Expert Control of Deep Hole Machining System

An Artificial Intelligence Technique for Real-time Control of Deep Hole Machining Process

Perdur S. Subramanya

A Thesis
in
The Department
of
Mechanical Engineering

Presented in Partial Fulfilment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montreal, Quebec, Canada

May 1994

ISBN 0-315-97668-3

CHAPTER 1

AUTOMATION OF MACHINING PROCESS

1.1 INTRODUCTION

Automation of manufacturing process has become one of the most important aspects of global economic competition since the World War II. Importance of automation has continued to increase along with the global competition and it has been one of the top lists of technologies identified, by the United States of America, as critical to national economic competitiveness and security in recent studies [Thurow, 1992]. Manufacturing activities constitute nearly a third of the value of all the goods and services produced in industrial nations [Galip Ulsoy et al., 1993].

Machining, or metal removal, processes occupy a very significant part of manufacturing activities. The emphasis towards automation of machining process is due to a continuous demand for higher product quality and higher productivity. The economic benefits of automation is illustrated in Figure 1.1 [Galip Ulsoy et al., 1993]. Consider an investment of $30,000 for automating a machine tool of worth $300,000 during its life time. Assuming an increase in productivity of only 10% (higher in reality) due to increased metal removal rate, lower rejections, longer tool life, lower down time, improved quality, lower labour cost etc., leads to an economic return of $3,000,000 during the life time of the machine tool. This amplification in returns for a small investment provides the much needed competitive edge for the manufacturing industry.

1.2 DEEP-HOLE MACHINING PROCESS

The constant drive for higher accuracy, surface finish and at the same time higher productivity to withstand the economic competition has lead to many improvements in the cutting tools and cutting methods. Considering the enormous amount of hole production in a manufacturing activity, the strive for better drilling tools and procedures, were always at the fore front of such drives. This constant quest has lead to the development of deep hole drilling/machining methods. Even though, the deep hole drilling procedure was developed for holes with a length to diameter ratio of greater than 5 [Griffiths, 1977], the higher quality of holes produced by this method has lead to its application for much smaller length to diameter ratios [Fink, 1977].

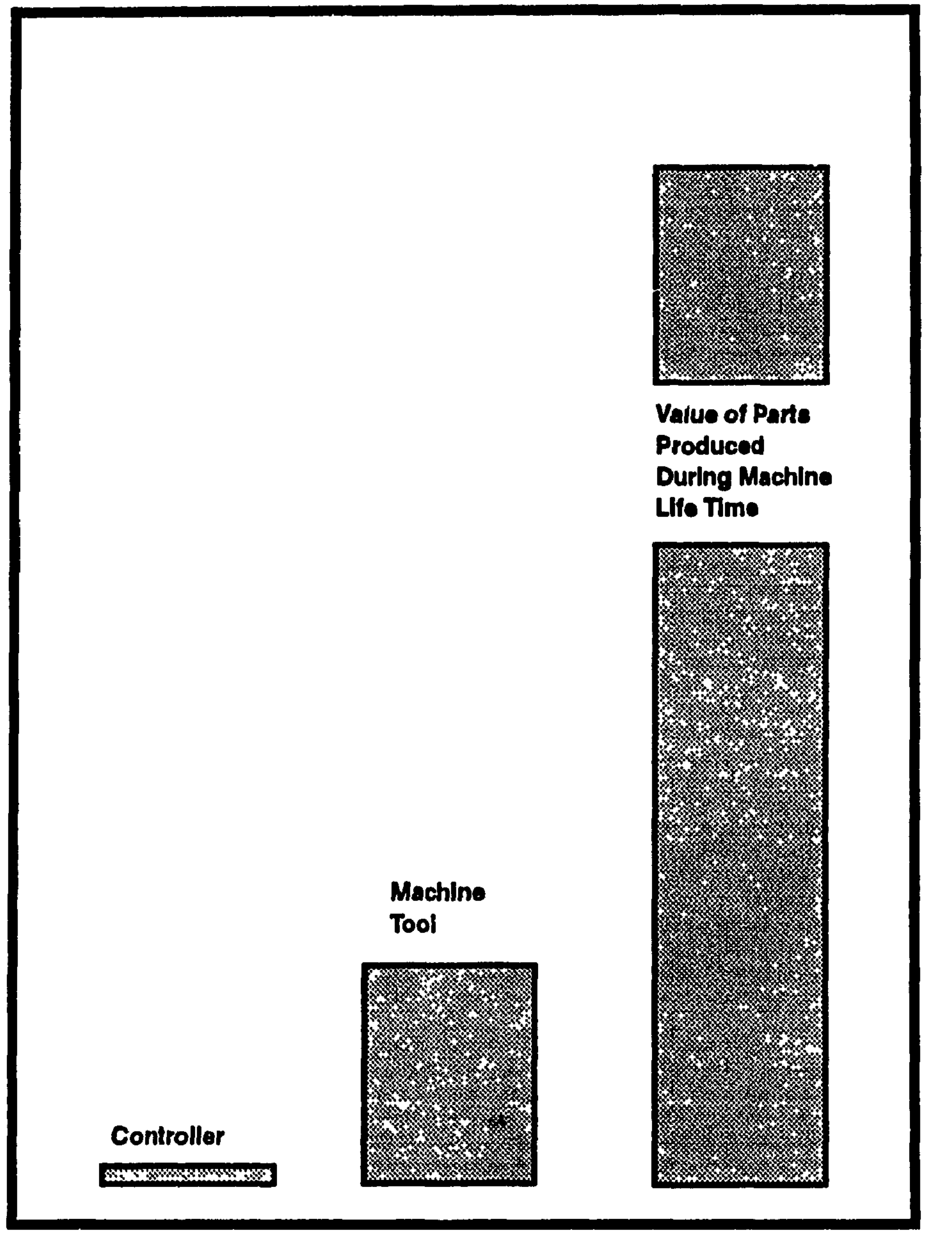


Figure 1.1: Benefits of automation[31].

During the entire process of deep hole drilling the tool is maintained in alignment with the bore axis initially by (a) a guide bush and (b) by the self guiding principle of the cutting tool [Cawdery, 1977]. Thus one of the most important requirements of deep hole machining is a machine which can provide an initial guidance (the guide bush) and subsequently required power to the tool or workpiece to maintain sufficient cutting and hence the correct balance of cutting forces to enable the tool to become self guided. Therefore, the process is independent of the guiding accuracy built into the machine unlike in turning process on a lathe [Cawdery, 1977].

The deep hole machining process has evolved continuously in terms of both the machine and the tools to meet the exacting requirements. This has resulted in development of special purpose deep hole drilling machines as illustrated in Figure 1.2 and special deep hole machining principles namely, Gundrilling, B.T.A drilling and Ejector drilling. These principles are illustrated in Figures 1.3a, 1.3b and 1.3c respectively.

1.2.1. THE MACHINE

The extreme nature of conditions imposed during the deep hole machining operation necessitate special considerations for building a machine and its structure to meet these conditions. Many of the conditions are unique to deep hole machining and are not

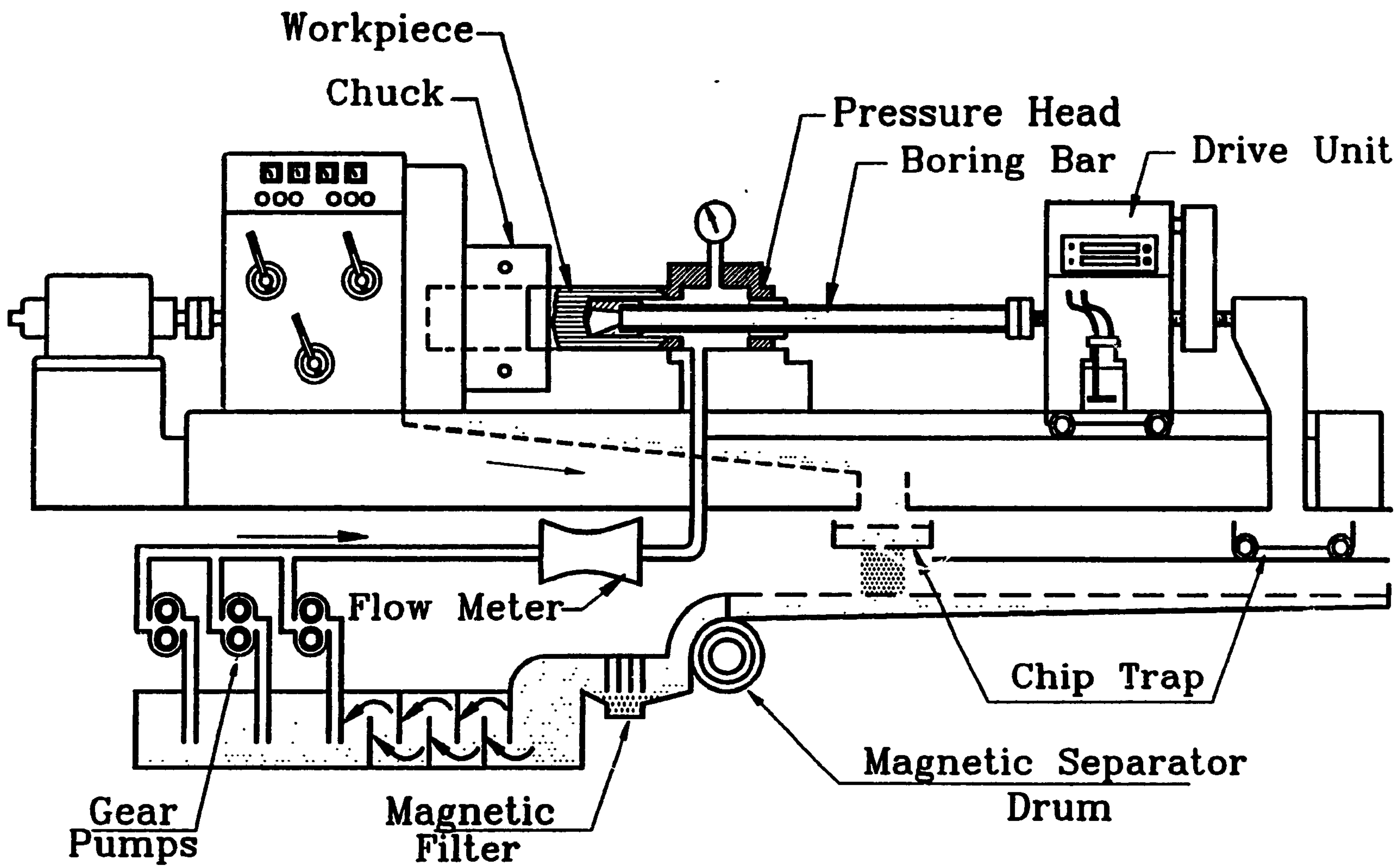


Figure 1.2 : Schematic illustration of a deep hole drilling machine[58].

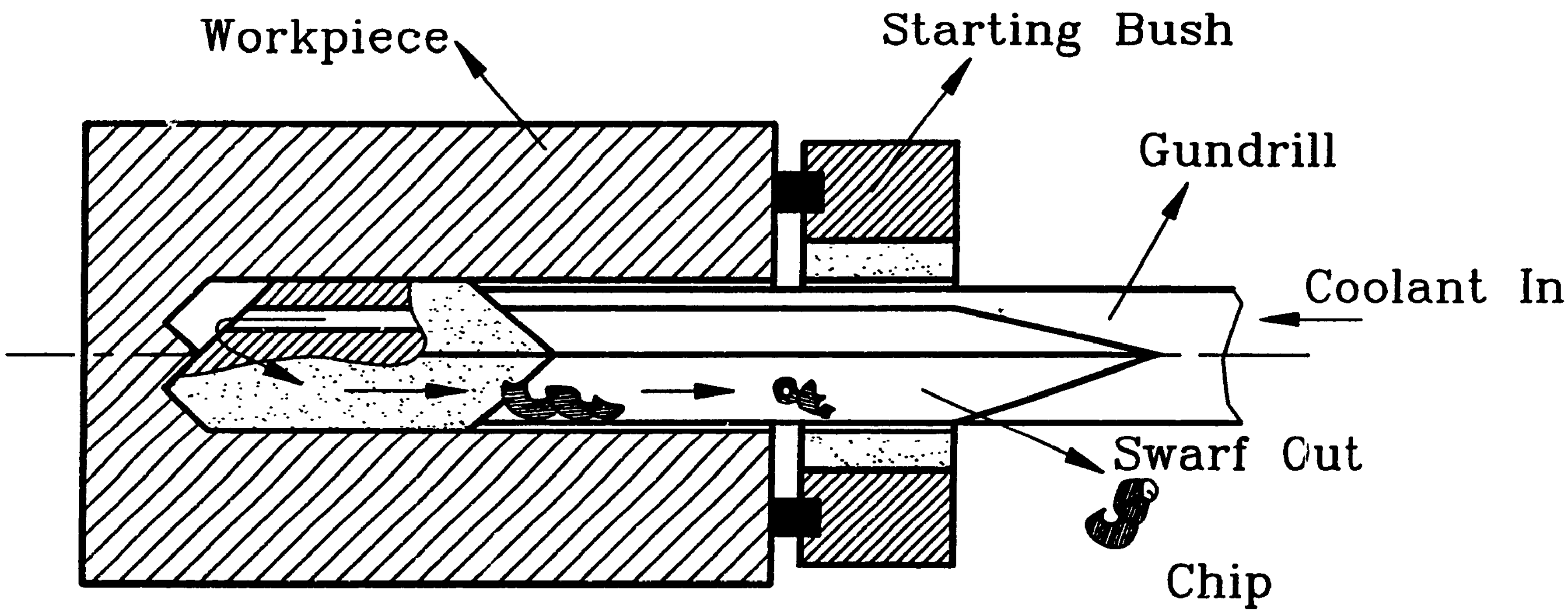


Figure 1.3(a): Principle of gundrilling

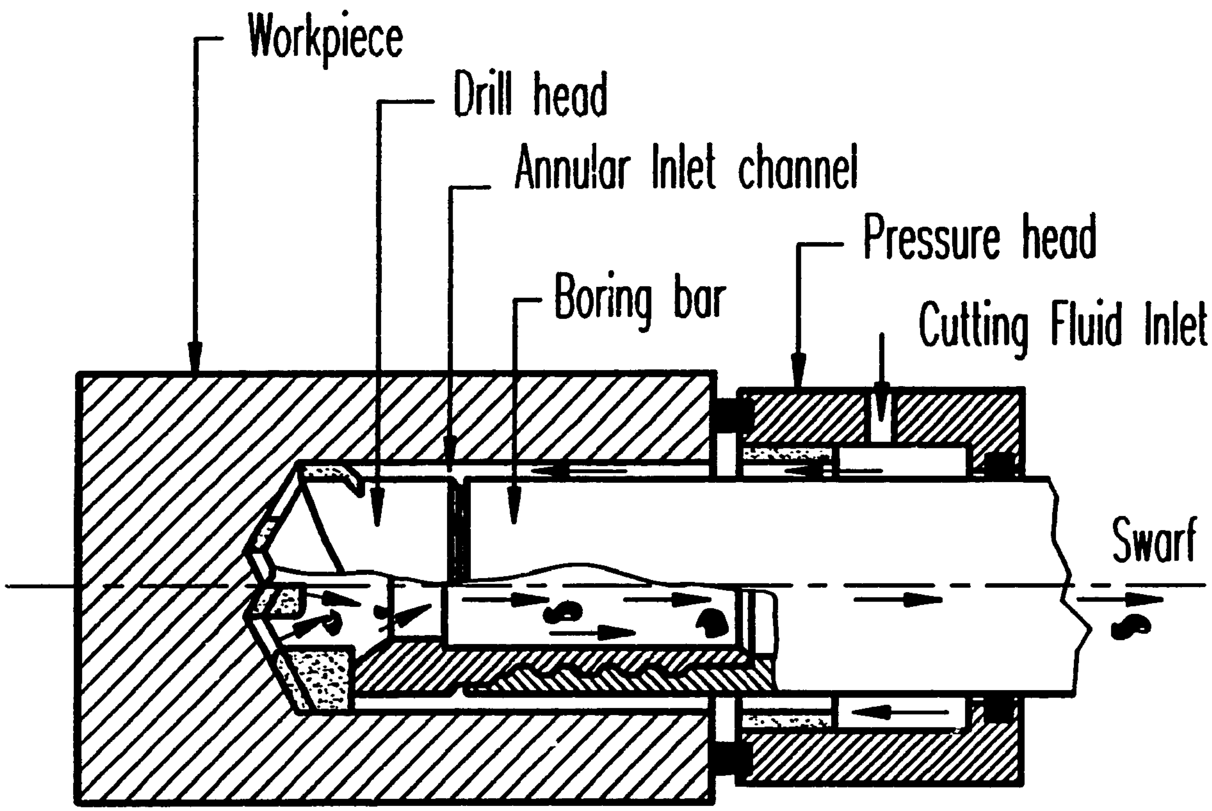


Figure 1.3(b): Principle of BTA drilling

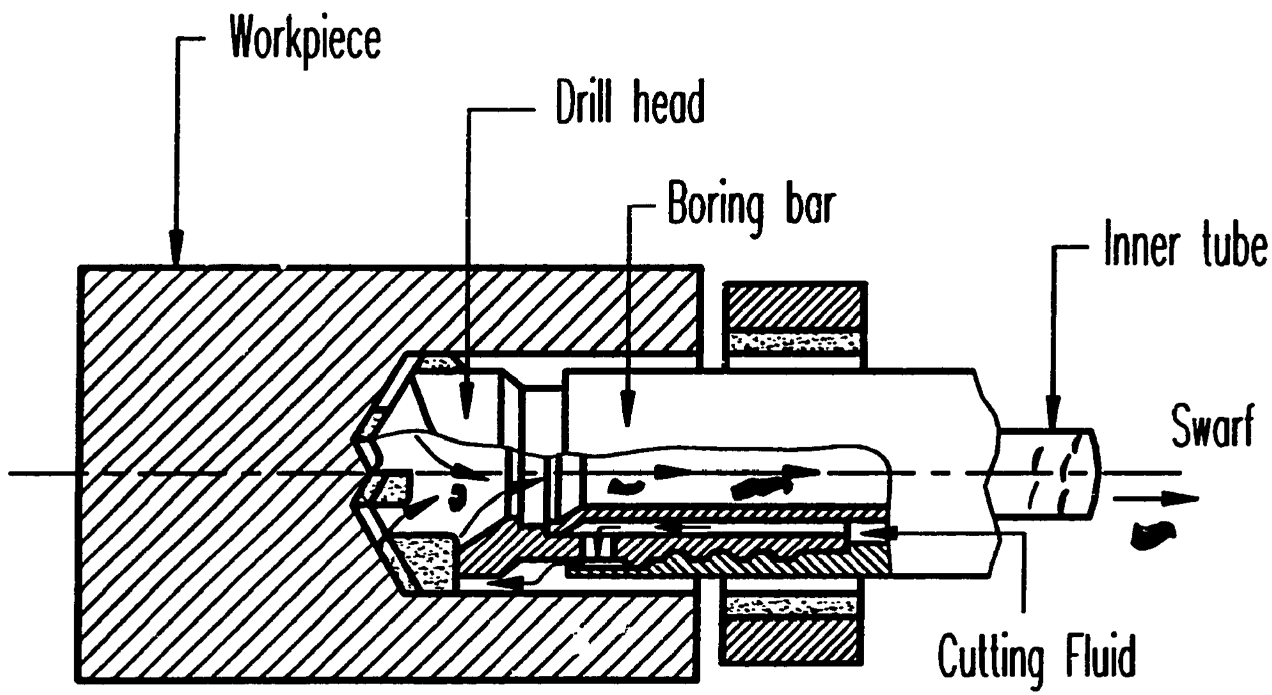


Figure 1.3(c): Principle of ejector drilling

common to other conventional machine tools. Therefore, without a special purpose deep hole drilling machine, the economic capabilities of the process cannot be fully exploited.

