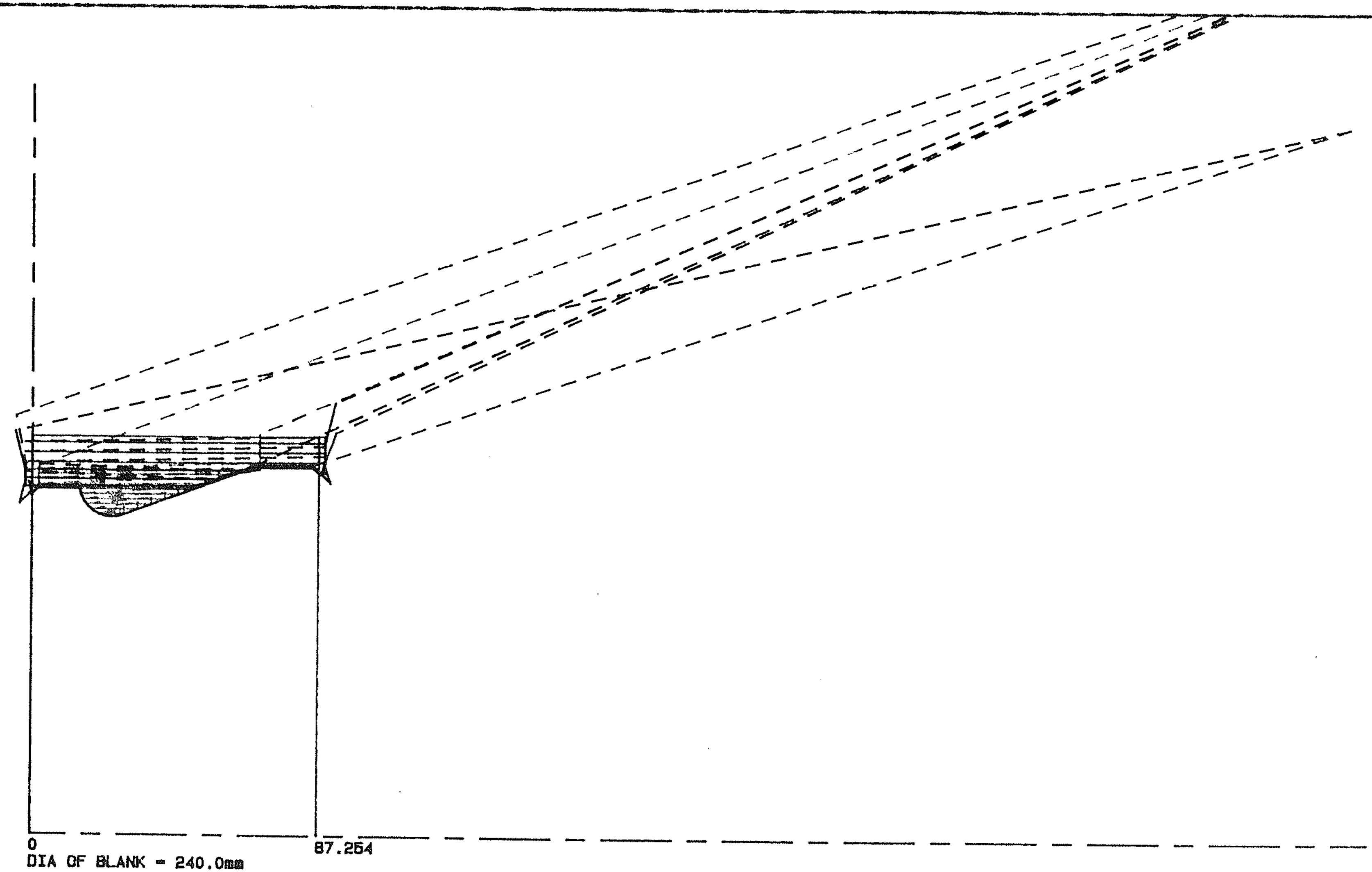

0
DIA OF BLANK = 220.0mm
81.158
ROLFOM
CU1790-4R
dCU1790-4R
STAGE 4 RIGHT SIDE ROLL
SC: 0.50


ROLFOM
CU1790-5T
dCU1790-5T
STAGE 5 TOP ROLL
SC: 0.50
DIA OF BLANK = 320.0mm
0
278.255



0
DIA OF BLANK = 175.0mm
180.570
ROLFOM
CU17901-5B
STAGE 5 BOTTOM ROLL
dCU17901-5B
SC: 0.50


ROLFOM
CU1790-5B
dCU1790-5B
STAGE 5 BOTTOM ROLL
SC: 0.50
150.095
0
DIA OF BLANK = 320.0mm

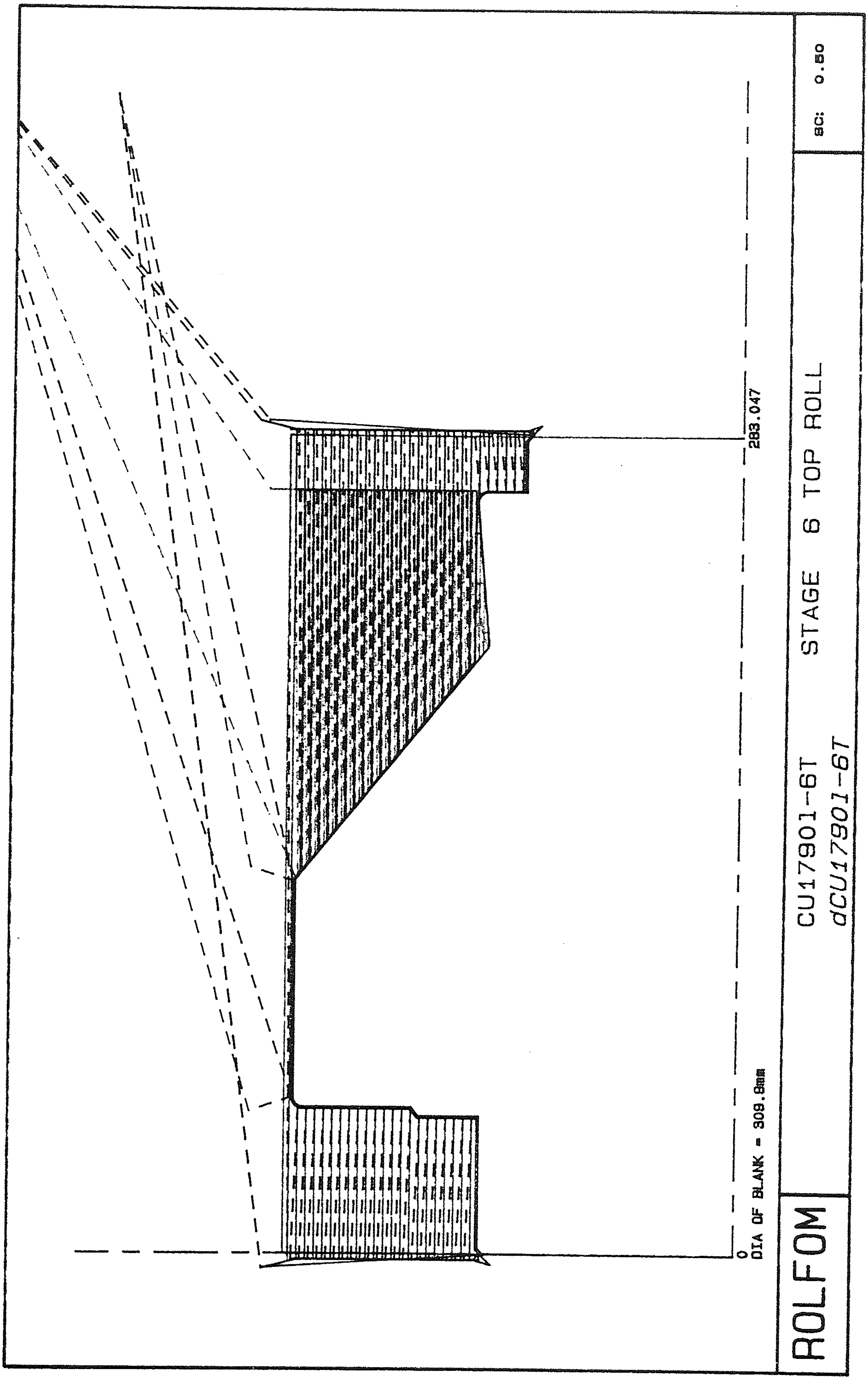


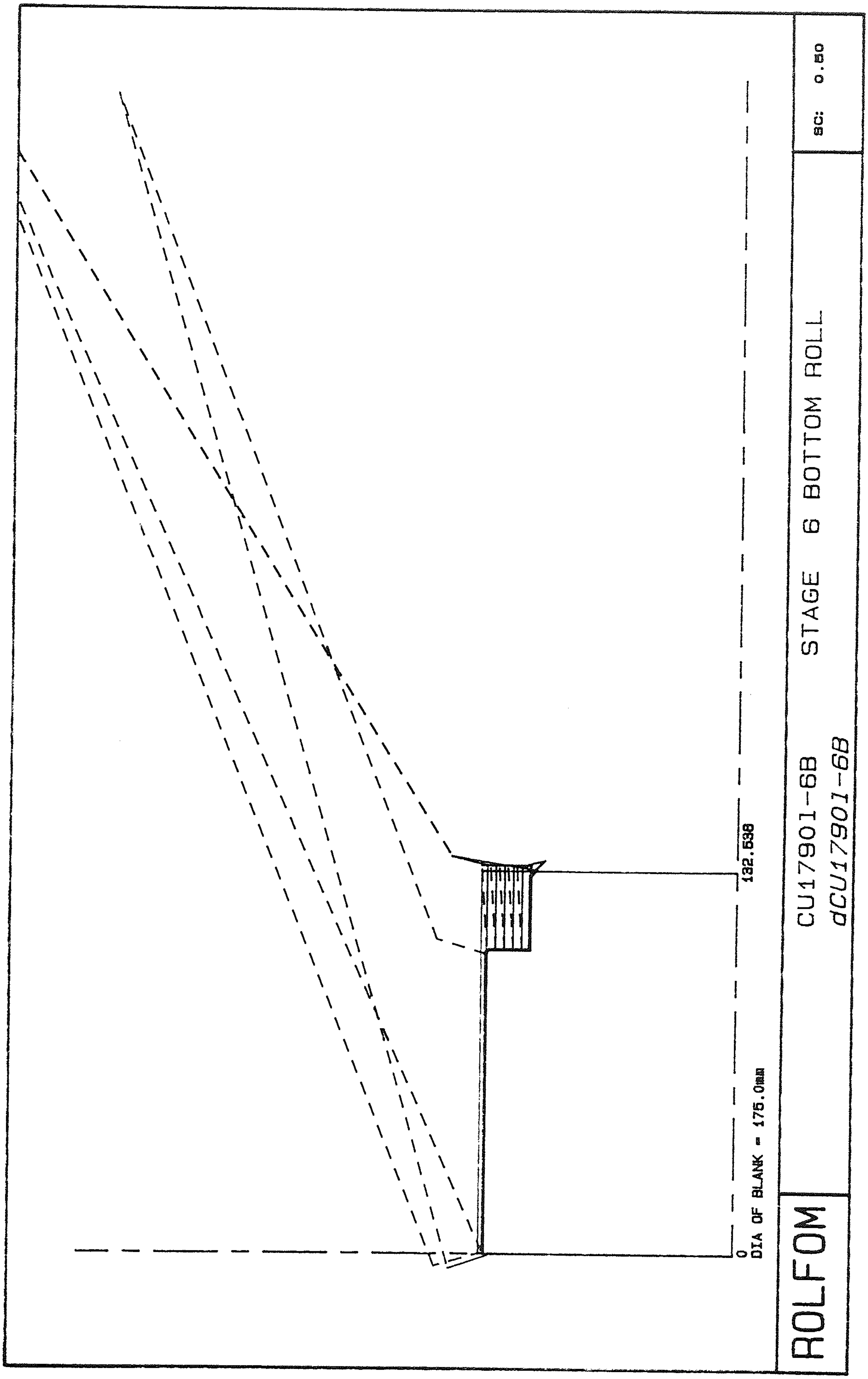