The deep hole drilling machine contain the following features to ensure economically satisfactory performance:

1. Spindle with ample power,
2. Feed drive unit,
3. Special coolant supply system,
4. Swarf filtration system,
5. Starting bush,
6. Rigid structure to withstand vibrations, and
7. Accessories.

The spindle and feed drive assembled over a rigid structure determines the capability of the machine. The design and function of these units are similar to such units of other machine tools. The next most important component of any deep hole machining system is the coolant supply system. The difficulty in supplying the coolant to the cutting zone and the need to remove the chip from the machining zone to the swarf filtration system over great lengths calls for specia' high pressure coolant supply system in deep hole machining. This special coolant supply and swarf removal system is unique to deep hole machining process and is hardly evident in any other machining processes. Coolant is pumped by heavy duty pumps, equipped with non return and pressure relief valves, through the pressure head to the point of cutting. Swarf returns along the bore of the boring bar to a chip basket where the bulk of the chips are retained. Partially cleaned coolant is then directed along ducting where the speed of the oil should not exceed two meters per second. Remaining particles will be deposited in the ducting. The coolant then passes over magnetic gravity flow filters or is dumped through pressure filters where the normal degree of filtration is 40 to 50 microns. The coolant tank contains approximately 10 times the maximum possible pump delivery per minute. The pressure head is mounted on the guide ways of the deep hole boring machine and is one of the most important components in a successful boring operation. It serves as coolant transfer unit and also supports the starting bush. The pressure head has three main functions

1. Transferring the cutting forces at the starting of the boring/drilling operation to the machine bed through the starting bush.

2. Transferring the coolant flow from stationary chamber into a rotating part of the boring bar.

3 Acts as a seal in preventing the coolant leakage, between stationary and rotating parts of the head.

1.2.2 B.T.A[1]. SYSTEM

The B.T.A method was developed in Germany between 1935 and 1945, using the fundamental principles of gundrilling technique [Griffiths,1977]. Figure 1.4. shows the B.T.A. System using a solid boring head, trepanning head and counter boring head. All these systems use the standard principle shown in Figure 1.2. Figure 1.5 shows a B.T.A solid boring head (a), counterboring head (b) and trepanning head (c). Solid boring heads are generally used from 6 - 60mm diameter but occasionally up to 120mm diameter. Trepanning heads are used from 45 - 500 mm diameter and counter boring heads up to 1000 mm diameter.

Another frequently used way of deep hole production is trepanning and sizing by counterboring, where a large diameter hole is required, it is not necessary to drill across the complete diameter; instead cutting takes place only over some 55-65% of the diameter, leaving a core of material. The advantages of trepanning as opposed to solid drilling are [Griffiths, 1977]:

1. Reduction in power requirements due to only cutting an annular ring.
2. Avoidance of a point zero cutting velocity.
3. Resulting core can be sued for other purposes.

---

[1]Boring and Trepanning Association

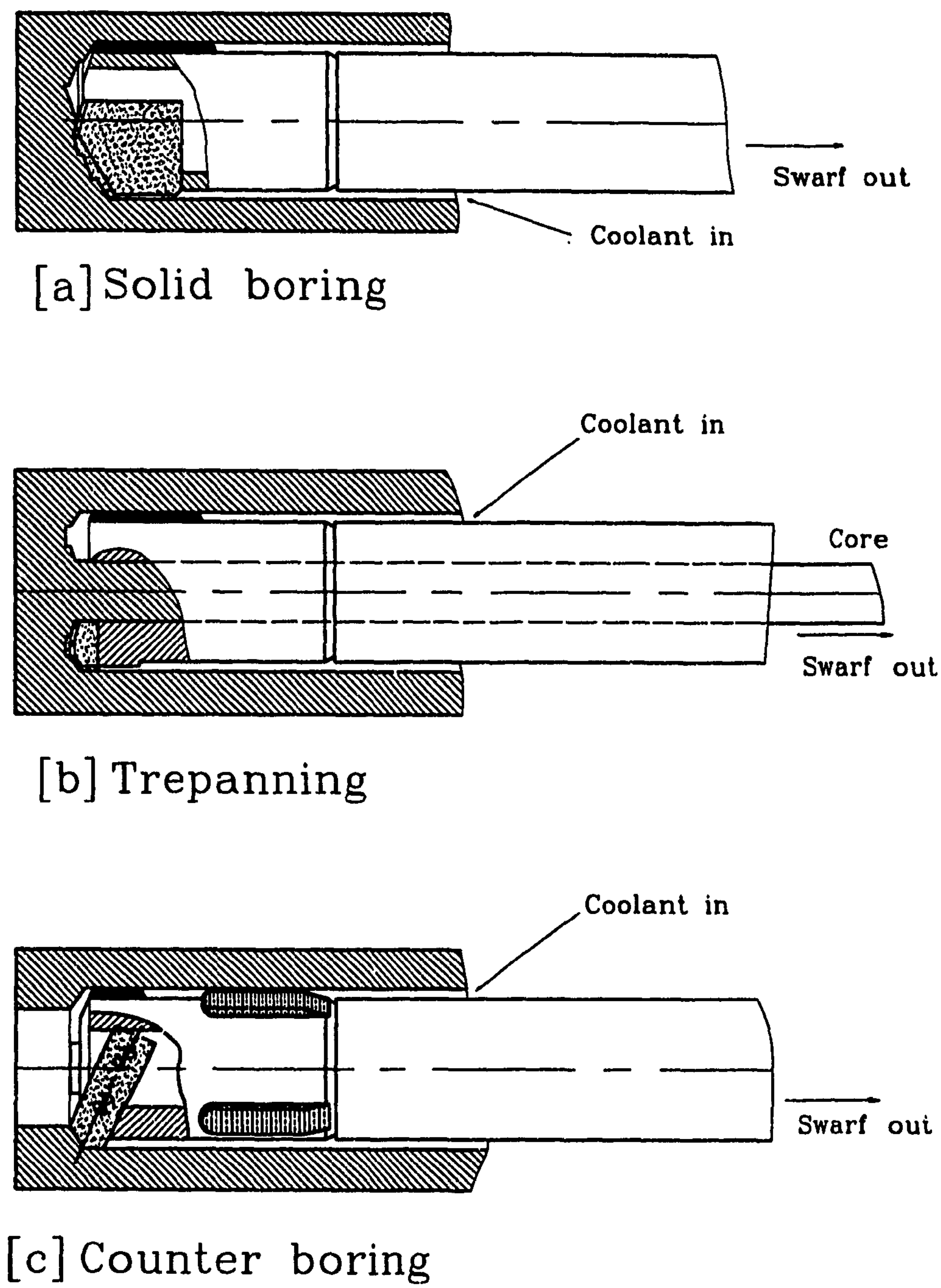
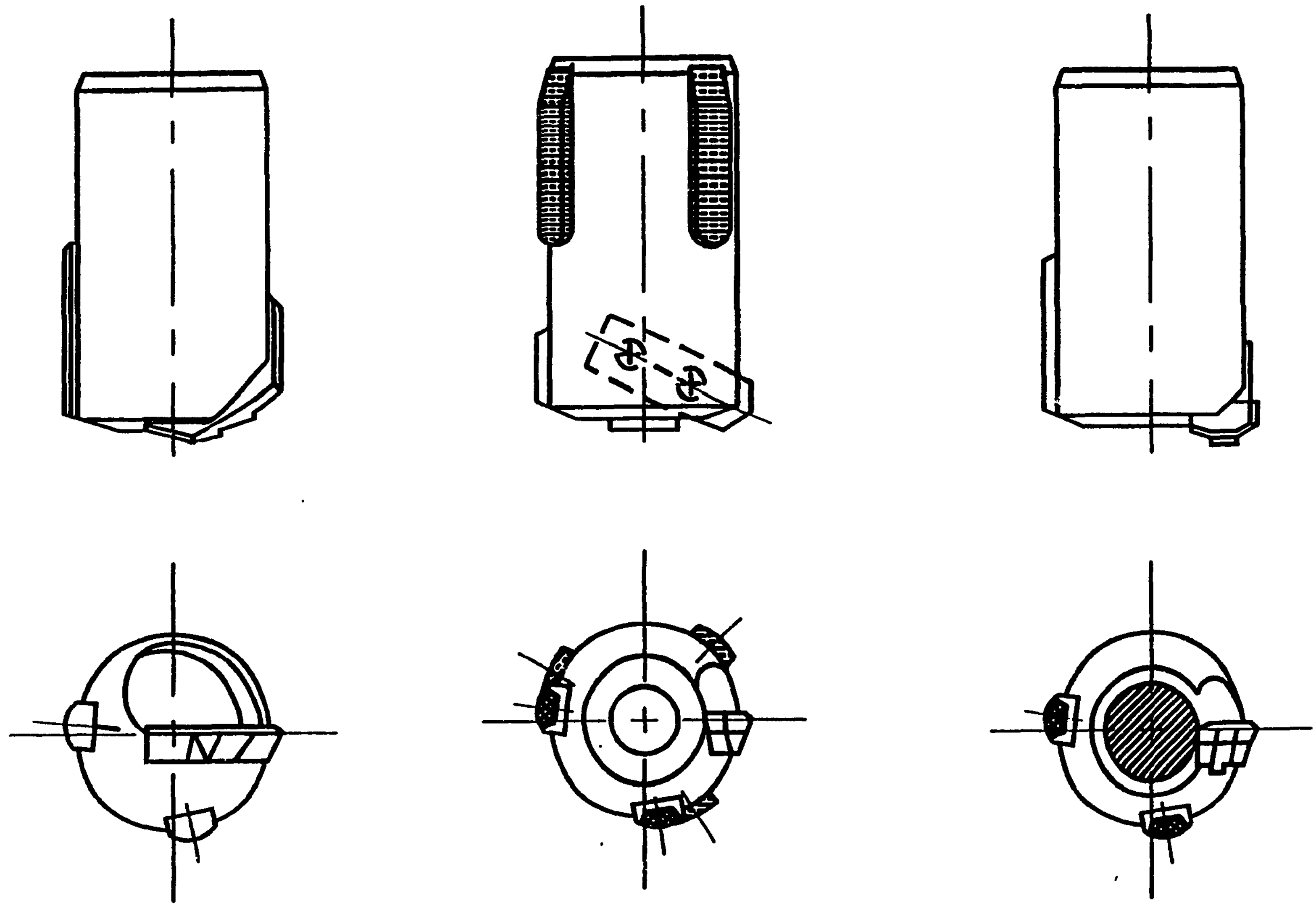


Figure 1.4: BTA drilling systems.



a) Solid boring drill b) Counter-boring drill c) Trepanning drill

Figure 1.5 : BTA type tools[59].

Counterboring is used for finish boring or opening out pre-drilled holes. Using the counterboring technique, it is possible to eradicate any small errors in concentricity or runout of an existing hole. Since counterboring tools are end cutting tools (as with solid boring tools) they can be used to produce stepped bores.

The surface finish of the bore achieved by the B.T.A. method is as follows [Greuner, 1977.]:

1.	Solid Boring	32 - 63	Micro inches (μ in) CLA.
2.	Trepanning	63 - 125	Micro inches (μ in) CLA.
3.	Counter Boring	16 - 125	Micro inches (μ in) CLA.

A bore tolerance of H9 (ISO), can be maintained through the life of the tool but if the component is counter bored this tolerance can be reduced to approximately H8. The runout of a 30 mm diameter solid boring head for example would be between 0.1 mm and 0.2 mm per metre. The improvement in the hole quality with respect to runout can be improved by rotating the workpiece while the tool is kept stationary or by counter rotating the workpiece and the tool. The ovality of a 50 mm diameter solid bored hole would generally be maintained within 0.025 mm [Sakuma et al 1980,1981.; Osman et al, 1974,1975].

1.3 NECESSITY OF AUTOMATIC CONTROL OF DEEP HOLE MACHINING PROCESS

Deep hole machining process has many advantages over conventional twist drilling process in addition to its capability to drill deeper holes. The cutting and burnishing actions of the deep hole drilling process produces better quality holes [Griffiths, 1977.; Osman et al, 1974,1975]. This has lead to adaptation of deep hole drilling methods in automotive engine production, aircraft engine production, hydraulic components manufacturing etc [Greuner, 1977; Saunders, 1977; Sturenburg, 1977]. It has been observed that [Sturenburg, 1977], in case of engine components such as pump barrels gundrilling gives a better bore than reaming and, it is both straighter and less tapered. A better component can thus be presented to the hone, which removes less material to produce the final hole. In addition to the examples mentioned here, the use of deep hole drilling technique is growing rapidly in major manufacturing industries [Greuner, 1977; Saunders, 1977; Sturenburg, 1977] which are candidates for factory automation. Recent developments in computer networks have enhanced full integration of DNC and CNC in a hierarchical manufacturing system with distributed control. Within such a system the individual machine tool should ideally be able to carry out the machining activities independently and effectively without human intervention at the machine hardware level. This is the primary goal of complete automation. In order to achieve this integration the deep hole machining process which has become a common manufacturing process should lend itself to automation.

Furthermore, the deep hole machining process is a highly complex operation with parameter control required at various stages of the process. The cutting mechanism of deep hole drilling is unique in the sense that the cutting speed varies along the radius of the tool and cutting is accompanied by burnishing process by the guide pads. At the beginning of the tool entry into the work piece, the machining is done at nearly zero cutting speed. The cutting speed increases to a maximum when the entire width of the tool is engaged in metal cutting operation. At the end of this phase, the guide pads enter the work piece and the chamfer of the guide pads causes additional thrust forces [Griffiths, 1982]. The thrust/axial force values reach their maximum at this phase. As the entry of guide pad progresses the burnishing torque and cutting torque together reach the maximum [Peter Streicher, 1975]. To avoid instantaneous increase of these machining responses to their maximums for a given cutting speed and feed, lower speed and feeds have to be initially selected. In order to fully utilize the economic capabilities of deep hole machining, higher cutting parameters have to be adopted after tool entry into the workpiece. This requires increasing the parameters at certain phases of machining. Moreover, the flow rate of the cutting fluid has an enormous effect on the quality of the hole produced [Syrett 1979; Zwingmann,1975] and different flow rates have to be applied for different operations. The chip shape and its size are an other important factor in determining the quality of the hole produced. Thus the deep hole machining process is highly labour oriented and needs extensive training and skill to operate economically. Also the actual machining operation is internal and not visible to the operator. This requires additional skills to have a feel for the machining condition. The general tendency of the machine operator is to stop the machining process whenever some abnormalities are suspected. This is highly uneconomical as the workpiece involved in the process is larger and the tool is expensive. The aim of the operator is to continue the normal machining operation by appropriate modification of the cutting or coolant flow parameters. Thus the reliability of the deep hole boring operation entirely depends on the reliability of the operator which is highly uneconomical and unacceptable to the modern industries. Constant human intervention at the machine hardware level in case of deep hole drilling operation is unsuitable for any production line even though the quality of the product might be superior. The main problems in deep hole machining systems can thus be summarised as follows:

1. Chip control
2. Variation of cutting forces during different phases of machining
3. Variation of torque during different phases of machining
4. Varying cutting speeds at different points of the tool makes it extremely difficult to estimate the tool wear.
5. Complexity of deep hole machining, i.e., metal cutting and burnishing, makes

it difficult to find an empirical relation between the quality of the hole produced and the cutting parameters.

6. Loss of work piece and tool when the tool fails is enormously costly.
7. Work stoppage for inspection is highly uneconomical and unacceptable in deep hole machining.

Thus, it is necessary to carry out the deep hole machining operation without any stoppage. The optimal use of the cutting tool and the material is essential for the process to justify the economic constraints. It is highly unlikely that a machine tool operator can provide all the knowledge required to successfully carry out the deep hole machining process, thus the need for automation of this process is further emphasized.