ROLFOM

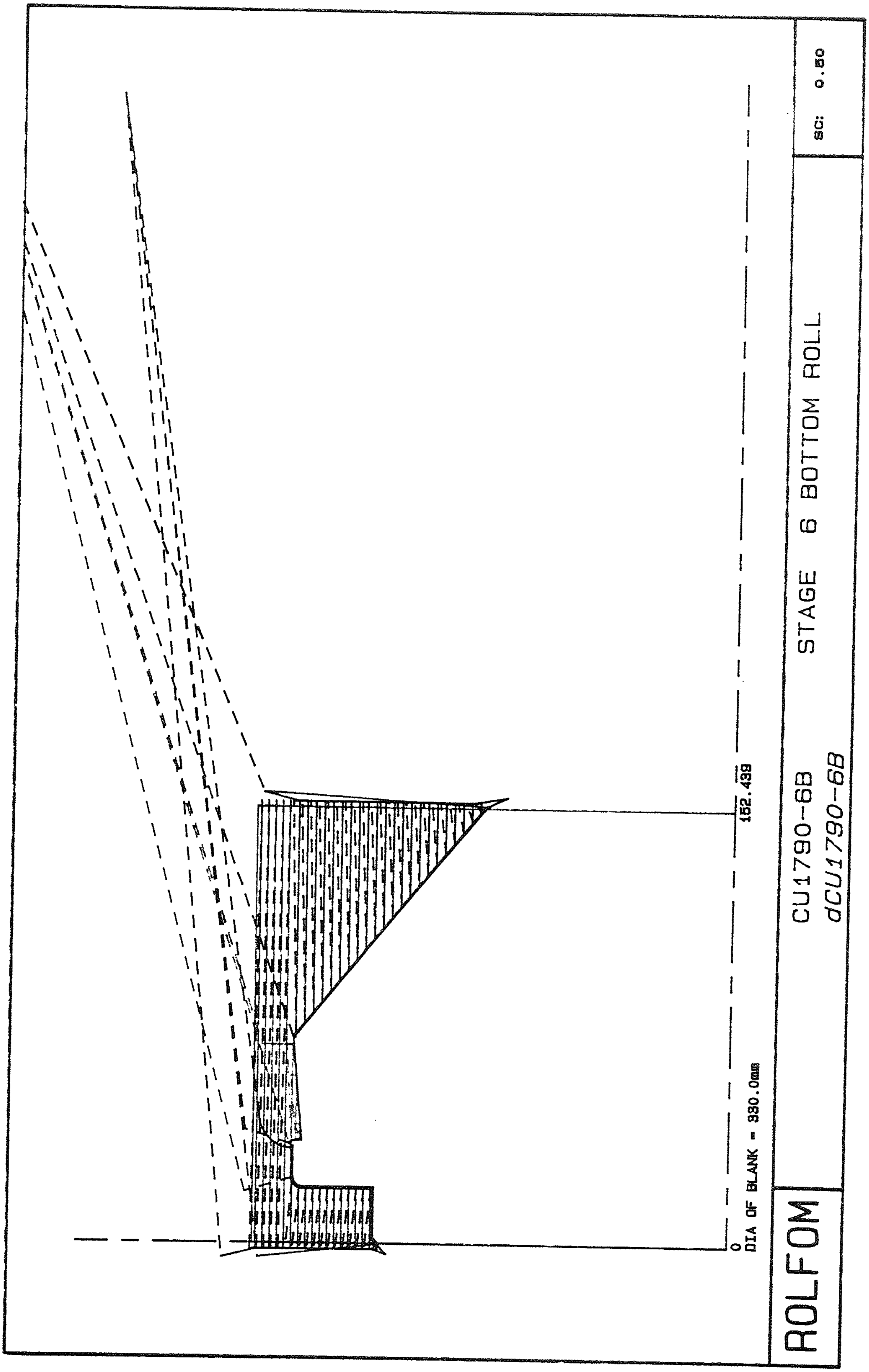
CU1790-5R STAGE 5 RIGHT SIDE ROLL

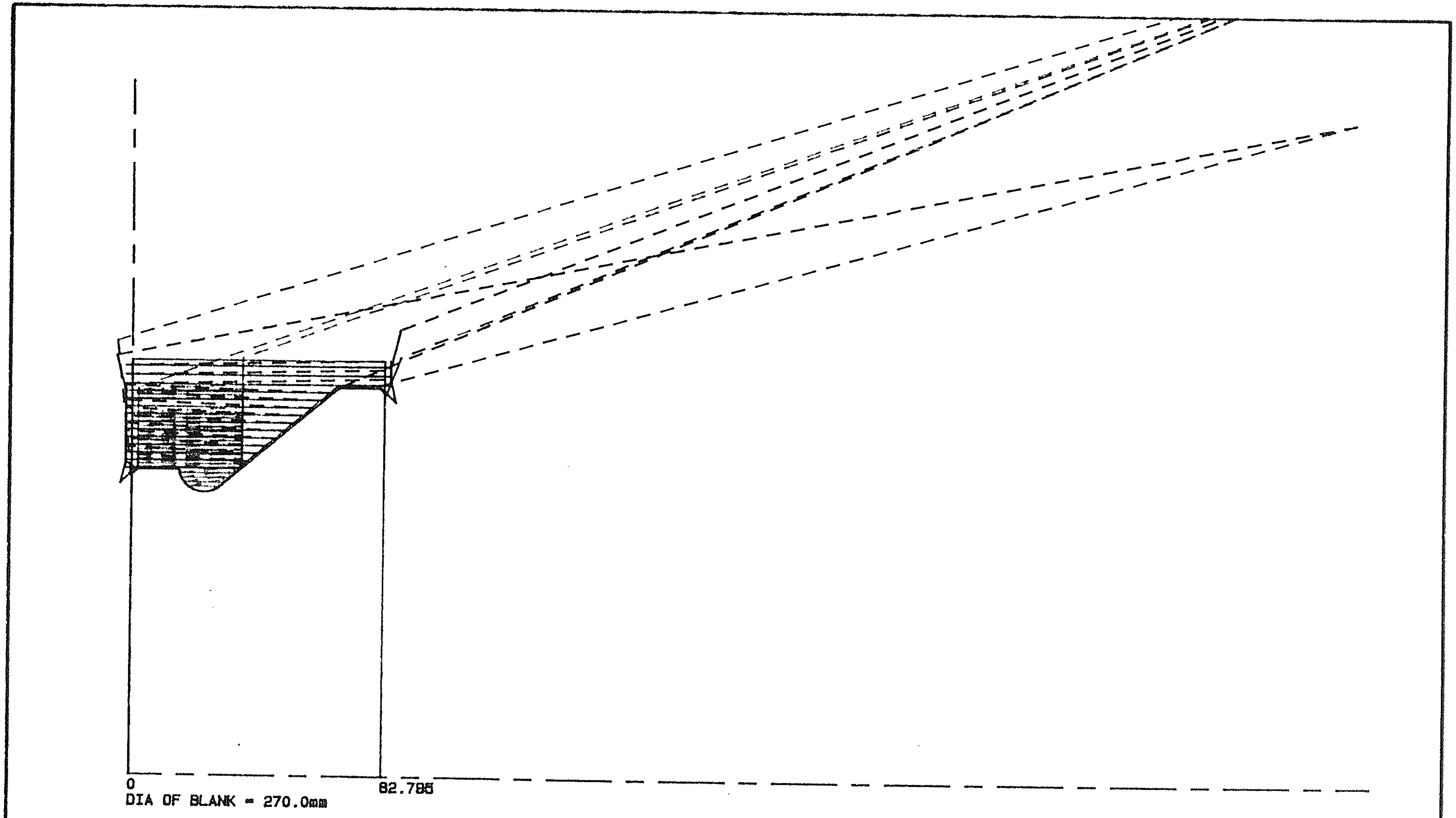
dCU1790-5R

SC: 0.50

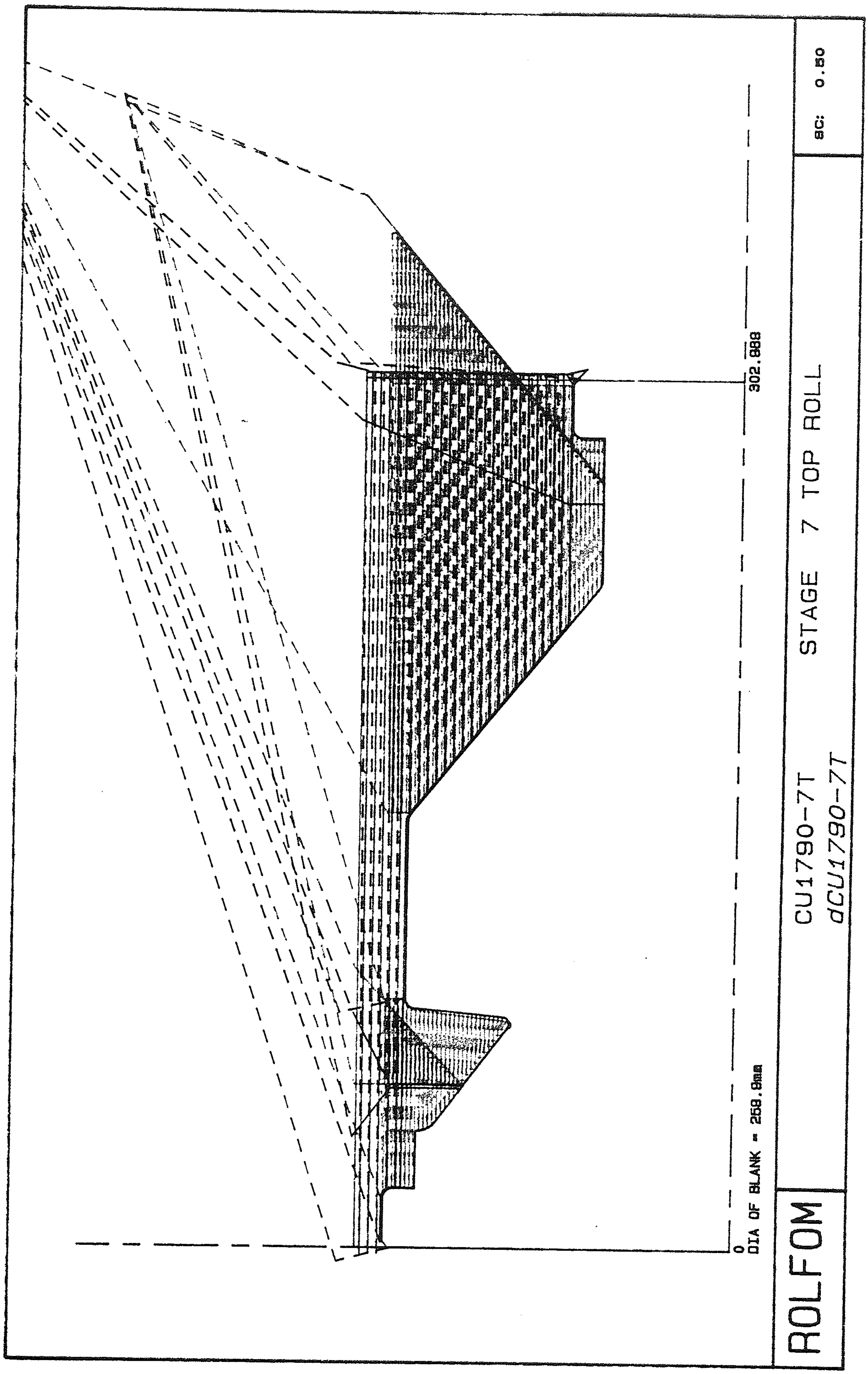

ROLFOM
CU17901-6T
dCU17901-6T
STAGE 6 TOP ROLL
SC: 0.50
283.047
0
DIA OF BLANK = 309.9mm

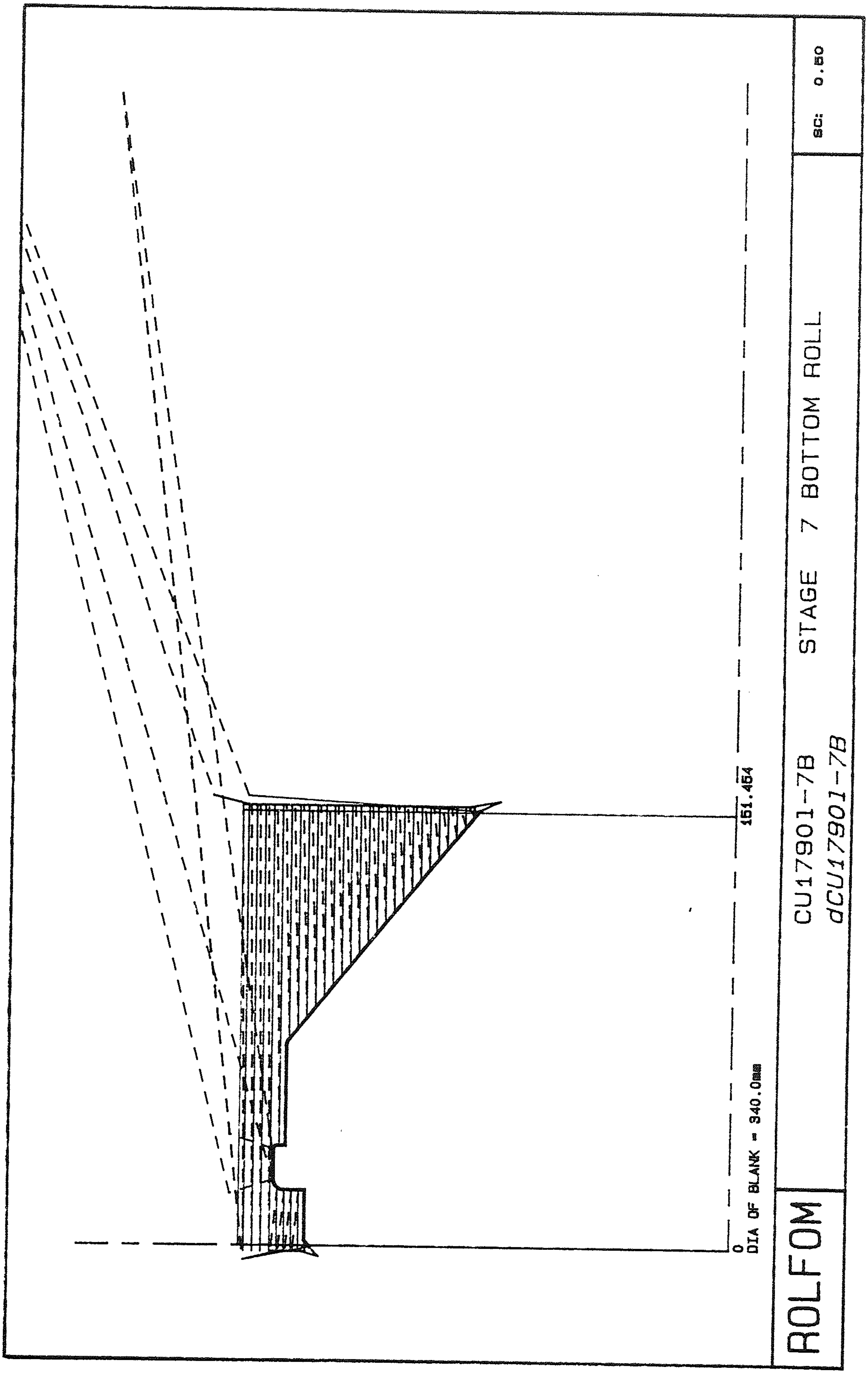

ROLFOM
CU17901-6B
dCU17901-6B
STAGE 6 BOTTOM ROLL
SC: 0.50
DIA OF BLANK = 175.0mm
0
132.538

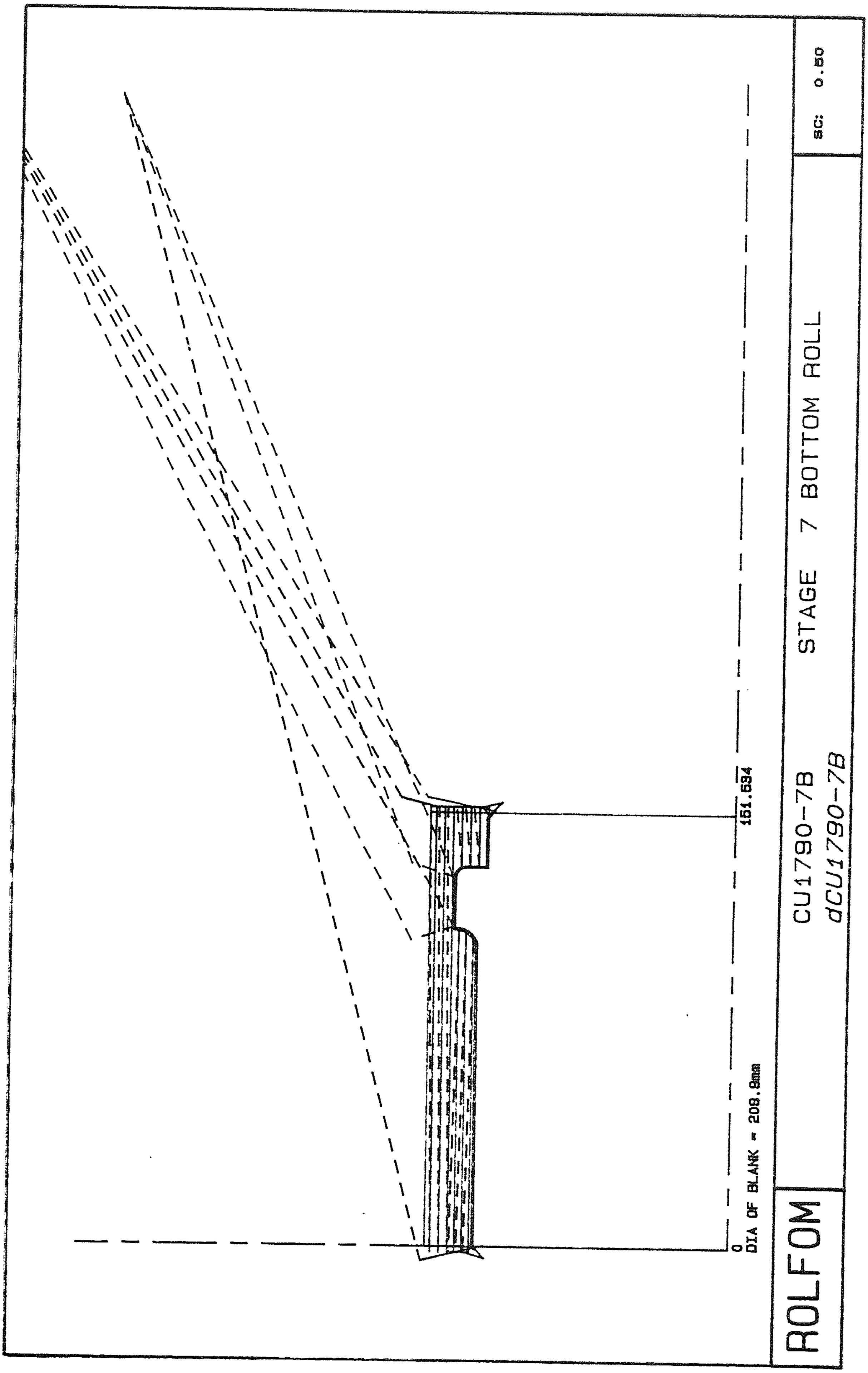