1.4 EXISTING CONTROL SYSTEMS AND THEIR APPLICABILITY TO DEEP HOLE MACHINING PROCESS

During the past decade there has been a steady increase in the number of Computer Numerically Controlled (CNC) machine tools. A common drawback of these systems is that their operating parameters, such as feed rates and cutting speeds are prescribed by part programmers and consequently depend on their experience and knowledge. To prevent tool breakage, part programmers tend to select conservative values for these operating parameters, thus reducing the production rate. Adaptive Control (AC) systems have been developed to address this problem. Traditional AC systems for machine tools are classified into two groups [Koren, 1983; Pressman et al 1977; Groover 1980; Wick 1977]:

(1) Those using adaptive control for optimization (ACO), maximize a performance index (economic function) subject to process and system constraints.

(2) Those using adaptive control with constraints (ACC) maximize machining parameters (viz., feedrate or cutting speed) subject to process and system constraints (eg. allowable cutting torque, force etc)

The difficulties in formulating realistic performance indices and in measuring required variables in a process environment, has limited the application of ACO systems to grinding process [Galip Ulsoy, 1983; Amitay, 1981]. Most of the systems used in practice for milling, turning and drilling operations are of the ACC type [Kegg,1978; Porter et al, 1969; Peklenik, 1972].

A conventional ACC system is shown in the Figure 1.6, for a CNC machine. The feed (f) is manipulated to maintain a required value (F_R) for the cutting force (F_C). The control loop is indicated in Figure 5(a). The cutting process is a part of the

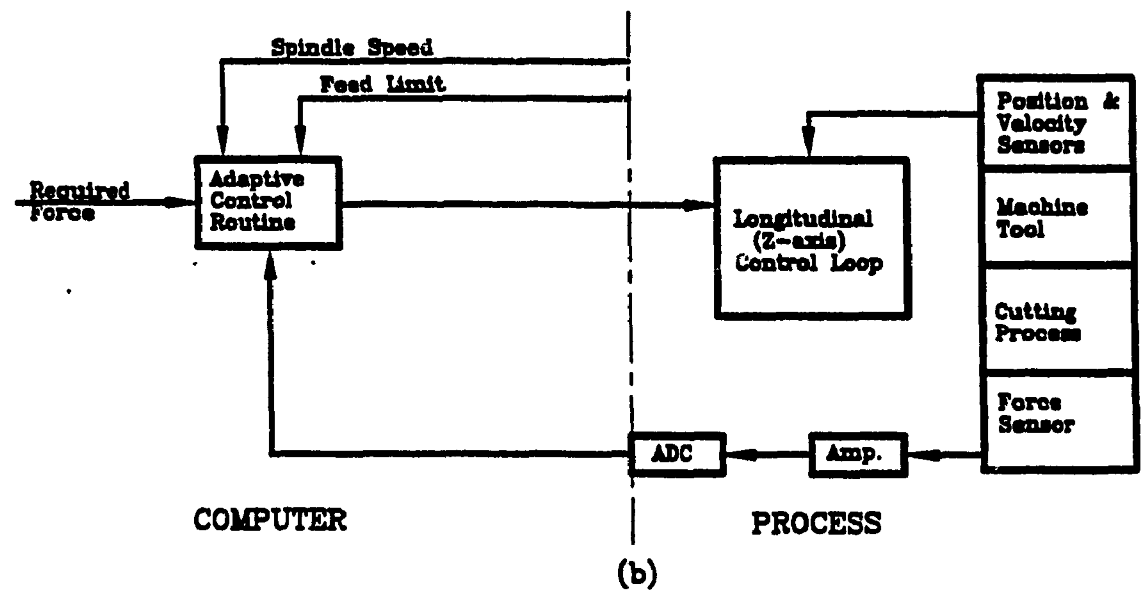


Figure 1.6 : A conventional ACC system[32].

control loop and the variations in the parameters of the cutting process affects the performance of the AC System. This type of system is termed 'adaptive' in the manufacturing literature, though it is not an adaptive system in the sense defined in the control literature [Groover, 1970; Mishkin et al., 1961; Landau 1979]. An adaptive system in the real sense, in addition to adapting the feed & speed to the cutting force, must also adapt the AC controller to the changing parameters of the cutting process. Such systems are referred to as "Parameter adaptive systems". A block diagram of a Parameter Adaptive System is shown in the Figure 1.7.

During the last few years many researchers have reported various schemes for implementing the parameter adaptive control systems for machine tools such as automatic gain control [Mathias 1977] and digital logic for gain control [Weck 1980]. The most recent work to date has been put forward by Masory and Koren [1980,1981] wherein the gain of the AC systems is based on the measured cutting force and manipulation of feed rate. In addition, the 'Adaptive Model Following Control' (AMFC) does not require explicit parameter estimation and uses a reference model to specify the desired close loop system characteristics. Structure of an AMFC is illustrated in Figure 1.8.

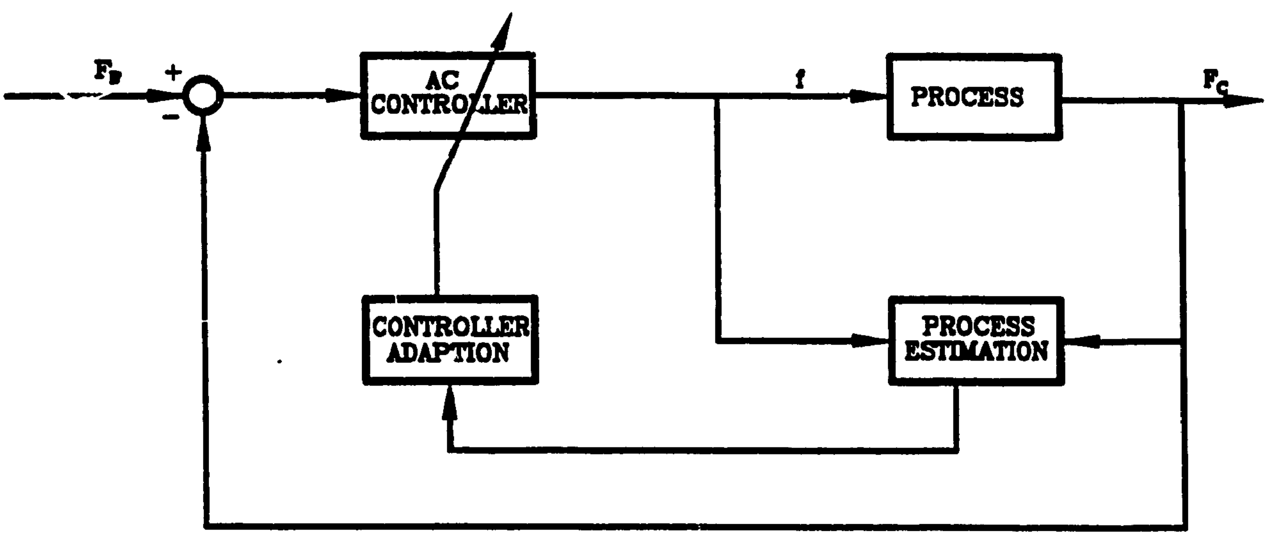


Figure 1.7 : Parameter adaptive system[32].

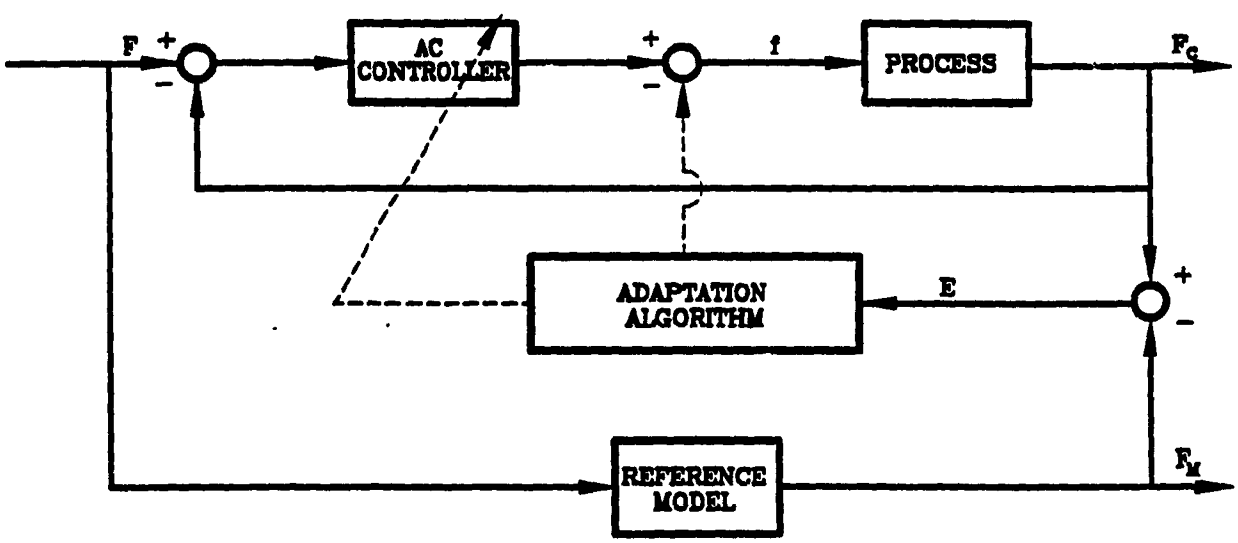


Figure 1.8 : Adaptive Model Following Control System[32].

1.4.1 ADAPTIVE CONTROL OF DEEP HOLE DRILLING MACHINE

L.S. Lukic [1987] has applied the ACC method for adaptive control of deep hole drilling machine. This method incorporated the chip size factor along with the torque and axial force for determining the machining condition. The amplified signals from a special chip sensor was analyzed with an FFT analyzer. The power spectrum shows a remarkable correlation with the size of the chips. The results are shown in Figure 1.9. The control setup and the control loop are schematically illustrated in Figure 1.10 and Figure 1.11 respectively.

This method even though for the first time has taken the chip size as a determining factor, has inherited all the disadvantages of the ACC system. Heavy reliability on the mathematical description of the process and unavailability of suitable mathematical models to describe the process makes the system unsuitable for a production environment.

Another important contribution to the adaptive control of deep hole machining operation is made by Bernhard Buse[1987]. The system works on the following principles:

- Automatic detection of chatter vibrations with the help of sensors in deep-hole machining process.

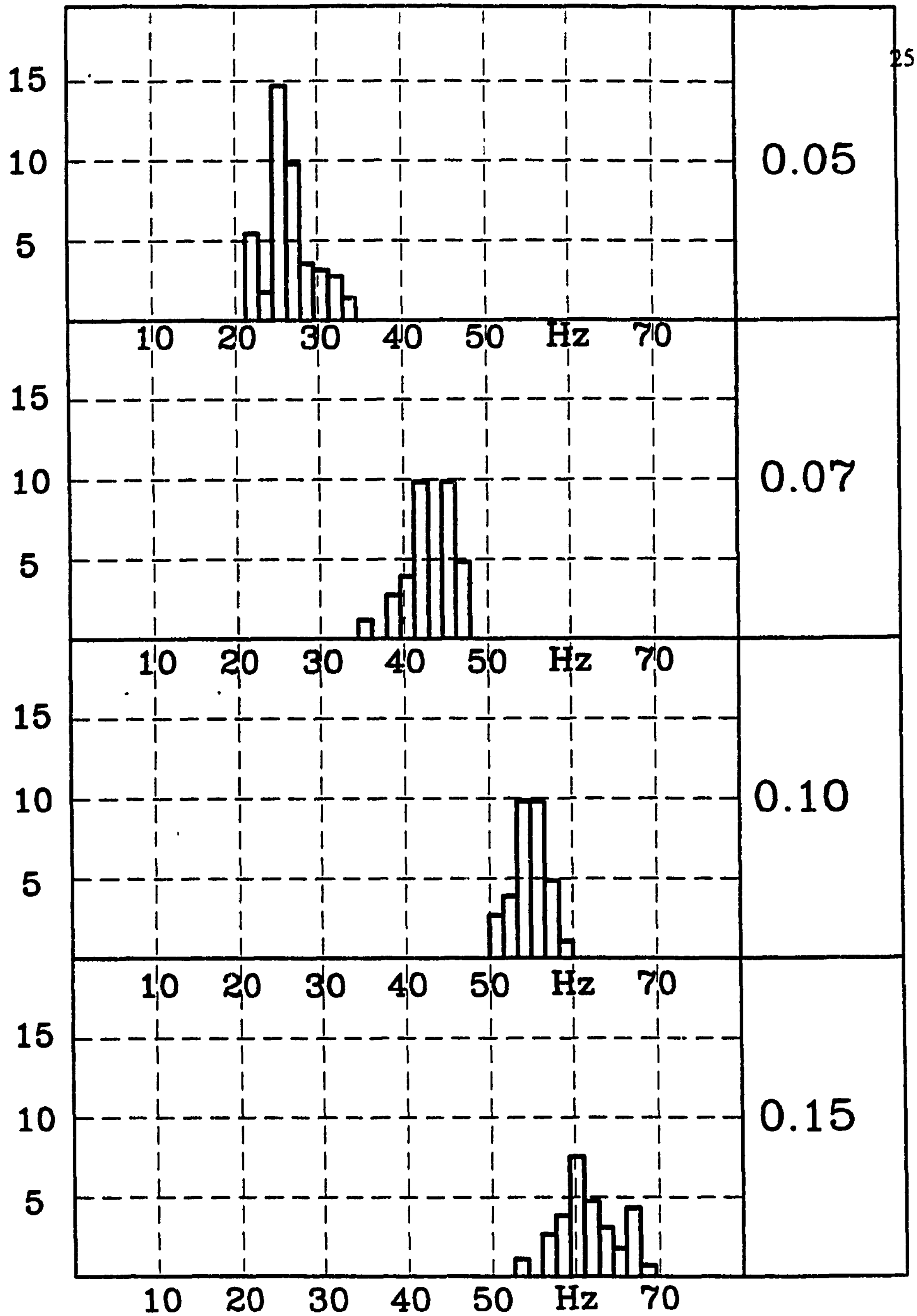


Figure 1.9 : FFT diagram for chip size[51].

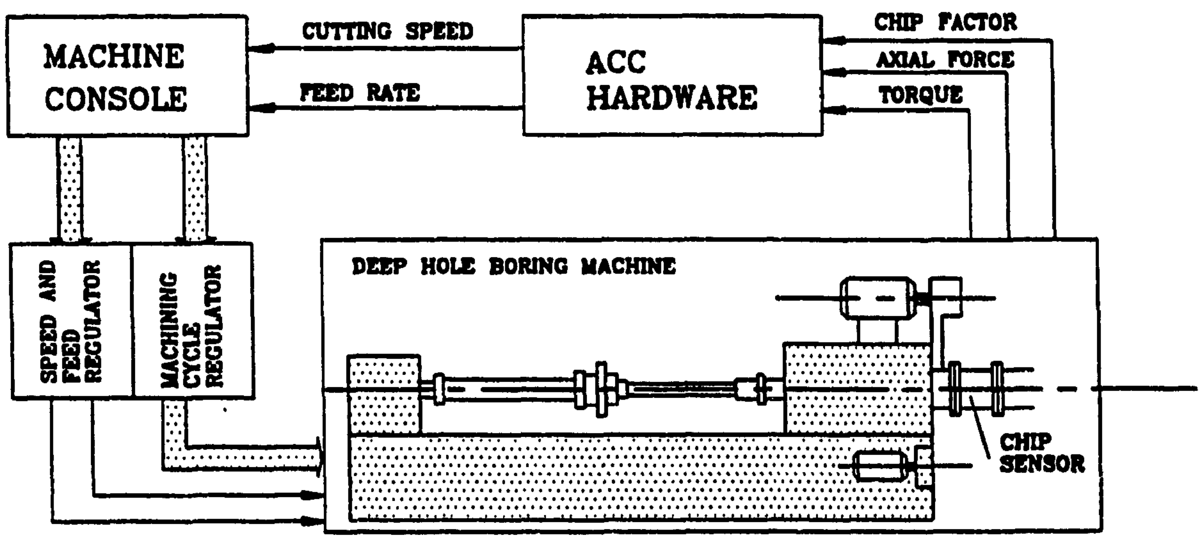


Figure 1.10: Adaptive control setup for deep hole drilling[51].

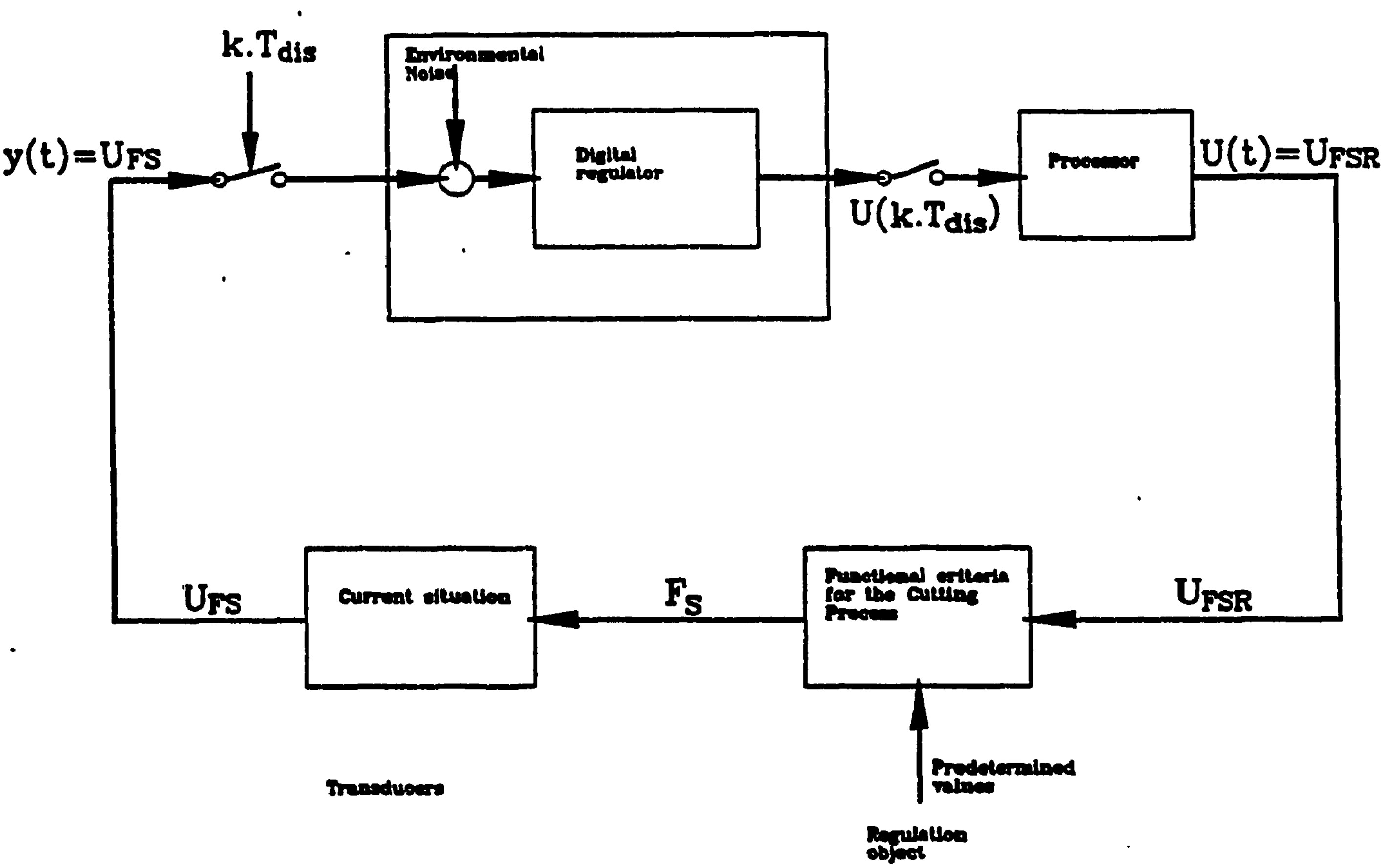


Figure 1.11: Adaptive control loop diagram for deep hole drilling[51].