ROLFOM
CU1790-6B
dCU1790-6B
STAGE 6 BOTTOM ROLL
SC: 0.50
152.439
0
DIA OF BLANK = 330.0mm

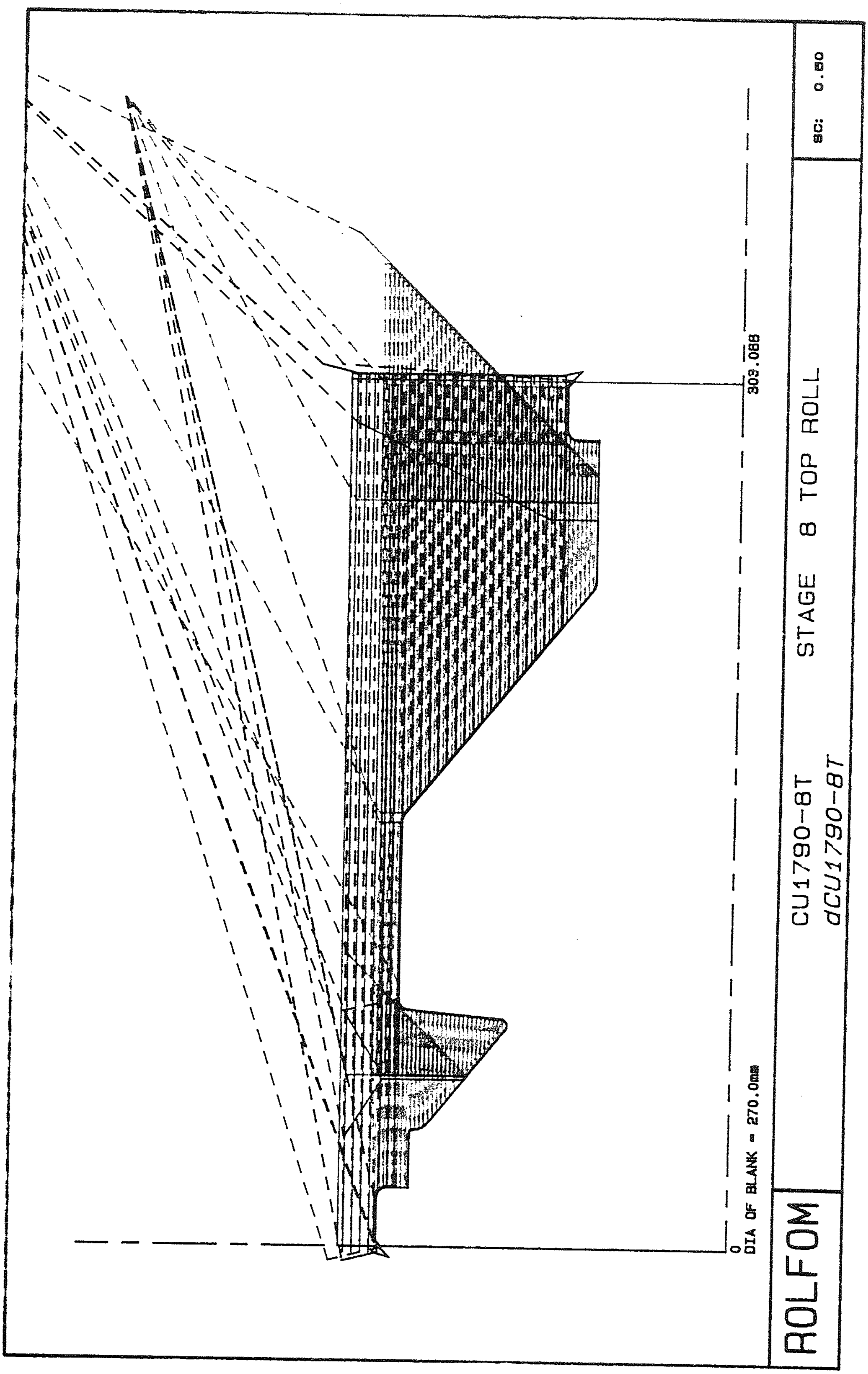


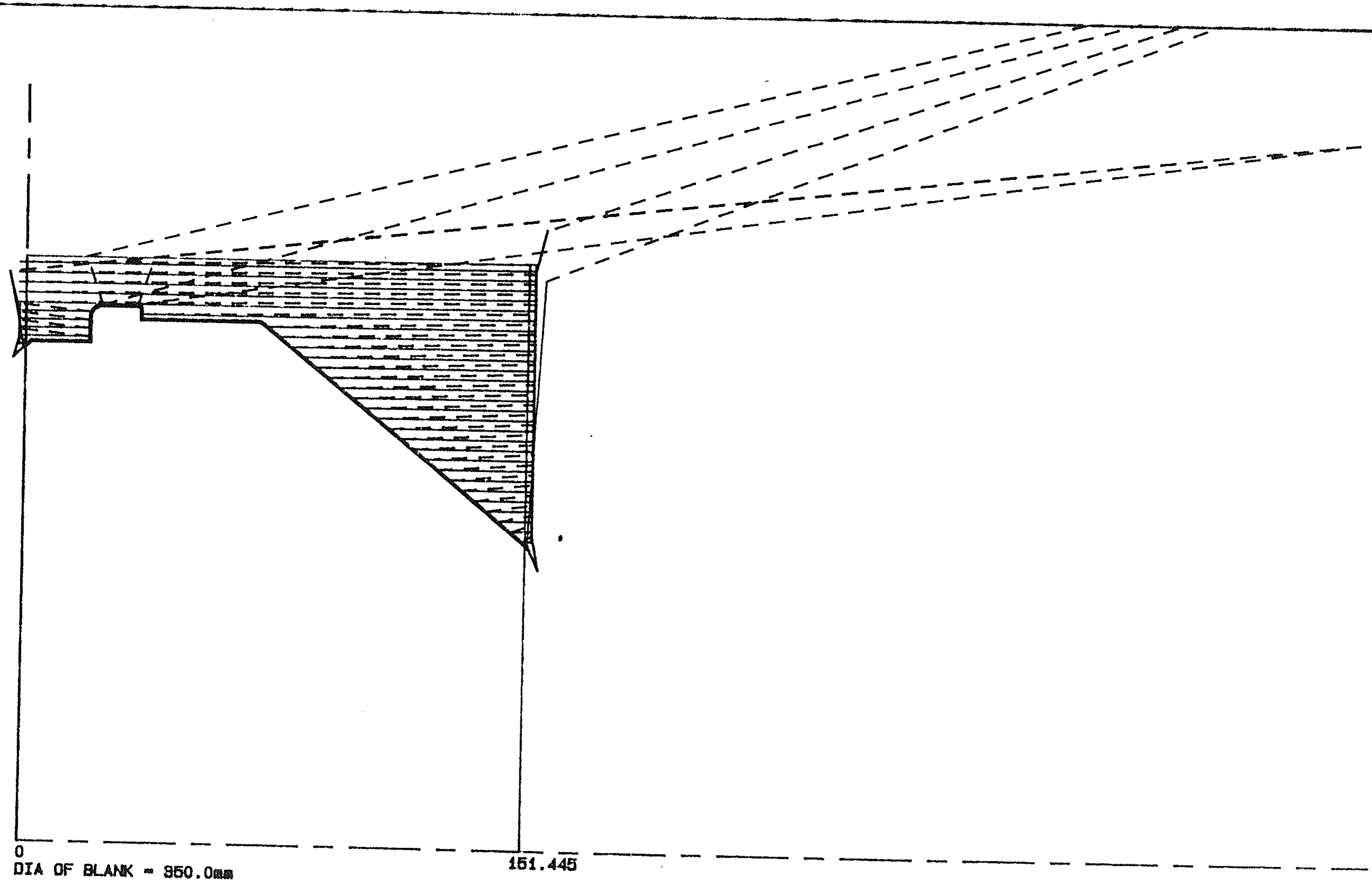
ROLFOM	CU1790-6R STAGE 6 RIGHT SIDE ROLL *dCU1790-6R*	SC: 0.50


0
DIA OF BLANK = 258.8mm
302.888
ROLFOM
CU1790-7T
dCU1790-7T
STAGE 7 TOP ROLL
SC: 0.50

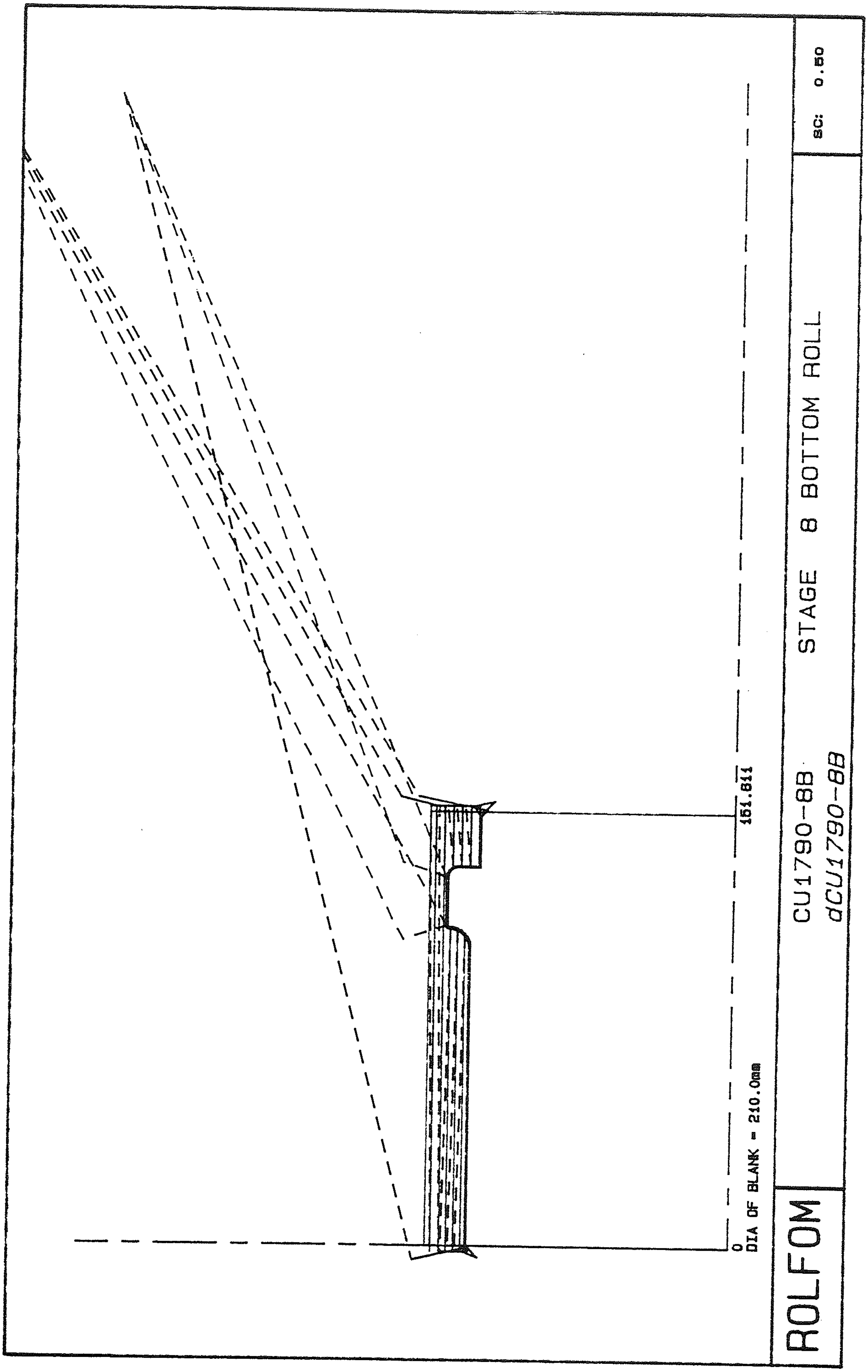

ROLFOM
CU17901-7B
dCU17901-7B
STAGE 7 BOTTOM ROLL
SC: 0.50
151.454
0
DIA OF BLANK = 340.0mm

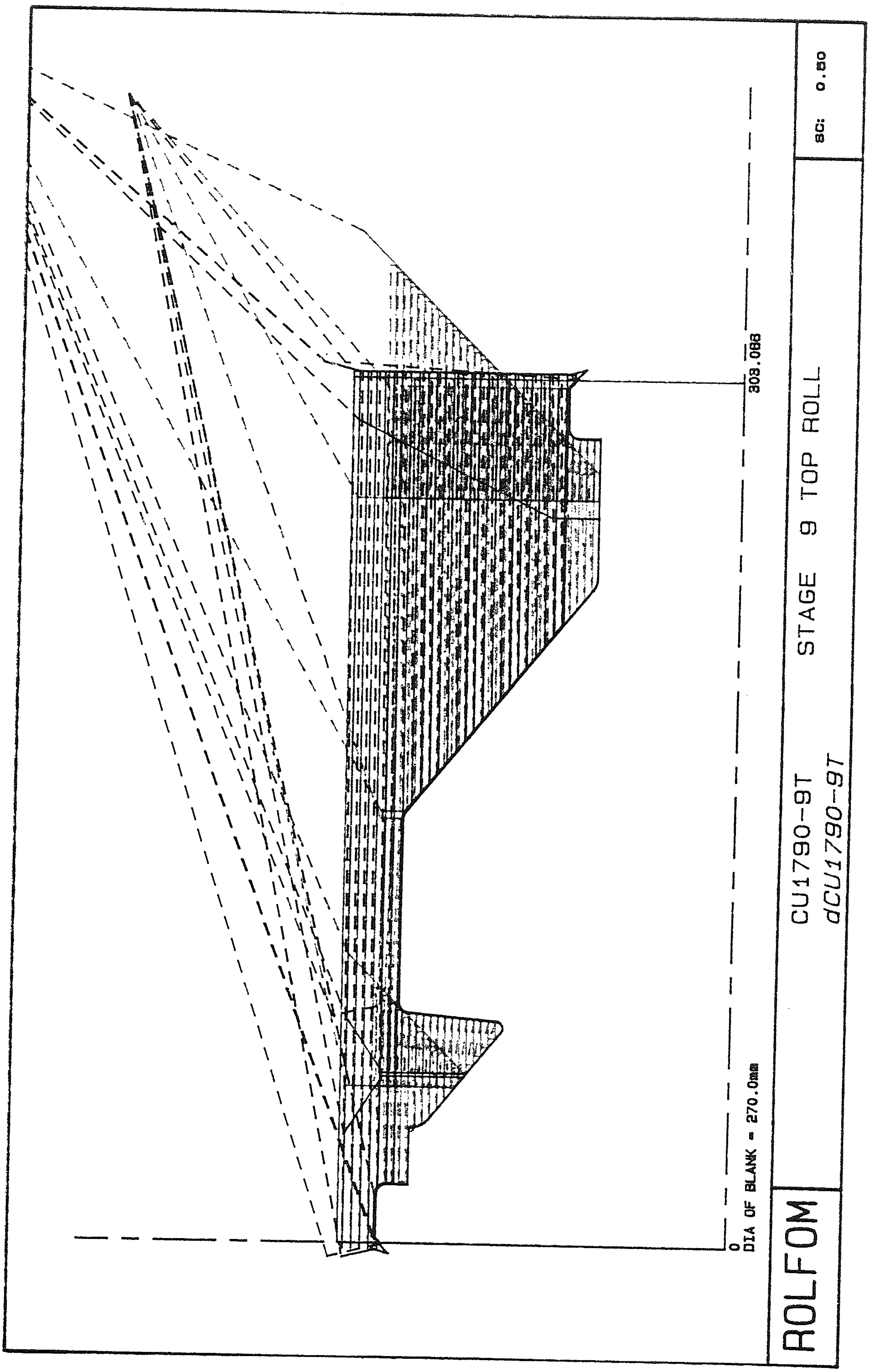

0
DIA OF BLANK = 208.8mm
151.534
ROLFOM
CU1790-7B
dCU1790-7B
STAGE 7 BOTTOM ROLL
SC: 0.50

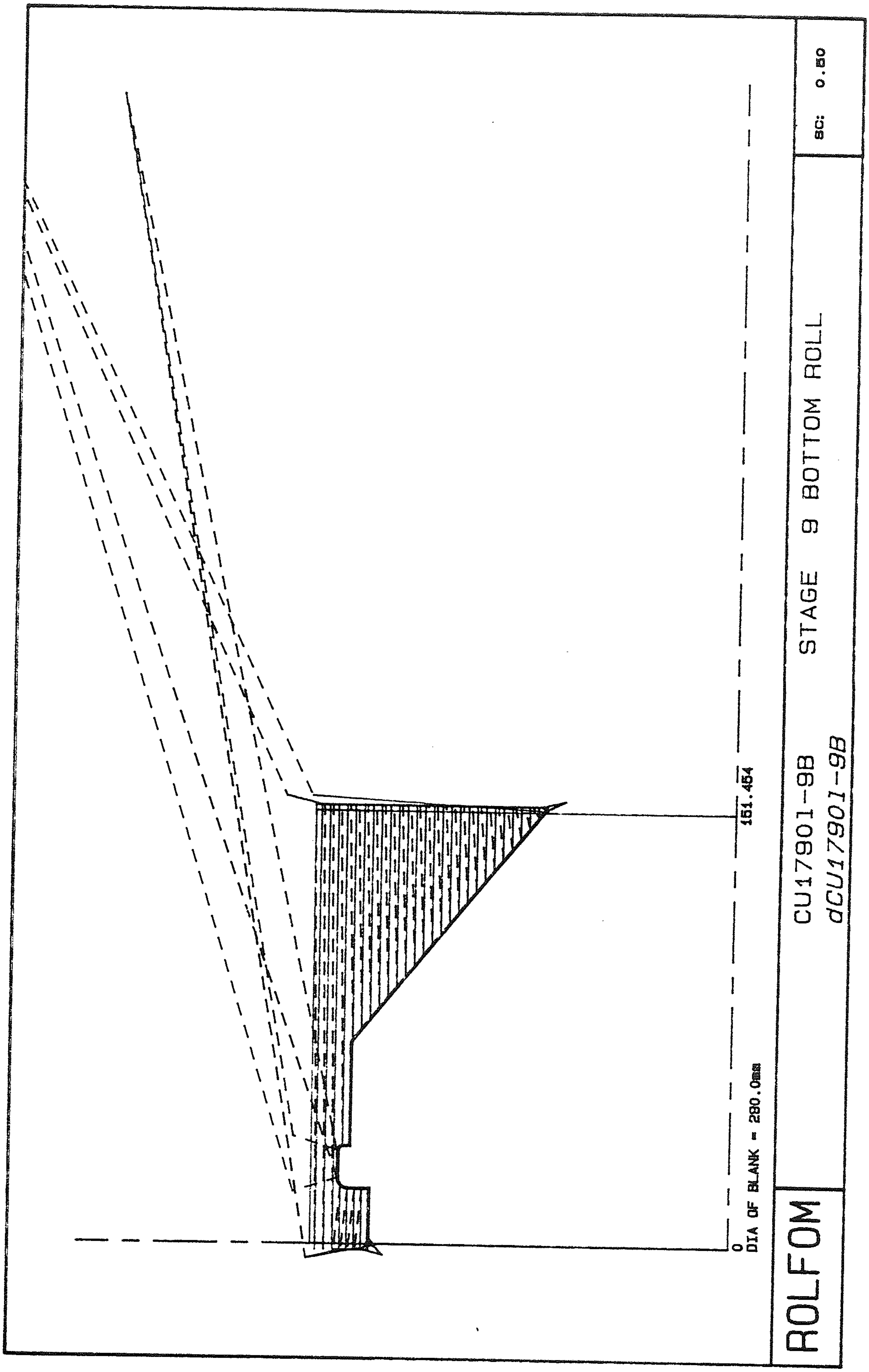

303.088
0
DIA OF BLANK = 270.0mm
ROLFOM
CU1790-8T
dCU1790-8T
STAGE 8 TOP ROLL
SC: 0.50

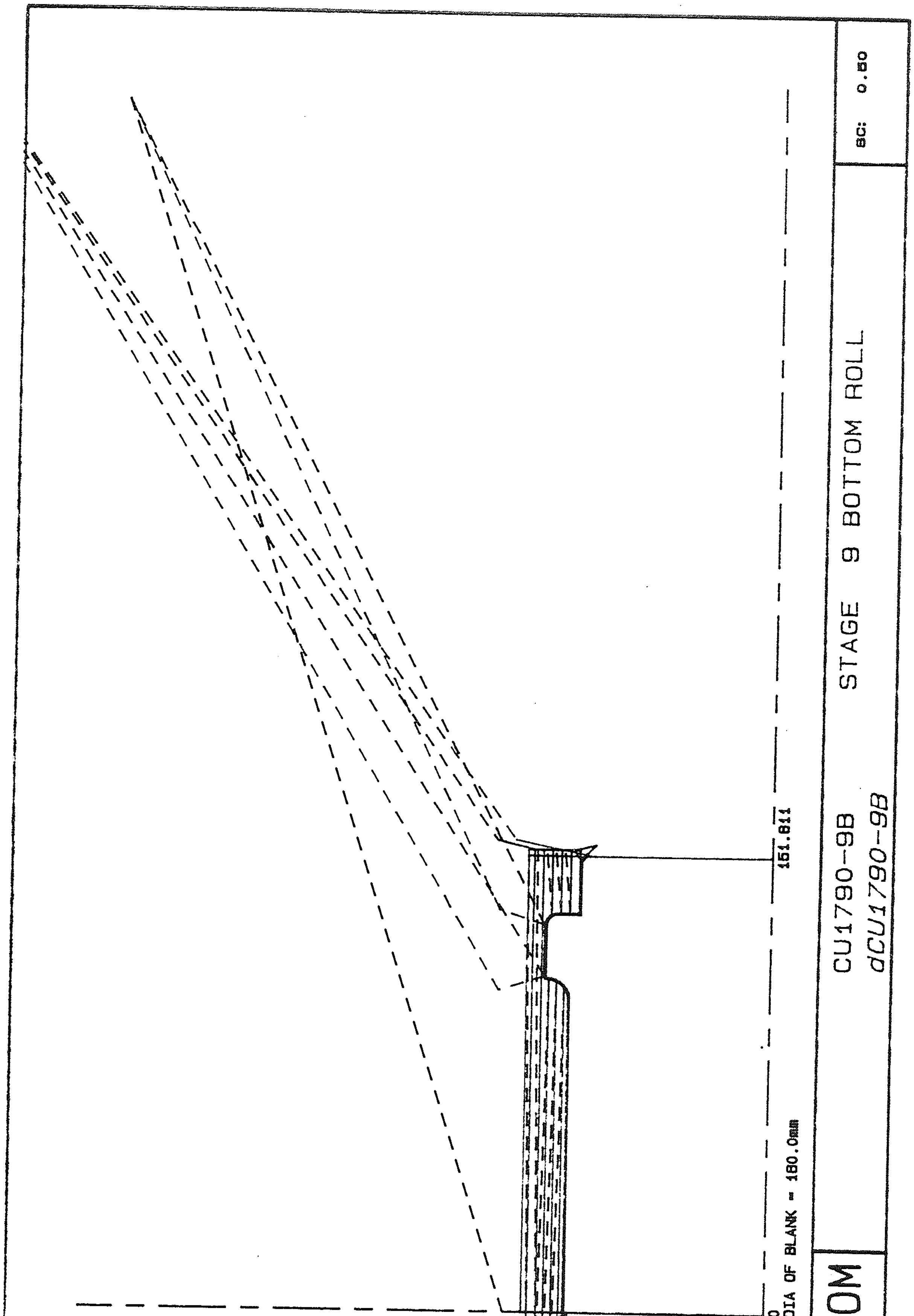


ROLFOM	CU17901-8B STAGE 8 BOTTOM ROLL *dCU17901-8B*	SC: 0.50


ROLFOM
CU1790-8B
dCU1790-8B
STAGE 8 BOTTOM ROLL
SC: 0.50
DIA OF BLANK = 210.0mm
0
151.811


ROLFOM
CU1790-9T
dCU1790-9T
STAGE 9 TOP ROLL
SC: 0.50
303.066
0
DIA OF BLANK = 270.0mm


ROLFOM
CU17901-9B
dCU17901-9B
STAGE 9 BOTTOM ROLL
SC: 0.50
DIA OF BLANK = 280.0mm
0
151.454


ROLFOM
CU1790-9B
dCU1790-9B
STAGE 9 BOTTOM ROLL
SC: 0.50
0
DIA OF BLANK = 180.0mm
151.811

APPENDIX 8

PROGRAM LISTINGS OF THE CAD/CAM SYSTEM SOFTWARE*