- Controlling damping effect (here for a hydraulic Lnachester damper, an electro-hydraulic circuit has been developed).

- Controlling damper position on the boring bar.

The main draw back of the system is that the researcher has concentrated only on controlling the chatter during machining and the system is incapable of adopting the cutting parameters or the coolant flow parameters for increasing the productivity. The dearth of mathematical knowledge relating the vibration levels to the various states of the tool work interaction makes it much harder to apply this system in the real life situation.

Chandrashekhar[1984] had proposed an adaptive control system for deep hole machining based Random Function Excursion [Sankar et al.,1975; Osman et al.1975]. The following parameters were defined to be included in the system:

$$MCE = \sum_{i=1}^{n_e} \lambda_i P_c(\lambda_i) \qquad (1.1)$$

$$\text{RMSCE} = \left[\sum_{i=1}^{n_e} (\lambda_i - \text{MCE}) P_c(\lambda_i)\right]^{1/2} \tag{1.2}$$

$$\text{MVE} = \sum_{i=1}^{n_e} \Lambda_i P_V(\Lambda_i) \tag{1.3}$$

$$\text{RMSVE} = \left[\sum_{i=1}^{n_e} (\Lambda_i - \text{MVE})^2 P_v(\Lambda_i)\right]^{1/2} \tag{1.4}$$

These four parameters are evaluated at a specified level namely the CLA value of the roughness profile together with the height indices, CLA and RMS values to obtain a good knowledge of the machined surface.

The signals representing the resultant force system are represented by corresponding stochastic excursions about the mean value of the respective components. These four parameters are:

$$\text{MCEGF} = \sum_{i=1}^{m_e} \lambda_{if} P_c(\lambda_{if}) \tag{1.5}$$

$$RMSCEGF = \left[\sum_{i=1}^{m_e} (\lambda_{if} - MCEGF)^2 P_c(\lambda_{if})\right]^{1/2} \tag{1.5}$$

$$MVEGF = \sum_{i=1}^{m_e} \Lambda_{if} P_v(\Lambda_{if}) \tag{1.7}$$

$$RMSVEGF = \left[\sum_{i=1}^{m_e} (\Lambda_{if} - MVEGF)^2 P_v(\Lambda_{if})\right]^{1/2} \tag{1.8}$$

In addition, two other parameter for the complete force representation are calculated as:

$$CLAGF = \sum_{i=1}^{n} \frac{y_i}{N} \tag{1.9}$$

$$RMSGF = \left[\frac{1}{N}\sum_{i=1}^{n} y_i^2\right]^{1/2} \tag{1.10}$$

From the above set of parameters six parameters describing the surface texture are evaluated using the following relations:

CLA $= F_{1i}$ **[CLAGF, feed, speed]**

RMS $= F_{2i}$ **[RMSGF, feed, speed]**

MCE $= F_{3i}$ **[MCEGF, feed, speed]**

RMSCE $= F_{4i}$ **[RMSCEGF, feed, speed]**

MVE $= F_{5i}$ **[MVEGF, feed, speed]**

RMSVE $=F_{6i}$ **[RMSVEGF, feed, speed]**

where 'i' is the number of generalized coordinates and functions F_{1i} to F_{6i} vary depending upon the finishing operation. The computations are performed within the microprocessor. The values computed in equations above are then compared with the 'acceptable' standard and depending upon the output, the correction in feed and speed are computed using the controller subroutine in the microprocessor. A schematic diagram of the entire system is shown in Figure 1.12.

The above system is hypothetical and has failed to address the various factors which strongly influence the quality of the deep hole drilling operation such as coolant flow rate cutting torque etc., Moreover the optimization specified by the author along with various calculations required to estimate the feed and speed are beyond the scope of the available time for decision making in real-time. No simulation or experimental results validating the proposal has been presented and hence the system is hardly applicable for real-time adaptive control of deep hole machining process.

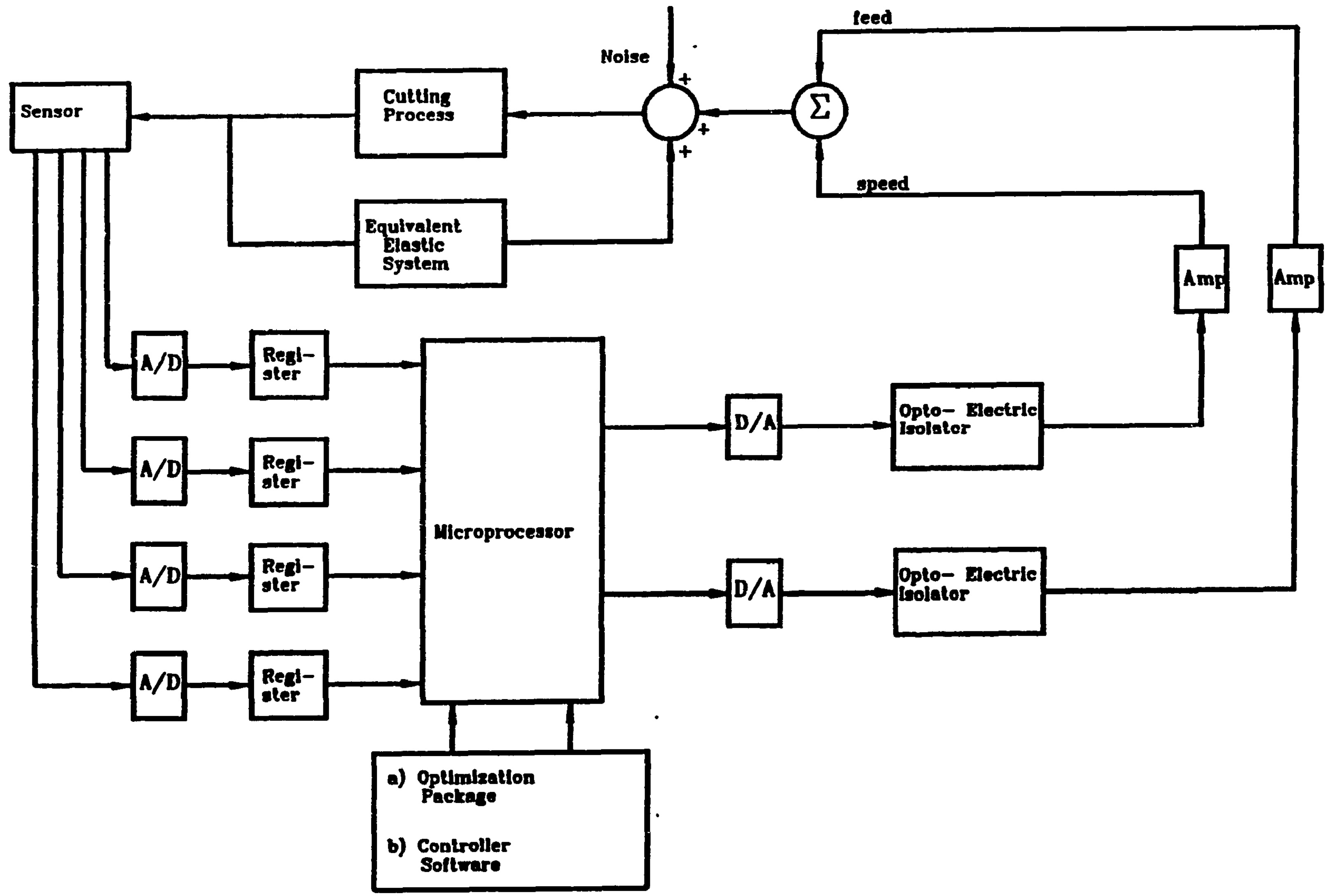


Figure 1.12: Adaptive control using random function excursion[17].

1.5 INTELLIGENT MACHINING SYSTEM

Advances in the field of artificial intelligence has lead to the development of intelligent machining systems. Such machining systems can be grouped into machining systems using:

a. Fuzzy logic systems, and

a. Expert system or Knowledge based systems.

1.5.1. FUZZY SYSTEMS

Fuzzy logic was developed by L.A. Zadeh in 1965. Since then it has been successfully applied to solve wide variety of problems in many fields such as information processing, decision making, medical diagnosis, pattern recognition etc [Zhu et al, 1982]. Research on the application of fuzzy logic for process control has been proposed by many researchers [Gupta et al., 1978; MacVicar et al 1976] and the application into machining process has been very few[Sandrasegaram 1981; Zhu 1981,1982]. The results of these investigation have shown that the fuzzy theory has many advantages for its application to manufacturing process control [Zhu 1982]. These advantages are:

* It can be used to control uncertain processes including the very complicated ones which may have the property like multiple input-output, non-linearity, time delay, dead band etc.

* Fuzzy controller can simulate human experience and experimental information for controlling processes.

* It is very convenient for automatic computer control. When using fuzzy controllers, the amount of computation task and time can be enormously reduced as compared to classical or modern control theory. Hence, it is possible for real-time control of processes with a small minicomputer or even micro-computer and great savings in hardware cost can also be achieved.

* The fuzzy controller can be used for controlling to accuracies of up to 5%, which is suitable for many industrial process control applications.

These advantages makes fuzzy controller an ideal candidate for consideration in automation of deep hole machining process. But, These conclusions are drawn after theoretical studies and real application of such a system has not be demonstrated to back the claims. Even in theoretical sense, the suitability of fuzzy controller is overemphasized. Fuzzy logic is based on classification of uncertain data into different classes by evaluation of class membership. For example, partitioning of the set

$$X = \{x_1, x_2, \ldots\ldots x_n\} \tag{1.11}$$

into fuzzy subsets and let μ_{si} be the indicator function so that

$$\mu_{si}: \Rightarrow X \rightarrow [0..1] \tag{1.12}$$

Assignment of the pattern to different clusters can be visualized in terms of partition matrix wherein:

$$\mu_{ik} := \mu_{si}(x_k) \tag{1.13}$$

μ_{ik} denotes the degree of membership of x_k to the subset si. The most important aspect of application of fuzzy logic is the determination of μ_{ik}. Following the work of Bezdek [1976, 1980], the fuzzy partition is obtained by:

$$\min\ Z_m(u,v) = \sum_{i=1}^{N_c} \sum_{k=1}^{n} (\mu_{ik}) \|x_k - v_i\|^2 \tag{1.14}$$

Differentiating the variance with respect to v_i(fixed u) and μ_{ik}(fixed v) and normalizing:

$$v_i = \left[\frac{1}{\sum_{k=1}^{n} (\mu_{ik})^m}\right] \sum_{k=1}^{n} (\mu_{ik})^m x_k \tag{1.15}$$

$i = 1,\ldots\ldots,c$

and

$$\mu_{ik} = \frac{\left(\frac{1}{\|x_k - v_i\|^2}\right)^{\frac{1}{m-1}}}{\sum_{j=1}^{N_c} \left(\frac{1}{\|x_k - v_j\|^2}\right)^{\frac{1}{m-1}}} \tag{1.16}$$

$i = 1,2,\ldots\ldots\ldots,N_C$; $k = 1,\ldots\ldots n$

where N_c is the number of cluster centres, m is the exponential weight and v_i is the cluster centre.

The system of equations (1.15 and 1.16) cannot be solved analytically but approximate solutions can be obtained using iterative procedures. The time required to find the solution iteratively make it hardly useful for real-time application where high speed decisions during emergencies are involved.

1.5.2. EXPERT/KNOWLEDGE BASED SYSTEMS

The most recent works to date in machining automation are, the Hierarchical, Knowledge based control in turning [Teltz et al, 1993] and the Intelligent Machining System (IMS) by G. Chryssolouris and M. Domroese [1988]. In the intellignent machining system the machining control is treated as a decision making problem. The decision making approach provides a framework in which artificial intelligence techniques can be effectively utilized. The system (IMS) uses a multiple-sensor approach for processes monitoring. A schematic illustration of the system is given in Figure 1.13. In this work, a concept was developed for a controller that optimizes a machining process with respect to criteria which are complex nonlinear functions of the input parameters. The decision making approach used by the IMS can be most concisely expressed in a decision matrix format where the matrix elements indicate the consequence of each feasible alternative with respect to a set of optimization criteria. Three sensors are utilized for process monitoring: force, temperature, and acoustic emission. These sensors feed their signals into three independent process models. Each model provides an independent estimate of the current flank and crater wear as well as the wear rates that would be profiled by each alterative. In general, different sets of alternative machining inputs such as feed rates and cutting velocities will provide different flank and crater wear rates. A sensor integration module determines the final estimates for each wear and wear rate based on the corresponding model

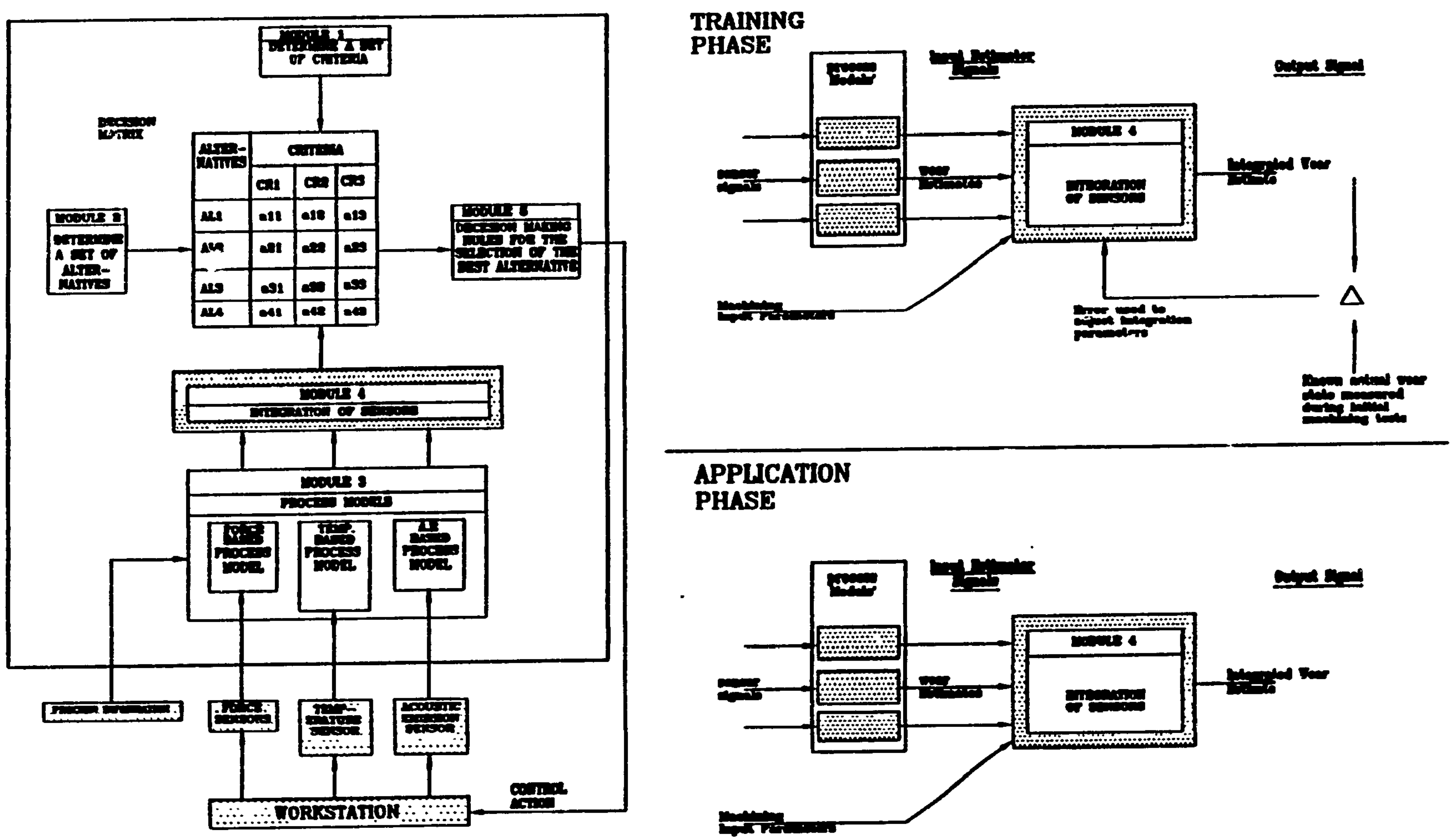


Figure 1.13: Intelligent machining system[18].

estimates. The expert system is embedded in the control loop. The expert system contains the decision making rules for the selection of the best alternative. The sensor integration is done by a neural network system in the control loop. The expert system forms a minor part of the control system and participates only the in the evaluation of the decision criterion. The criterion is based on the sample mean and variance in multiple regression of the acquired data. It is the common knowledge that the force signal can hardly be relied on as an indicator of the tool condition as the cutting force, instead of increasing abruptly at the tool failure, may decrease if the failure that it forms a positive rake on the cutting edge. Also, The cutting force increases with increase in flank wear while decreases with increase in crater wear. If there is a combination of both flank wear and crater wear the cutting force may hardly change due to the opposing effects of flank and crater wear. Thus this system, even though novel in its approach, has inherited the disadvantages of machine-tool-workpiece specificity and excessive computation time. The optimization of cutting parameters consume considerable computing time and are not suitable for real time control of machining process. Moreover, the system is deficient in its decision making criteria. For example, a current situation reflected by the sensors may be due to the situation which had already occurred or the situation which may be entirely new and unexpected. In such situations weighing the previous situation and action along with the current unexpected situation should lead to a decision other than stoping the machine. The stoppage of the machine is uneconomical and hence should only be adopted during an emergency. Also, IMS does not take into consideration the environmental situations of machining namely, the coolant failure, inclusions or air gaps in the workpiece etc. The Acoustic emission sensor adopted for detecting tool failure is unreliable as these sensors are sensitive to chip breaking. Hence an acoustic signal can indicate either tool breakage or chip breakage. IMS is novel in approach and can be modestly applied to simple single point turning operations by incorporating additional facilities in the system but is entirely beyond the scope of complex machining process like the deep hole machining process. An entirely new approach to machining process control for deep hole machining should be sought. The new approach while incorporating the advantages of the proposed systems should overcome the disadvantages which have plagued all these systems and should also be a stand alone system with generalized character rules in the machining rule base. Incorporation of a static rule base to include the environmental situations should also be sought especially, as the environmental factors like the coolant flow rate, chip size etc, have a direct bearing on the effectiveness of the machining process.

Even though the above systems were novel in their approach they all have a common draw-back. The decision making criterion is based on only one parameter. It is highly unlikely that a single parameter can be used for controlling the deep hole machining. For a multi parameter control, a better understanding of the relation between various parameters of machining and the machining response is necessary.

A complete set of mathematical models defining these relations forms the core of such a system. Non availability of such mathematical relations with multiparameters makes it harder to adopt these control schemes. The only way out seems to be the expert systems used for decision making based on multiple parameter criterion.

An expert system for deep hole machining was proposed by Astakhov [1987]. This model of expert control was based on a symbolic modelling of deep hole machining and control based on axial force, torque and swarf flow rate. The decision process is accomplished by boolean algebra. The model of this expert system was built on a LISP machine for testing under simulated conditions. The main disadvantage is that this system is not utilized in real time control but for off-line expert advice.

1.6 EXPERT SYSTEMS AS CONTROLLERS

It was first observed that the actual implementation of control laws often incorporates a substantial levels of heuristics logic. This is true for simple regulators as well as for more sophisticated multivariable control loops. Expert system methodologies provide a systematic approach for dealing with heuristic logic by being a part of the primary feedback loop [Astrom et al. 1986].

Building of a control system can be classified into modelling, identification, analysis, simulation, control law design and implementation. The heuristic as parts of a control design had remained hidden in practical designs and rarely discussed in the control theory literature [Astrom 1986]. It can be seen that such mundane problems as PID control can be explained by heuristics in addition to the theory. In multivarible and self-tuning regulators, heuristics are even more important. This is referred to by different names such as supervision safety nets or safety jackets [Clark 1981, Isermann 1982]. Once one accepts that control algorithms will contain heuristics one may also ask what a more intensive use of heuristics may contribute to control systems.

There have recently been proposals for other types of tuning algorithms that have a wide range of operability [Astrom and Hagglund, 1983, 1984]. It seems appealing to explore the possibility of designing systems that combines a range of algorithms. This entails orchestration of different algorithms to achieve varying control objectives. The objective of such a control, called expert control is to encode knowledge representation and decision capabilities to allow intelligent decisions and recommendations automatically rather than to preprogram logic which treats each case explicitly. Important analogies are found in expert systems. Hence it is desirable to explore the possibility of applying expert systems as an aide in solving the complex control problems.

1.7 EXPERT SYSTEM

An expert system is a specialized computer program that exhibits the same level of problem solving skills as an expert for a narrow problem domain. It embodies knowledge and reasoning capabilities that allow it to draw quality conclusions comparable to those drawn by a human expert [Parsaye et al,1988; Adeli, 1990]. Structure of an expert system is illustrated in Figure 1.14.

Experts, generally defined, are people who are very good at solving specific types of problems. Their skill usually comes from extensive experience, and detailed specialized knowledge of the problems they handle.

An expert system seeks to capture enough of the human specialist's knowledge so that it too can solve problems expertly [Adeli, 1990].

1.7.1 KNOWLEDGE

Knowledge consists of symbolic descriptions that characterize the definitional and empirica. relationships in a domain, and the procedures for manipulating those descriptions. Relations are expressed as dependencies and association among objects. Typically, these relations describe taxonomic, definitional and empirical associations.

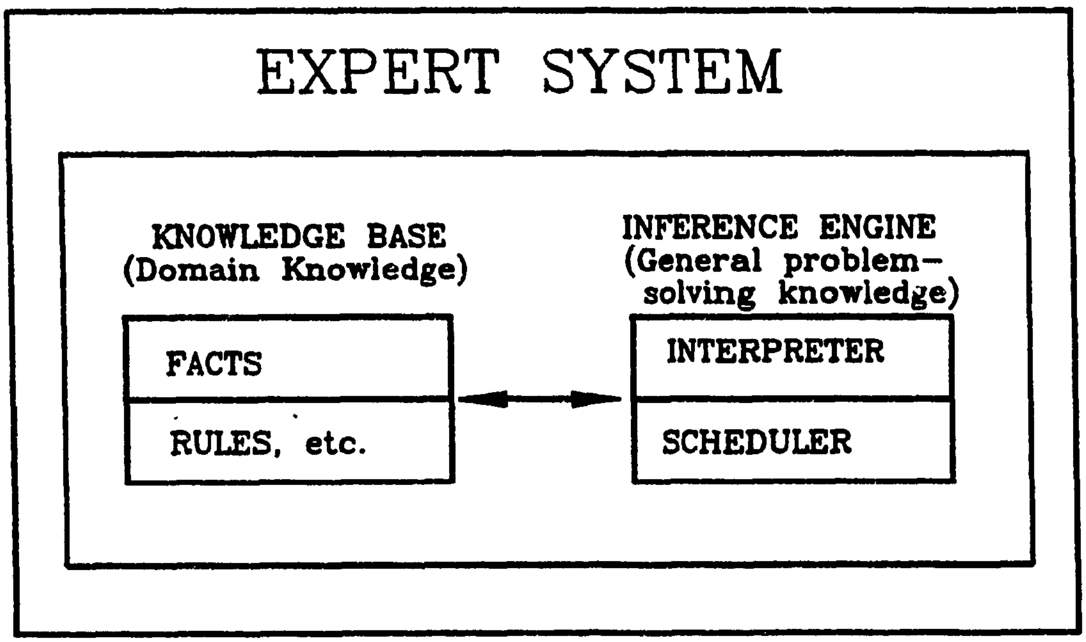


Figure 1.14: Structure of an expert System[74].

Procedures specify the operations to be performed when attempting to reason or to solve a problem.

Knowledge is more than a static encoding of facts. It includes the ability to use those facts when interacting with the world. A basic premise of AI is that knowledge is the ability to form a model that accurately represent the object as well as the actions that can be performed by it and on it. Models then are just abstractions of reality.

1.7.2 KNOWLEDGE REPRESENTATION

Knowledge representation is a term used in AI to describe how knowledge is structured in a program. A representation is a set of syntactic and semantic conventions that make it possible to describe objects, relations and/or procedures. Knowledge representation also includes a set of operations for manipulating the described things. Representation seems to be the key to creating computer intelligence.

There is a set of standard knowledge representation techniques, each having its own strengths and benefits. The benefits of each can be delineated along the lines of efficiency, ease of use, ease of understanding, flexibility and robustness. The three most popular formalisms in use today are:

- Production rules,
- Frames, and
- Semantic nets.

Figure 1.15 illustrates different classes of knowledge representation.

A) PRODUCTION RULES

Production rules are a natural way of expressing heuristics because they utilize the simple **IF** *condition* **THEN** *action* format. These two parts of a production rule represent an antecedent -some pattern containing several clauses linked by logical connectives and a consequent- that specifies an action. A production rule is fired when the current situation matches the condition of the IF part. Successive rule firing produce an inference chain.

Production rules can represent both declarative and procedural knowledge. Although the appearance of the production rules are similar in style to the IF statements of conventional programming languages, there is a major difference in how they are handled. The rules in a rule-based system are not executed in a sequential or predetermined order. In addition, the flow of control is not limited to branch only at selected points.

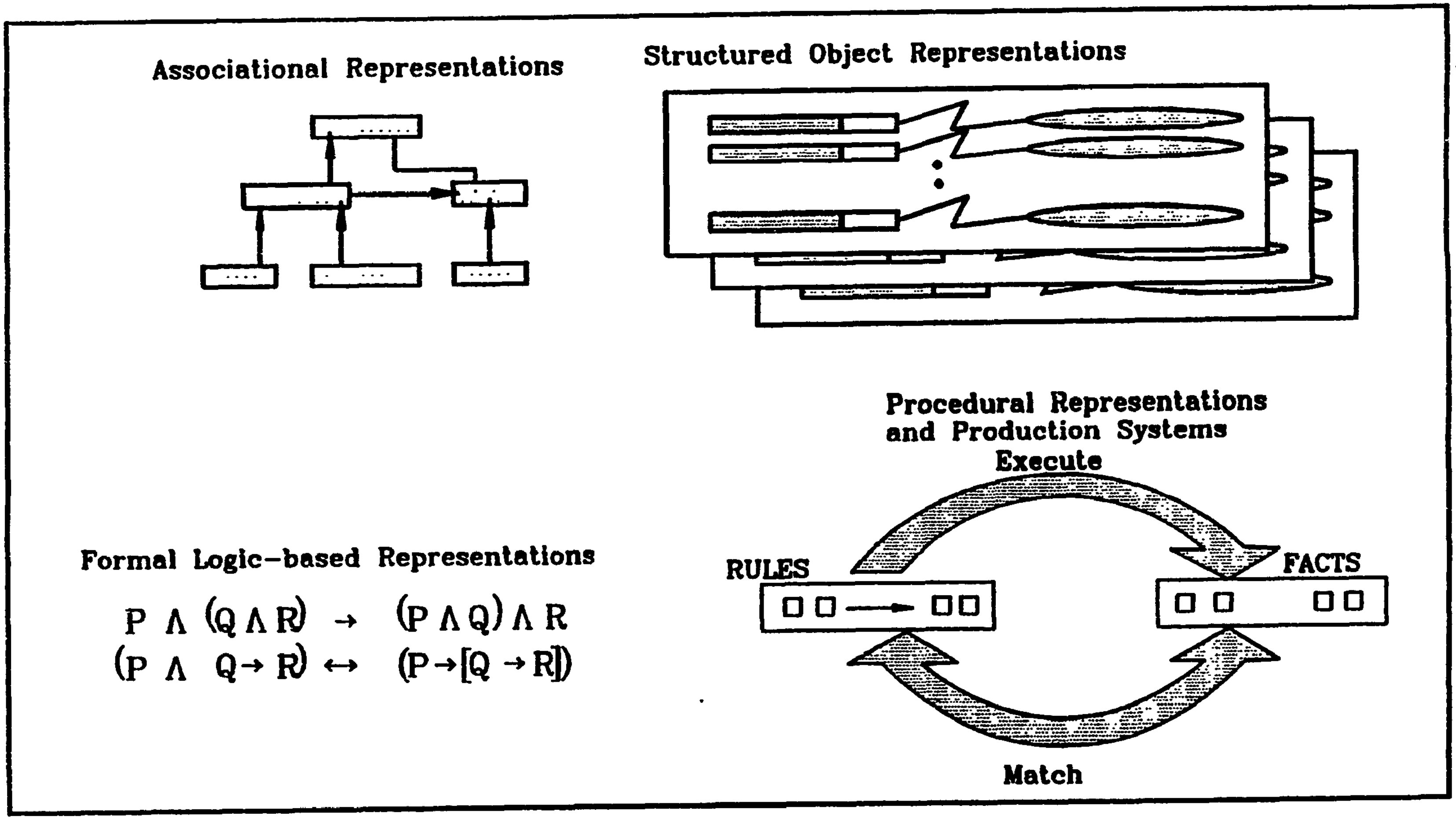


Figure 1.15: Classes of knowledge representation[74].

Since a rule-based system does not require detailed advance knowledge about the flow of control it can better accommodate dynamic and rapidly changing conditions. Furthermore, rule-based systems provide a multiple line of reasoning capability. This capability is derived from the way rule-based systems automatically cycle back to find all the rules that have satisfied IF parts. Rule firing via a match-execute sequence is illustrated in Figure 1.16.

B) FRAMES

A frame is a collection of standard features that describe an object, act or event. It resembles the data structure constructed in conventional programming languages. The features in a frame are denoted in terms of attributes (called slots) and their contents. The contents of a slot are either actual values or procedural attachments. A frame structure is shown in Figure 1.17.

Frame-based representation is a structured object representation. It structures object into categories which constitute the underlying framework for constructing hierarchies, taxonomies and stereotypes. A frame based representation is ideal for applications which are predictable because it supports the notion of standard stereotypes. Stereotypical situations are represented by attaching the following type of information to the slots of a frame:

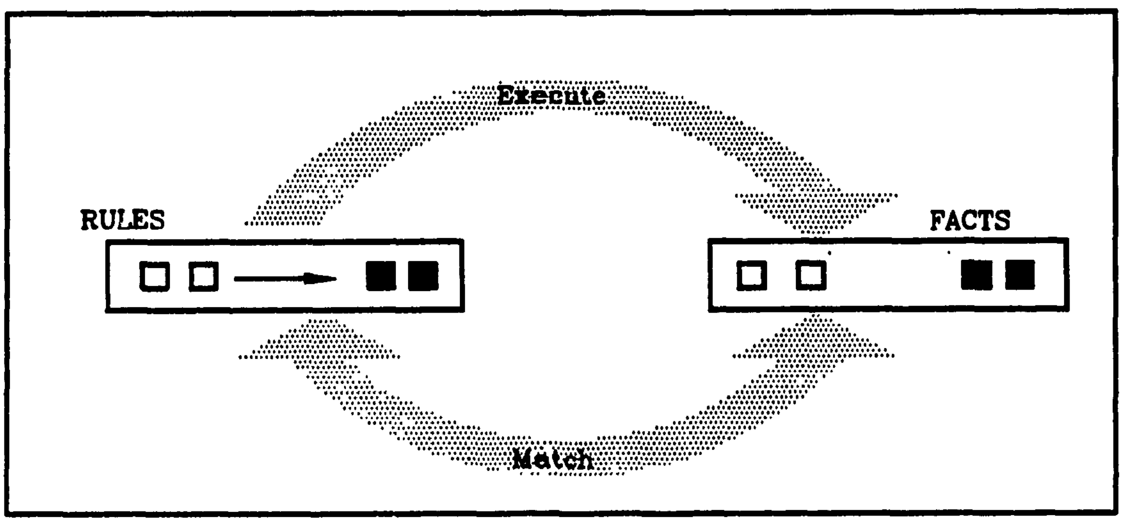


Figure 1.16: Production rule system[74].

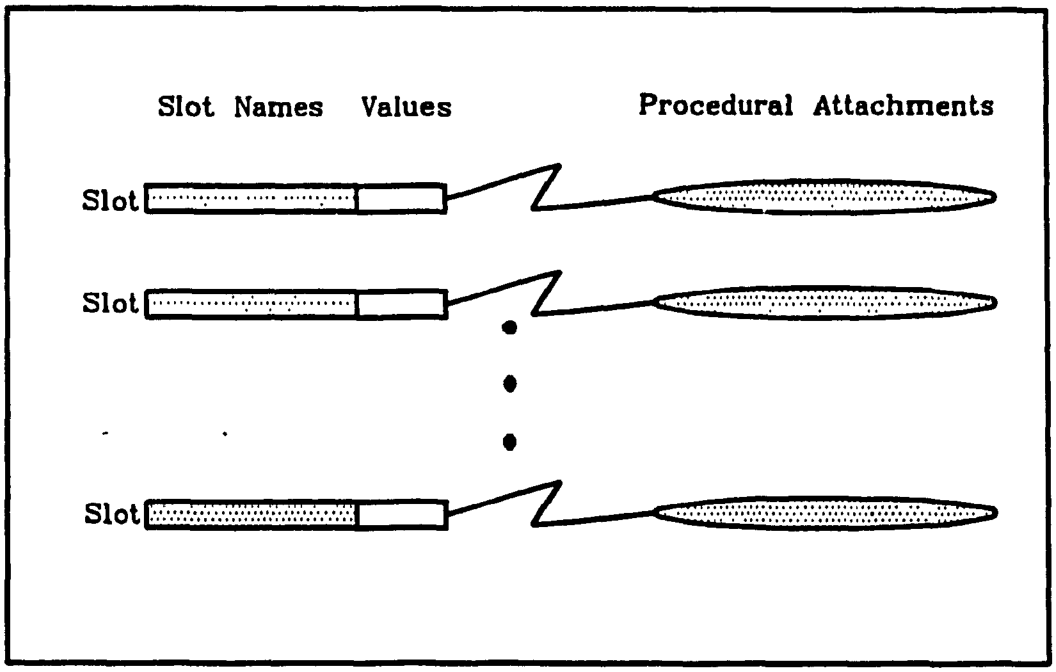


Figure 1.17: Frame structure[74].