1. THE FINISHED SECTION PROGRAM (R1PLOT)

2. THE FLOWER PATTERN PROGRAM (R2PLOT)

3. THE TEMPLATE PROGRAM (R3PLOT)

4. THE ROLL DESIGN PROGRAM (R4PLOT)

5. THE ROLL EDITOR (ROEDIT)

6. THE ROLL MACHINING PROGRAM (CUTPLOT)

7. THE POSTPROCESSOR (MFPOST)

8. THE TAPE CHECKING PROGRAM (CHKTAP)

* Refer to the microfiches attached at the back cover

REFERENCES

REFERENCES

1. Hatvany, J., Newman, W.M., Sabin, M. "A world survey of Computer-Aided Design", CAD, Vol. 9, Part 2, April 1977, pp 79 to 98.

2. Arnold, E., Senker, P. "Computer-Aided Design in the UK Engineering Industry", CAD82 - 5th International Conference and Exhibition on Computers in Design Engineering, Butterworth, 1982, pp 1 to 7.

3. Angel, R.T. "Designing Tools for Cold Roll Forming", The Iron Age, 3 Nov 1949, pp 83 to 88.

4. American Society for Metals "Metals Handbook - Vol. 4 (Forming)", 8th Edition, 1969, pp 224 to 238.

5. American Society of Manufacturing Engineers "Tool and Manufacturing Engineers Handbook", 3rd Edition, 1976, pp 15-139 to 15-152.