- How the frame is to be used;
- Scenarios that can happen to the frame;
- Plans on how to react to each expected scenario.

C) SEMANTIC NETS

A semantic net is a form of associational representation. It is intuitively easy to understand because it offers a graphical structure to facilitate visualization and comprehension. The representation is composed of nodes, interconnected by facts about the object. Each link explicitly expresses a relationship between a pair of objects. The links connect individual objects into a cohesive structure.

A node and link net is not necessarily a semantic net. There must be semantics in order for it to be a semantic net. the semantics of a representation specifies how meaning is embodied by the symbols, while the symbols are arranged in accordance with the syntax. Therefore a semantic net must have at least a *prima facie* description, and in some cases, the accompanying procedure that enables the description is also included.

Instantiation is a well-known aspect of semantic net formalism. The term "instantiation" refers to the reproduction of a more general class description. Each instantiation has some unique characteristics that sets itself apart from the others. In most notations, an instance is the construction of an individual description based on a generic class description.

1.7.3 INFERENCE ENGINE AND INFERENCING TECHNIQUES

The inference engine solves a problem by interpreting the domain knowledge contained in the knowledge base. The two most commonly used reasoning mechanisms used to interpret the domain knowledge are 'backward' and 'forward inferencing'. Other control strategies include the use of meta-knowledge and the blackboard mechanism. Meta-knowledge is the knowledge about the contents of the knowledge base and knowledge about reasoning strategies. Meta-knowledge provides the inference engine with additional flexibility and power in choosing the appropriate approach to solve the problem at hand. A blackboard is a highly appropriate approach to solve the problem at hand. It is a highly structured workpiece. The blackboard mechanism allows multiple knowledge sources to pursue various aspects of a problem independently, periodically combining their result.

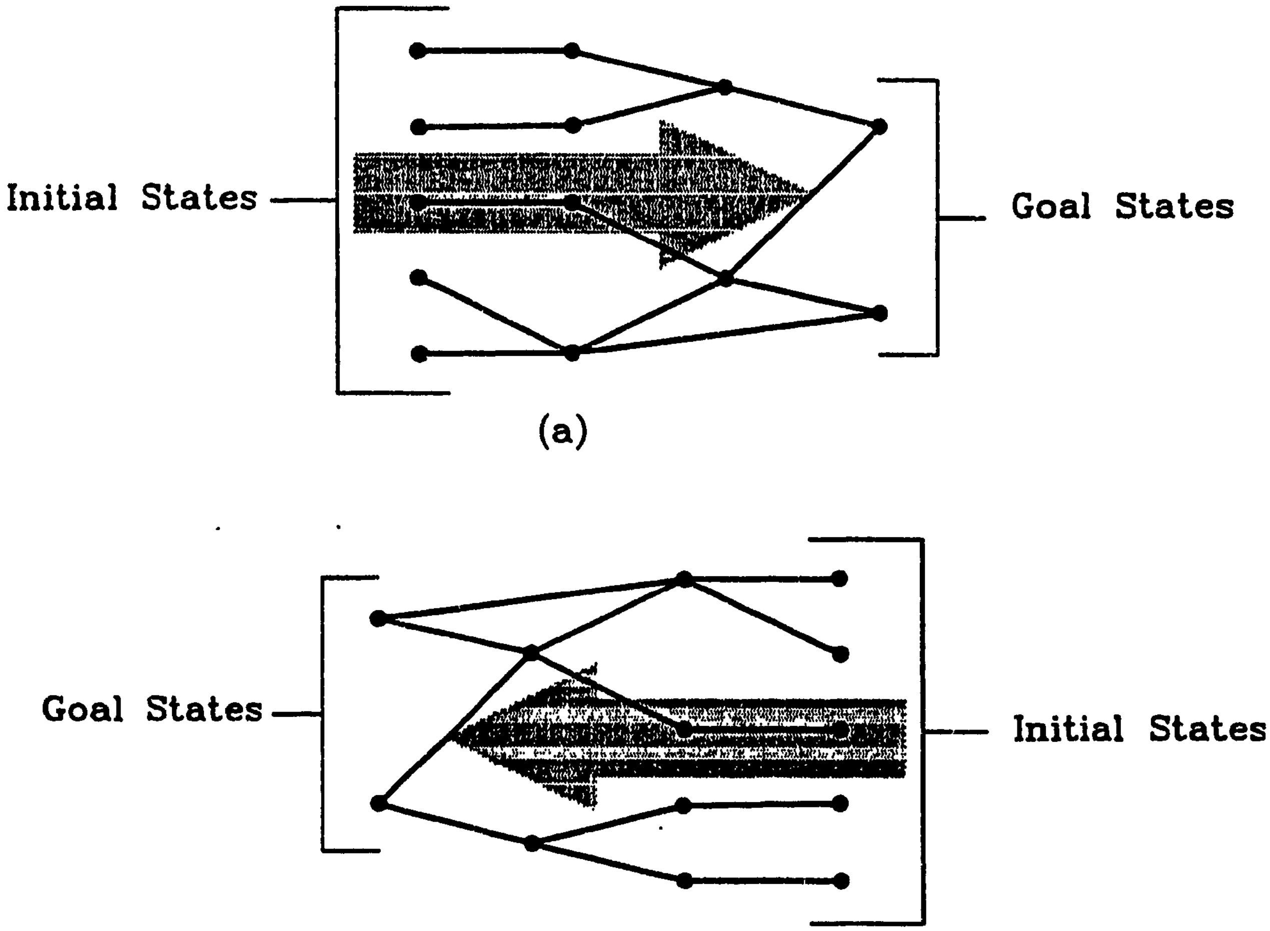
In backward (goal driven) inferencing, the inference engine works with a hypothetical solution (the goal) to find evidence supporting the hypothetical solution. Often, it entails formulating and testing intermediate hypotheses (sub goals). With production systems, backward inferencing is achieved by a process called *backward chaining*.

The inference engine begins by searching the knowledge base for a rule (or rules), whose firing would give the desired conclusion. It then attempts to match the antecedent parts of the rule against the initial problem description stored in the working memory. If the antecedent matches, the search is finished. This inference engine then searches the knowledge base for another rule whose firing would satisfy the first rule. This process continues until either a rule is found that has a antecedent matching the initial problem description or the inference engine asks for information from the user.

In forward (data driven) inferencing, the inference engine attempts to reason forward from the facts to a solution. A sample data driven control strategy is to gather some initial evidence. The inference engine then considers each item of evidence in turn, attempting to work forward to a goal. Data driven reasoning will never terminate if the initial evidence does not lead to a conclusion. For this reason, most expert systems are primarily goal driven. It is also common to combine the use of both goal and data driven strategies. First, data driven inferencing is used to suggest a set of hypotheses based on initial data. The inference engine then begins to consider each hypotheses in turn, as in a pure goal driven system. However, when a new item of evidence is discovered during backward inferencing, the inference engine switches to forward inferencing to see if the new evidence would suggest another goal or a short cut in achieving the current goal. Forward inferencing and Backward inferencing are illustrated in Figure 1.18(a) and 1.18(b) respectively. The architecture of an expert system is shown in Figure 1.19.

Even though these systems are reffered to as expert systems, it is really an acronym. The so called expert system adopts heuristic search methods to reach the goals. It must be surprising to know that heuristics are little used by human experts Although we must be cautious in blindly aiming to replicate the structure of human brain, a good deal of insight can often be gained by considering some relevant aspects of human cognition. Analysis of human problem solving [Ernest and Newell, 1969; Newell and Simon, 1972] has shown that while general GPS[2] like methods such as means-ends analysis are sometimes used by people, in many situations where expertise is demonstrated, the human problem solver attempts to capitalize on as much of the problem specific knowledge as is available. In general human experts appear to do little search. In fact, human experts are often admired for doing little search, but for formulating the problem in a way that makes its solution apparent. This is well illustrated by the difference between an expert who can quickly diagnose a malfunction and a novice who must search through a vast number of possibilities.

[2]General Problem Solver



(b)

Figure 1.18: Inference methods[74].
(a). Forward chaining
(b). Backward chaining

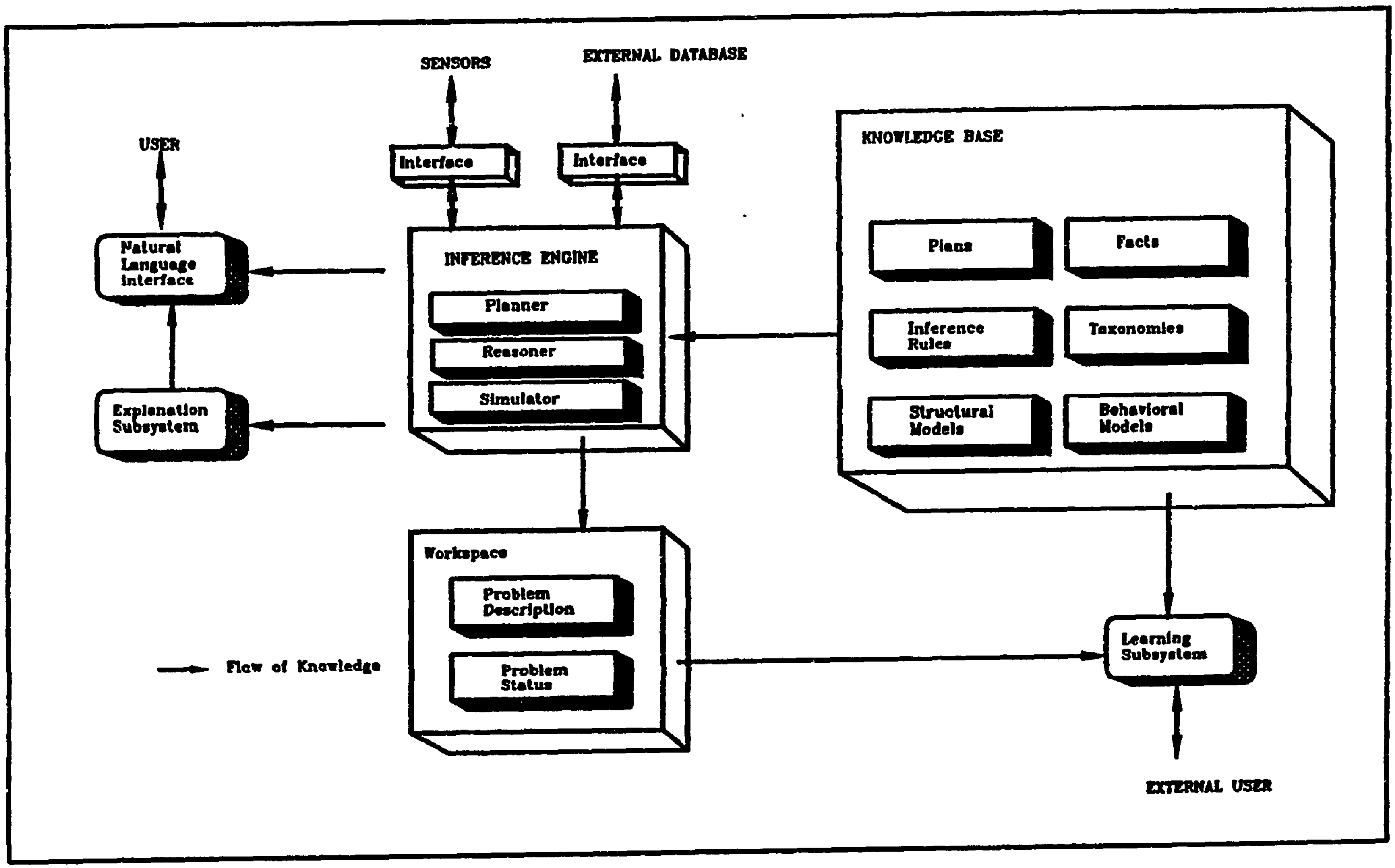


Figure 1.19: Architecture of an expert system[74].

The development of smart heuristics that search massive trees of possibilities is generally not as effective as the structuring of knowledge and the development of much smaller sets of possibilities. It is this emphasis on knowledge structuring rather than on search heuristics that distinguishes the expert systems approach from other AI approaches to problem solving [Adeli, 1990]. Analysis of human behaviour has also suggested that a large component of reasoning can be captured in the form of production system that uses a set of IF - THEN - ELSE rules [Anderson, 1983; Newell and Simon, 1972]. In such cases simply following a set of rules of the form "IF 'X' happens, then do 'Y'". Such systems provide a convenient format for combining general - purpose reasoning with specific knowledge.

A further fact to consider is that considerable amount of research supports the hypothesis that human memory is organized into at least two separate storage mechanisms: *long term* and *short term* memory long-term memory persistently stores many facts and structures over a period of time, while short term memory is relatively transient and information contained in it is soon forgotten, if not transferred to long term storage [Parsaye, 1988; Adeli, 1990]. Both of these memory classifications are useful in controlling a real time process. The long term memory containing the process knowledge while the short term memory contains current situations, previous actions etc.

Until very recently, applications of AI have been confined to situations where the user communicates with the AI software by means of a computer in an interactive mode [Parsaye, 1988]. This has resulted in the development of many computer softwares especially in the area of expert systems. It should be emphasized that the off line expert systems, even though, very useful in an industrial environment, are hardly useful in a real time situation which abound in such environment. The capabilities and the advantages of expert systems should be fully utilized to enhance the productivity. There is currently a large amount of activity directed at these goals. Expert system applications in real time situations have come alive in the field of computer vision, speech, character recognition and robotics [Adeli, 1990]. The most challenging area of real-world involvement is in the area of real-time control systems in which there are two very different processing functions [Wright, 1986]. The domain of expert systems should be confined to the solution of problems which make use of facts that are represented in symbolic form. In contrast, the real-time part of the system is concerned with sensing, actuation and the numerical manipulation necessary to supply the expert with information and implement actions proposed by the expert [Krijgsmann et al., 1991].

The typical environment required for an expert system consists of a symbolic language such as C or C++ running on a processor with considerable computing power and memory. Even with these provisions, the nature of the expert system is such that the computation time is highly variable and uncertain[Neapolitan,1986]. For example, in a backward-chaining inference engine for a rule based system the computing time can be determined as follows:

If 'N' is the number of rules, the maximum number of assertions deduced will also be 'N'. In the 'worst' case,

one assertion is added at the first rule, that is, after visiting 1 rule,

a second assertion is added at the second rule, that is after visiting 2 rules,

.

.

an Nth assertion is added at the Nth rule, that is after visiting N rules.

The total number of rules visited is then : $1 + 2 + \ldots + N = (N+1)N/2$,

and the computing time of the algorithm is $O(N^2)$.

Where O is the symbol of 'On the order of'.

It can be easily seen that if 'K' is the maximum number of premises within a rule, the computing time of the algorithm is O(K), and if 'M' is the number of true assertions, the computing time is O(M).

although the computing time is $O(N^2)$ in the worst case, in general the time should not increase proportional to N^2. This is due to the fact that the time really depends on the number of assertions added, since it is necessary to start the list traversal over whenever an assertion is added. The number of assertions added depends completely on the rule mix and the true assertion list, so it is totally unpredictable [Neapolitan, 1986].

1.8 REQUIREMENTS FOR REAL-TIME CONTROL EXPERT SYSTEMS

Issues particular to the design of expert system for real-time environment have been surveyed by Moore and Kramer [1986], and Sauers and Walch [1983]. The necessary features for such a system are:

1. *High-speed context sensitive rule activation :*

   Real time systems must be able to focus their attention on relevant sets of sensors and collect data at accelerated rates during critical events. This capability can be explicitly provided by allowing separate scan and alert intervals to be established for all rules and by associating rules with units or conditions. The focus mechanism explicitly brings rules into the scheduling process only when they are needed.

2. *Efficient recycling of memory elements that are no longer needed and maintenance of sensor histories:*

   When a real-time system runs continuously for long periods of time it is important that it "leaves its past behind" in order to avoid filling memory with useless remnants of old inferencing activities. It is also important, however that histories of sensor values be available for as far back as any rule requires them. Explicit management of both the garbage collection process and the archiving of sensor values as required by rules which refer to the past history of sensors (such as an average or rate of change) will facilitate continuous operation.

3. *Real-time expert systems must be able to interactively accept command sequence from the operators:*

In many application the system cannot stop, so it is important to provide for a means of "steering" which allows human operators or other expert systems to direct its activities. These steering commands can be regarded as special sensor inputs. Conditional command sequences are also a desirable feature which allow stored conditional execution of commands to be implemented with standardized responses or insufficient time would be available for evaluation of the response to the initial commands to allow full formulation and entry of subsequent commands.

4. *Communication between multiple expert systems:*

For reasons of redundancy and due to the sheer volume of process monitoring associated with many real-time operations, it will often be necessary to employ multiple expert systems as process controllers. Real-time expert systems should be organized so that an instantiation of one expert system can access the variables within another and perform inferencing tasks based on these observations. Cooperating expert systems can be hierarchically organized so that a master supervisory system can be constructed to provide oversight and summarized information regarding the operation of the component expert system.

O'Neill and Mullarkey [1983] have defined the main objective of the expert-systems for real time control environment as:

Signal interpretation - in which the objective is to infer the state of the external environment based on the sampled signal(s)

Multi-sensor fusion- in which the objective is to combine information from several different signal sources in order to arrive at a more complete description of the environment than could be determined from any individual sensor.

Anomaly detection - in which the objective is to determine whether the system under study is behaving in an anomalous manner.

Anomaly classification - in which the objective is to determine whether anomalous behaviour is due to a problem in the system under study or is due to some artifact of the environment or monitoring process.

Problem diagnosis - in which the objective is to determine the cause of a problem in the system under study.

There are a number of different levels of communication interfacing to considered in designing the interface between the real-time system and the expert system within [Ever et al., 1984]. The lower levels are concerned with the electrical and logical process of transmitting the receiving data. This has to be considered in any system design. The primary concern, however, is with the upper levels relating to information content. Whereas an expert system needs its data to be in symbolic form, the real world operates in terms of switch closures and voltage levels. The real-time controller must make the translation of signals into symbols to be used by the expert system. Its function would be:

* Acquisition : analyze the available information.
* Transmission: encode and transmit the results in symbolic form.
* Reception: receive and decode symbolic commands.
* Implementation: implement real-time control sequences.

Evers at al.[1984] recommended that communications be passed through a common interface handler, thereby creating a single 'point of contact' for the expert system. One of the tasks must then become the master of this communications link using the other as the 'slave'. Typically, the real-time system becomes the master and the expert is a slave advisor providing decisions on demand. Fox [1983] points out that expert systems functioning in real-time using sensor values acquired automatically (as opposed to information solicited from the user) need to make determinations regarding the quality of this sensor data since the sensing devices are subject to failure or degradation over time. A real-time expert system shell should support this "meta-diagnostic" function by providing facilities that allow checking of sensor histories and consistency between measured and derived values. Much of this activity can be off-loaded to a parallel data acquisition system. Last, but not the least important requirement of an expert controller is that it should be capable of detecting anomalies in the sensors and if one of the sensors is not providing the signals it should be capable of alarming the engineer to rectify the situation instead of inferencing the situation as one of the current situations.

1.9 SCOPE OF RESEARCH

Automation of manufacturing systems has been identified as one of the most important keys to the economic success. This has lead to the factory automation aided by recent advances in computer technology and networking. Importance of deep hole machining in a manufacturing system has been recognized and its application is constantly growing. In order to integrate deep hole machining system in to an automated manufacturing environment, it is necessary that the deep hole machining process be automated at the machine hardware level. The automation techniques based on mathematical theories are not sufficient for the automation of deep hole machining owing to its complexity. Efforts to use Artificial Intelligence (AI) techniques such as expert systems for automation of complex systems which are not amenable to mathematical modelling has lead to their application in real time control of machining units such as machining centres. Deep hole machining, due to the complexity and inherent uncertainty of the process, is an ideal candidate for automation using an expert system. In order to utilize the capabilities of expert systems for real time control of deep hole machining system, it is necessary to represent the deep hole machining process in form suitable for symbolic processing. Furthermore, the knowledge or the expertise of deep hole machining, which forms the sole of such a system has to be acquired and coded for symbolic interpretation. Thus the goal of this research is to develop, implement and validate a real time control for deep hole machining system using expert system techniques.

CHAPTER 2

BUILDING AN EXPERT CONTROLLER FOR DHMS[1]

2.1 INTRODUCTION

Research in the use of advanced control methods for control of manufacturing process grew steadily in the last decade [Galip Ulsoy et al, 1993]. This growth in automation research can be attributed to several factors: (a) Increase in the demand for high quality, precise components, (b) Demand for better productivity, (c) Increase in the complexity of machining process (d) failure in development of generalized model for machining process, (e) Significant strides in computer hardware and software technology, and (f) advances in artificial intelligence (AI) techniques for real time controls. A detailed account of the application of all the above are discussed in the previous chapter.

The focus of this research is the real-time control of deep hole machining operation.

---

[1]Deep Hole Machining System

Taking into account the complexity and inherent qualities of the deep hole machining process (Chapter 1), the next logical step would be to implement such a control system using expert system techniques (Henceforth referred to as expert control of DHMS).

Development of expert control involves many stages. The first and foremost is the definition of goal of expert controller. Acquisition of knowledge from the experts, classification of knowledge, designing the knowledge base and inference techniques and finally the implementation of such a system is called knowledge engineering and is carried out by the knowledge engineer. This chapter deals with the specification of goals and knowledge engineering to build an expert controller.

Two expert system approaches exist for closed-loop control. The first one is based on fuzzy logic and the second is rule-based approach [Tong, 1984; Arzen, 1989]. In both the approaches the goal is to model the manual control strategy of the process operator reinforced with knowledge obtained from experimental investigation. The manual control strategy is expressed as qualitative linguistic rules for how to choose the control signal in different situations. The rules thus replace conventional control algorithms.

The rule-based or the knowledge based approach is closer in spirit to conventional adaptive control than fuzzy control is [Arzen, 1989]. The approach is motivated by shortcomings of current adaptive controllers such as the requirements for prior process knowledge poor user understanding, and complex safety nets. The recursive identification algorithms and the control algorithms of a conventional adaptive controller can be seen as the final algorithmic representation of a large amount of underlying theoretical as well as practical control knowledge [Astrom and Wittenmark, 1989]. For practical realization , it has to be combined with logic safety jackets or safety nets [Isermann and Lachmann, 1985; Warvick, 1988] often heuristically derived, that assures controller performance under non-standard conditions such as switching between different operating modes, insufficient process excitation, control signal saturation, etc. This often is the domineering part of the controller and experience has shown that design and testing is highly time consuming. The expert control approach is to use expert system techniques to replace the conventional methods as much as possible.

2.1. THE GOAL OF EXPERT CONTROL OF DHMS.

The goal of expert control is a controller:

- # That can satisfactorily control the deep hole machining process which is highly uncertain, time varying and exposed to various unknown disturbances.

- # Which requires a minimal amount of prior process knowledge.

Which utilizes the available prior knowledge to its advantage.

Incorporates qualifiers namely, bigger than, steep, abrupt, small etc, entered by the user as specifications.

Constant learning of process knowledge in order to gain the real expertise.

That can perform self diagnosis and is able to detect any actuator and/or sensor malfunctioning.

That can actively communicate with the user, give reasons for its actions and take overriding instructions from the user, give performance and statistics of this operation when requested, generate a report.

when the underlying control of process knowledge is transparently stored in a way that easily allows for modification and extensions.

2.2. ARCHITECTURE OF THE EXPERT CONTROL SYSTEM:

Basic architecture of expert control system consists of (a) expert system which forms the soul of the expert controller and (b) an interface system which acts as a messenger between the DHMS and the expert system. The structure of the expert system has already been discussed in the previous chapter. The structure of the interface is the assembly of data acquisition system, the sensors on the DHMS and the machine control units. The architecture of the system is illustrated in Figure 2.1.

Designing of the e·pert control system for DHMS according to the architecture involves:

i. Knowledge Acquisition
ii. Designing the Knowledge base
iii. Designing the Inference Mechanism
iv. Implementation of expert controller, and
v. Validation.

Knowledge acquisition, Designing the Knowledge base and Inference mechanisms is termed as Knowledge engineering.

2.3. KNOWLEDGE ENGINEERING

A knowledge Engineer is one who extracts domain knowledge from experts and translate this information to a format suitable for the inference engine. The traditional knowledge engineering approach follows EXPERT ⇒ KNOWLEDGE ENGINEER ⇒ KNOWLEDGE BASE paradigm. The domain encompasses the deep hole machining process and general theory of metal machining. The knowledge engineer is the researcher who identifies the experts in the domain and devises appropriate knowledge

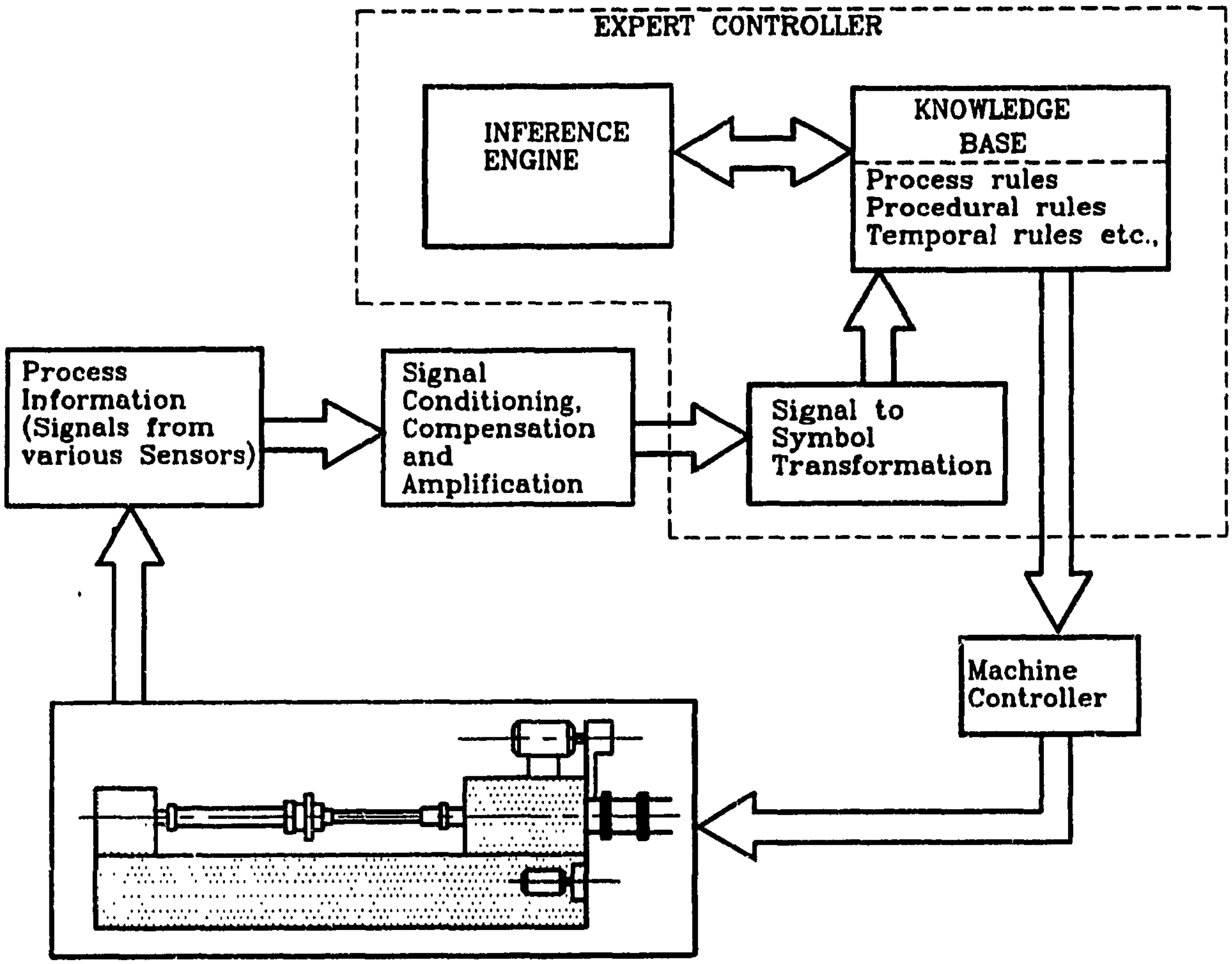


Figure 2.1: Expert control architecture.

acquisition techniques from such experts.

Knowledge acquisition for practical system has come to be termed knowledge engineering following Feigenbaum's [1978] use of the term to describe the reduction of large body of knowledge to precise set of facts and rules. Basic model of Knowledge acquisition by a knowledge engineer is shown in Figure 2.2. Role of the knowledge engineer in building any expert systems is of the utmost importance. Knowledge engineer is the architect of the entire system and the role of knowledge engineer is shown in Figure 2.3.

In order to model the expertise of the DHMS expert and a skilled operator for exploitation in the expert controller, the following must be considered.

- Fundamental limits of what can be achieved by an expert controller.
- Type of knowledge that can be elicited from the experts and operators.
- How that knowledge should be represented.

Each of the above, whether a formal theory or an informal description, whether explicitly stated or implicitly invoked, can be viewed as a model of human expertise in DHMS. A model focuses attention on certain aspects of a phenomenon and de emphasizes others. Which aspects are identified as important depends, in part, on the

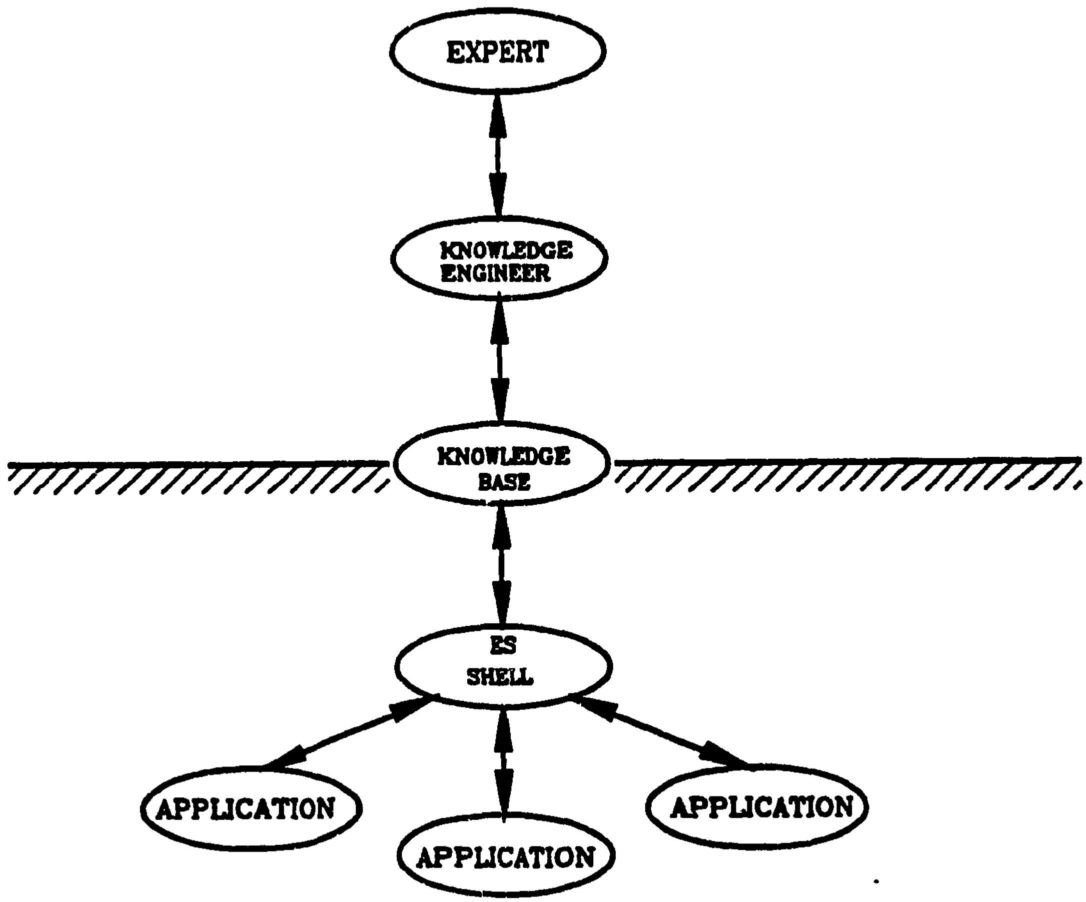


Figure 2.2: Knowledge acquisition by knowledge engineer[2]

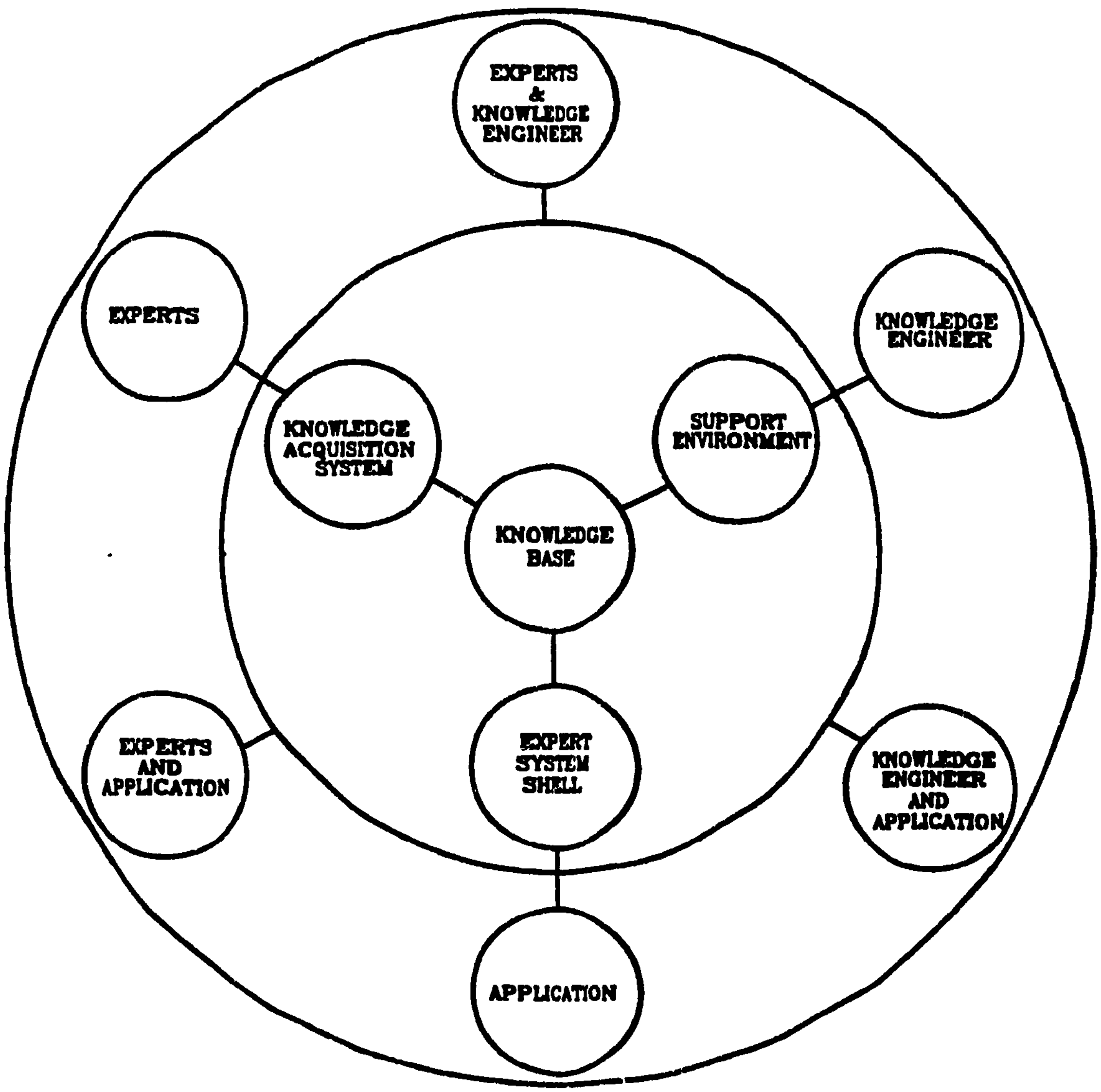


Figure 2.3: Role of knowledge engineer[2]

purpose of the model. The fundamental duty of deciding on each of the above items and acquisition, modelling, encoding and testing of expert system for expert controller falls on the knowledge engineer.