6. Kirkland, W.G. "Cold Forming Practice in the United States", Iron and Steel Engineer, November 1959, pp 134 to 149.

7. Vanderploeg, E.J. "Roll-formed Sheet Metal Parts", Materials and Methods, Vol. 42, No. 5, Nov 1955, pp 100 to 102.

8. Vanderploeg, E.J. "Forming Metal Shapes by Cold Rolling", Machinery, Vol. 54, Feb 1948, pp 153 to 158.

9. Walker, A.C. (Editor) "Design and Analysis of Cold-Formed Sections", Intext Books, 1975.

10. Cadney, S. "Cold Roll Forming - the art that could do with some science", Production Engineer, Sept 1981, pp 19 to 21.

11. Sutherland, I.E. "SKETCHPAD : A Man-Machine Graphical Communication System", Proc. Spring Joint Computer Conference 1963, pp 329 to 346.

12. Prince, M.D. "Interactive Graphics and Computer-Aided Design", Addition-Wesley Publishing Company, 1971.

13. Coons, S.A. "Surfaces for Computer-Aided Design of Space Forms", Project MAC TR 41, June 1967, Massachusetts Institute of Technology.

14. Bezier, P.E. "How Renault uses Numerical Control for Car Body Design and Tooling", Proc. Soc. of Automotive Enigneer Congress, Detroit, 1968.

15. Forrest, A.R. "Computational Geometry", Proc. Royal Soc. of London, Vol. A321, 1971, pp 187 to 195.

16. Chasen, S.H. "The Introduction of Man Computer Graphics into the Aerospace Industry", Proc. Fall Joint Computer Conference 1965, pp 883 to 892.

17. Bliss, F.W. "Selecting and Inplementing a Turnkey Graphics System", IEEE Computer Graphics and Applications, April 1981, pp 55 to 70.

18. Kirby, P. "Desktop Graphics", Datamation, May 1979, pp 164 to 170.

19. Spur, G., et al. "A Survey about Geometric Modelling Systems", Annuals of CIRP, Vol. 28/2, 1979, pp 519 to 538.

20. Spur, G., Dassler, R., Gausemeier, J. "Computer Aided Handling of Geometric Data for Design and Manufacturing", Proc 18th MTDR Conference, pp 725 to 732.

21. Voelcker, H.B., Aristides, A.G.R. "Geometric Modelling of Mechanical Parts and Processes", Computer, Vol. 10, No. 12, Dec 1977, pp 48 to 57.

22. Brown, C.M. "PADL-2 : A Technical Summary", IEEE Computer Graphics and Applications, Vol. 2, No. 2, March 1982, pp 69 to 84.

23. Fitzerald, W., Gracer, W., Wolfe, R. "GRIN : Interactive Graphics for modelling Solids", IBM J. Research and Development, Vol. 25, No. 4, July 1981, pp 281 to 294.

24. Boyse, J.W., Gilchrist, J.E. "GMSolid : Interactive Model for Design and Analysis of Solids", IEEE Computer Graphics and Applications, Vol. 2, No. 2, March 1982, pp 27 to 40.

25. Hamilton, C.H. French, H.T. "Graphic Aids for Finite-Element Modelling", Machine Design, Oct 21 1982, pp 71 to 75.

26. Bezier, P. "Numerical Control : Mathematics and Applications", John Wiley and Sons Ltd.,1972.

27. Charnley, C.J. "NUMERICAL CONTROL : Present status and future potential of Numerical Engineering Methods. Part 6 : Current Trends and Future Potential", Engineers' Digest, Vol. 39, No. 5, May 1978, pp 9 to 17.

28. Klahorst, H.T. "Flexible Manufacturing Systems and their Applications", Industrial Engineering, Vol. 13, No. 11, Nov 1981, pp 112 to 117.

29. Houtzeel, A. "Elements of an Effective Computer Assisted Process Planning System", Industrial Engineering, Vol. 13, No. 11, Nov 1981, pp 60 to 64.

30. Weill, R., Spur, G., Eversheim, W. "Survey of Computer-Aided Process Planning Systems", Annuals of CIRP, Vol. 31/2, 1982, pp 539 to 551.

31. El-Midany, T.T., Davies, B.J. "AUTOCAP - A Dialogue System for Planning the the Sequence of Operations for Turning Components", Int. J. MTDR, Vol 21, No. 3/4, pp 175 to 191.

32. Anon "Robots, Robots, everywhere", Production Engineer, May 1983, pp 15 to 16.

33. Anon "Multi-task Robot serves Machine Cell", Industrial Robot, Vol 10, No. 1, Mar 1983, pp 20 to 23.

34. Donelson, W.S. "MRP - Who needs it", Datamation, May 1979, pp 185 to 194.

35. Anon "Computer takes on MRP, Savings Multiply", Industrial Engineering, March 1979, pp 26 to 27.

36. Chassen, R.J., Mastrom, E.M. "Capacity Planning as a Productivity Tool in the Automated Factory", Industrial Engineering, Vol. 24, No. 4, April 1983, pp 73 to 80.

37. Gand, H., Cook, M.E. "Choosing an MRP System", Datamation, January 1983, pp 84 to 98.

38. Kerry, T.M. "Special Software for NC turning", American Machinist, Feb 18, 1974, pp 78 to 80.

39. Takeyama, H., et al. "An Automatic Programming System for Lathe Operations MELTS" , Annuals of CIRP, Vol. 22/1, 1973, pp 167 to 168.

40. "COMPACT II Programming Manual", Manufacturing Data Systems International UK Ltd., 1980.

41. Craig, D.P., et al. "The Use of Interactive CAD Techniques and NC Machining in Mould Design and Manufacture", Proc. of 18th MTDR Conference, pp 743 to 750.

42. Eversheim, W., Holz, B. "Computer Aided Programming of NC Machine Tools by using the System AUTAP-NC", Annuals of CIRP, Vol. 31/1, 1982, pp 323 to 327.

43. Sata, T., Okada, N., Matsushima, K. "The Integration of CAD and CAM by application of artificial intelligence techniques", Annuals of CIRP, Vol. 31/1, 1982, pp 329 to 332.

44. Spur, G., et al. "CAPSY - A Dialogue System for Computer Aided Manufacturing Process Planning", Proc. of 19th MTDR Conference, pp 281 to 288.

45. Murry, J.L. "CAM-X - A Linked CAE System", CAD82 - 5th International Conference and Exhibition on Computers in Design Engineering, Butterworth, 1982, pp 44 to 53.

46. Kobayashi, K. "The Practical Approach to an Unmanned FMS Operation", Industrial Robot, Vol. 9, No. 1, March 1982, pp 36 to 40.

47. Riedlinger, T. "Roll Form Die Design Time Slashed from Days to Hours", Modern Metals, May 1980, pp 38 to 40.

48. Rhodes, A. "Computer-Aided Design for Cold Roll Forming", Production Engineer, Sept 1981, pp 32 to 34.

49. Ellen, C.H., Yuen, W.Y.D. "Development in Computer-Aided Design for Roll Forming", Proc. of the 2nd Internatioal Conference on Rotary Metalworking Process, 1982, pp 175 to 186.

50. Ng, K.H. "Computer-Aided Geometric Design of Form-Rolls For NC Manufacture", Ph.D Thesis, The University of Aston in Birmingham, 1981.

51. "Extended FORTRAN - ICL Student Edition", Technical Publication 4269A, International Computers Ltd., 1974.

52. "GINO-F User Manual", Computer Aided Design Centre, Cambridge, 1976.

53. "ICL PERQ : Introduction to PERQ", Edition 1, RP10100, International Computers Ltd., 1982.

54. "ICL PERQ FORTRAN 77", Edition 1, RP10108, International Computers Ltd., 1982.

55. "ICL Publication Notice, GINO-F User Manual Issue 210", RP1196, International Computers Ltd.

56. Wong, T.N. "Production of Form-rolls by CNC", M.Sc. Thesis, The University of Aston in Birmingham, 1981.

57. "ICL PERQ GRAFIKS Reference Manual", RP10111, Edition 1, International Computers Ltd., 1982.

58. "Programming Manual Mori-Fanuc 6T", Mori Seki Co. Ltd.

59. "FAPT Turn/Mill Manual", Fujitsu Fanuc Ltd., 1979.

60. Sim, R.M. "The Design and Specification of Post-processors for Numerical Control Programming", CAD, Vol. 6, No. 4, October 1974, pp 221 to 225.