Knowledge engineer has to develop certain support tools for acquiring and coding the knowledge into the expert system. Figure 2.4 groups the support tools for editing, display, encoding and validating the knowledge into a knowledge based system support environment and lumps various forms of knowledge bases together. The knowledge-based system support environment is illustrated in Figure 2.5. The quality of this environment is very important in the development of expert system for any real time expert system in general and Expert controller for DHMS in particular. The relation of this environment to knowledge acquisition is best seen by putting both in the context of the complete virtual machine hierarchy for knowledge base systems.

(1). Machine architecture:

At the lowest level of the hierarchy is the machine on which the expert system runs. For an off line expert system such as an expert adviser, this is irrelevant at present due to advancement in computers. For an expert controller, the machine should be able to provide for communication with the object to be controlled namely the deep hole boring machine. It is thus imperative that the computer should be

8	KNOWLEDGE SUPPORT SYSTEM	Integrate acquisition and performance in a real time environment.
7	KNOWLEDGE ACQUISITION TOOLS	Elicit knowledge from experts, Model operator behaviour, Observe, Generate rules, Classify rules.
6	KNOWLEDGE BASE SYSTEM SUPPORT ENVIRONMENT	Display, Edit, Debug and validate the conceptual structure, inference rules and Performance of Knowledge base.
5	KNOWLEDGE BASE SYSTEM SHELL	Elicit problem specific data, apply and explain the plication of the knowledge base run time environments for the above system.
4	SHELL DEVELOPMENT LANGUAGE	Provide suitable facilities for the effective representation of knowledge and efficient inferences from it. Natural language interface.
3	IMPLEMENTATION LANGUAGE	Provide high-quality implementation environment for complex programs for computer communication with dynamic data structure and memory scavenging.
2	OPERATING SYSTEM	Real time operating system to work with common operating system and computer communication with real time data acquisition system.
1	MACHINE ARCHITECTURE	Establish the capability of the machine to handle the real-time operating system and communication/interface boards. Selection of communication boards to fulfil the input/output requirements.

Figure 2.4. The knowledge base support environment hierarchy for expert control system[2].

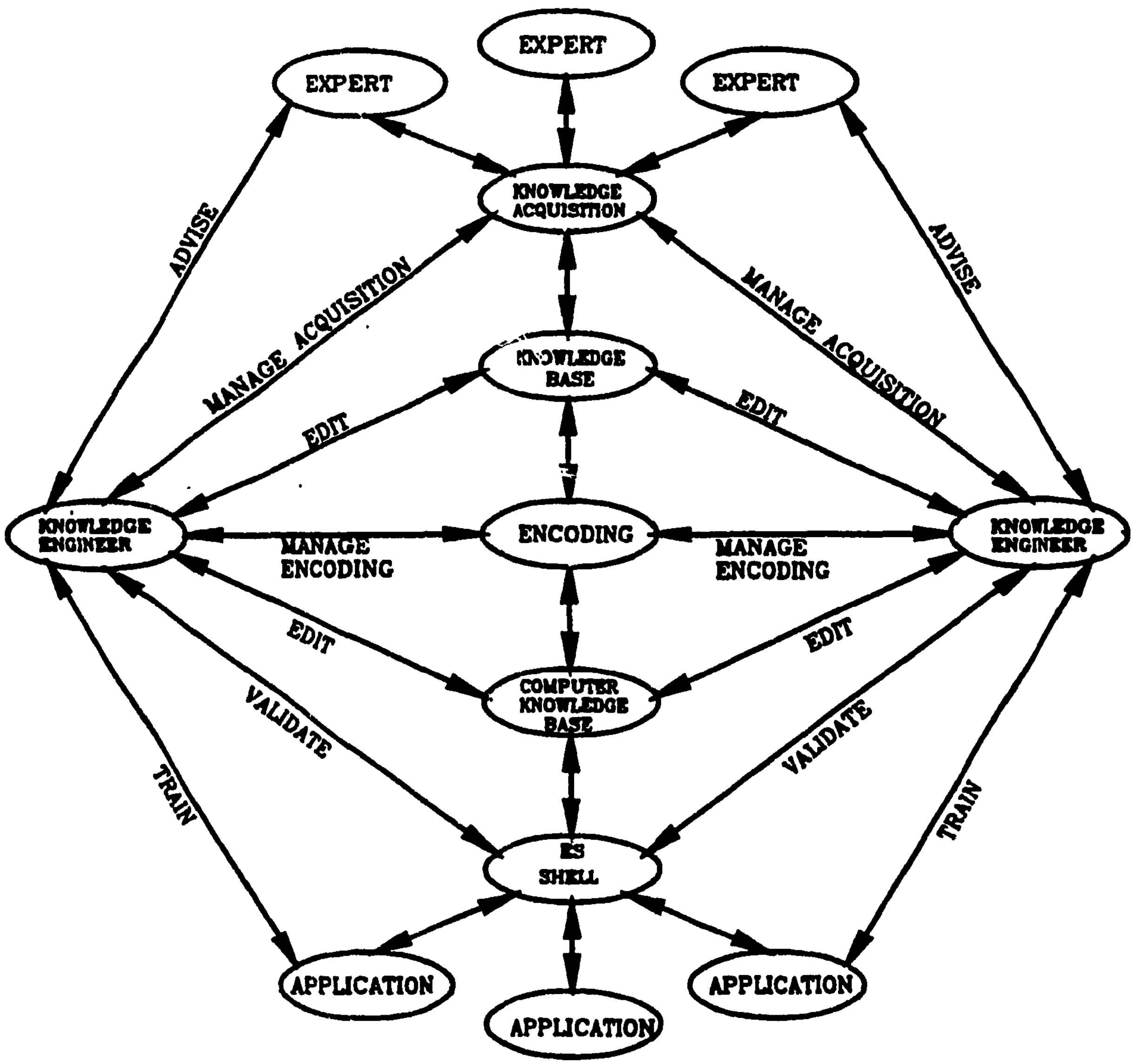


Figure 2.5: Knowledge base system support environment[2]

capable of accommodating the interface hardware to provide communication facility with the environment. The communication hardware is selected in such a way that its specification facilitates the burden of real time signal conditioning and number of input and outputs in analog or digital form as suggested by the expert controller.

(2). Operating system:

At the second level of hierarchy is the operating system within which the implementation runs. The system should be capable of operating in common machine operating system environments such as DOS , UNIX etc which are not real-time operating systems. The real-time operating system needs to provide good interfaces to other programs, large databases and communications.

(3). Implementation language:

At the third level of the hierarchy is the implementation language which actually interfaces to the computer. This position of hierarchy was previously reserved for LISP for off the line expert advisors. For a real time expert controller, the speed and space efficiency have become significant. The most ideal candidate for this hierarchy is the ASSEMBLY language. Difficulties involved in understanding

ASSEMBLY language by experts tend to develop suspension about the system and hence the next higher level language C takes its position. The other advantages of using C as implementation language is that it is strongly structured and supports dynamic implementation of data structures such as stacks, linked lists, ques etc. The inherited properties of some of such data structures is ade use of in the expert system development and safety jacket for the expert controller.

(4). Knowledge base development language:

At the fourth level of hierarchy is the knowledge base development language. This is an important hierarchy as this language should be easily understood by the expert who may or may not be familiar with computer languages such as C. An ideal system is a natural language system which will compile the knowledge represented in natural language into a complex format to be presented to the inference engine.

(5). Knowledge based system shell:

At the fifth level of the hierarchy is the expert system shell, i.e., expert system without knowledge about the domain. The shell incorporates the various modules of expert system namely, the knowledge base, inference engine, reasoning facility etc., The data acquired by the interface converted into symbols. This conversion activates the inference engine for rule matching and then initiating the action.

(6). Knowledge based system support environment:

At the sixth level off the hierarchy is the Knowledge based system support environment. This is also termed as user interface. This hierarchy provides editing, browsing, and debugging tools for the knowledge base and also, for over riding commands which the expert controller is at work.

(7). Knowledge acquisition tools:

At the seventh level of the hierarchy are the tools for automating the systems analysis for expert systems, through automatic knowledge acquisition procedures, modelling of actual expert behaviour, and analysis of knowledge in textual form. Automation of systems analysis is more significant for knowledge-based systems than for conventional data processing because knowledge-based systems tend to be targeted on narrow areas of in-depth specialist knowledge, rather than broad general problems. In conventional terms, knowledge-based systems are satisfying the requirement for highly customized data processing based on idiosyncratic operations rathe- than general algorithms.

(8). Knowledge support systems:

At the top of the hierarchy are the objectives of recent research in knowledge acquisition which focuses on the integration of acquisition and performance systems to provide an environment supporting a wide variety of knowledge process.

2.3.1 EVOLUTION OF EXPERTISE

Is highly unlikely that the expert knowledge in a given system like DHMS is universal, i.e., the acquired knowledge cannot fulfil the exact needs of a particular deep hole machining plant. Complete and comprehensive knowledge about a particular plant or installation can only be obtained over a period of time. In order to accommodate this period of gradual building of exact knowledge the expert controller should have to go through two stages:

A. Supervised learning period, and

B. Unsupervised operation period.

In supervised learning period, the generalized knowledge elicited from the experts was used for carrying out the deep hole machining operation. Each operation generates a report on the process and the particular action initiated during the process. These reports form the exact knowledge about the particular installation. Supervised learning period eventually generates comprehensive knowledge about that installation which will then serve as the knowledge for unsupervised operation. In unsupervised operation period, the expert controller does not need major modification to its knowledge base. A fine tuning once in a while to bring the knowledge base up to date due to various time dependent changes in the machine and operational conditions is performed.

2.4. KNOWLEDGE ACQUISITION

2.4.1 Domain specification:

The domain of the problem is the entire deep hole machining operation as illustrated in Figure 2.6. The main components of DHMS are The Machine, The Tool and The Workpiece. The coordinated action of the machine and the tool on the workpiece constitute the deep hole machining process. Each of these main components of DHMS has an enormous bearing on the outcome of the machining process and hence forms the domain of the problem. Since the workpiece is not an integral part of the deep hole drilling machine and can be any one of the vast variety of metals and alloys available in the market, its influence on the expert control system should not be overemphasized.

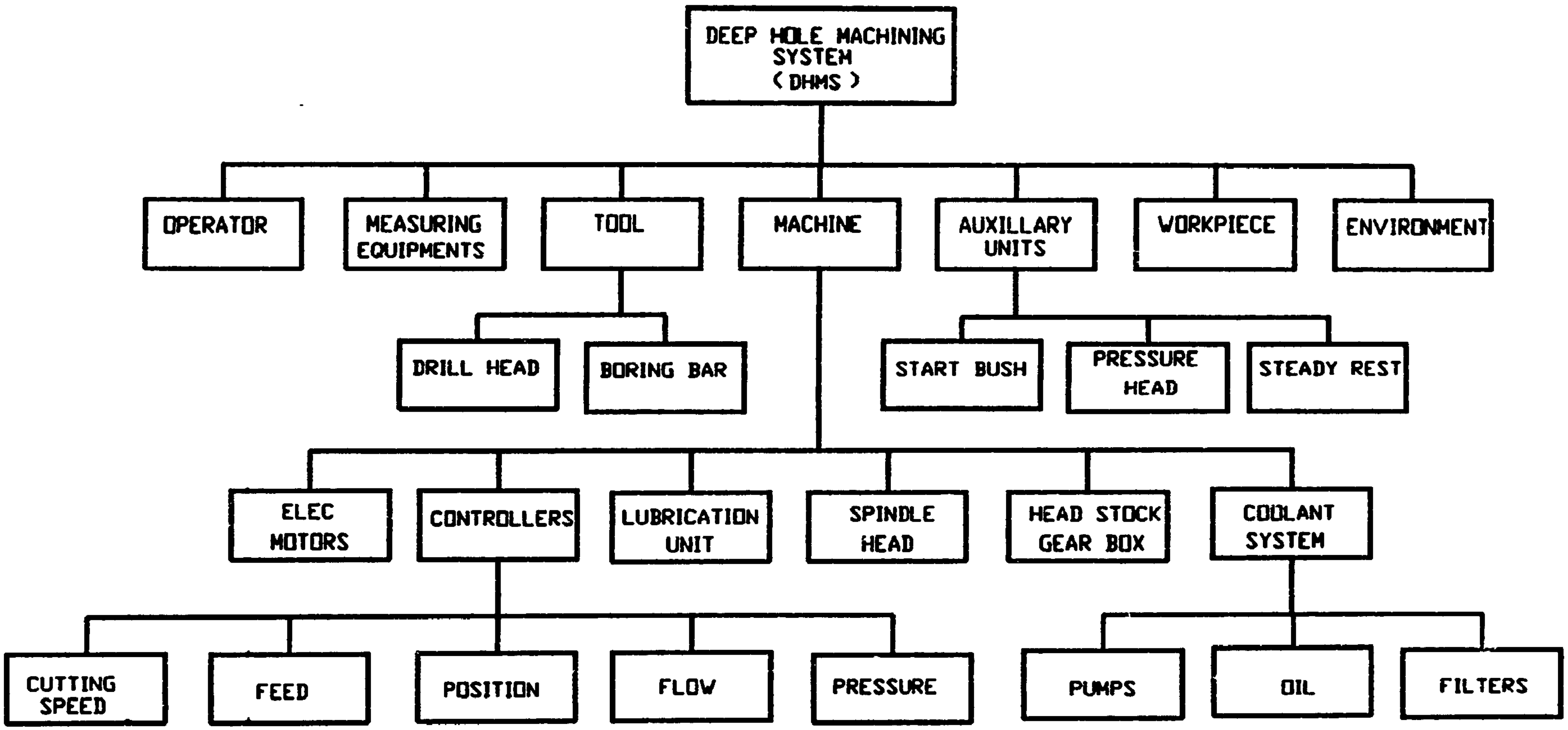


Figure 2.6: Problem domain - Deep hole machining system.

The Machine: The deep hole drilling machine is a 20HP Schaerer HPD 631 Lathe converted into a deep-hole drilling machine as illustrated in Figure 1.2. The machine is capable of providing a continuous ranges of speeds and feed rates. Continuous variation of cutting speed is enabled by a 20HP variable speed AC motor which drives the cutting tool via a boring bar assembly. Feed variation is incorporated using a 3HP AC variable speed motor which drives a lead screw via a worm reduction gear box. The rapid feed features of the deep hole machine are provided by a 2HP rapid motor. Rapid forward movements and retraction of the tool and boring bar assembly is carried out by the rapid motor. Both the spindle motor and the feed motor are controlled by respective AC inverters. Work piece is held stationary by the chuck on the head stock spindle. The machine is fitted with a high pressure coolant supply system. The flow rate of the coolant to the machining zone can be varied by a flow control valve provided in the flow circuit. The machine is also provided with auxiliary units such as start bush, pressure head and steady rest.

The Tool: The tool can be any of the deep hole drilling tools namely Gundrills, Ejector drills or BTA Drills etc. Even though any of these tools should be able to perform the deep hole machining operation, only Ejector drills and BTA drills are considered for the research.

The Ejector drill shown in Figure 1.1(c). is a special deep hole drilling tool wherein the coolant is introduced between the outer and inner tube. Approximately one third of the coolant is drawn off through the annular nozzle straight back through the inner tube to create a partial vacuum in the chip passage. The rest of the coolant passes through the machining zone, cooling and lubricating the cutting edges and burnishing pads of the tool. The coolant from the machining zone flows back to the coolant tank along with the chips. This flow of swarf to the tank is influenced by the partial vacuum created in the chip passage.

The ejector tool incorporates a three-edge arrangement. Two edges are located on one side and the third is located at 180 degrees from the two edges. The cutting action of the edges is progressive and overlapping. The supporting pads carryout the burnishing operation on the bore wall.

A typical BTA tool is shown in Figure 1.1(b). The BTA drilling tools are single edge or multi edge cutting tools with internal chip removal. The guide bushes, in addition to burnishing, enable the balancing of cutting forces. Figure 1.4 shows some typical BTA tools.

The deep hole drilling tools are very expensive and hence their life and utilization has to be maximized for economy. The life of the tools are directly effected by the efficiency of the coolant flowing into the machining zone and swarf removal efficiency. Thus from the point of view of the expert control system, the tool utilization has to be maximized by selection of appropriate machining parameters and coolant pressure in the machining zone.

It has been, earlier, stated that the coordinated action of the tool and the machine on the workpiece constitutes the machining operation. Thus the deep hole machining operation forms the domain of the expert control problem. The domain can be classified into Machining cycle and Machining process. The machining cycle involves all the operations that are to be carried out to perform a machining process successfully. Machining process is a component of machining cycle wherein the tool acts on the workpiece for metal removal. To build a comprehensive knowledge base, both the sequence of operations and the expertise of machining process have to be modeled. The sequence of operation involved in deep hole machining is obtained by observing the skilled operator carrying out machining on deep hole machining whereas the expertise on the actual machining process (cutting process) is hard to elicit from the operator or otherwise. This expertise is the real manifestation of experience in metal cutting in general and deep hole machining in particular. The knowledge obtained has to be coded into a form understandable to the inference engine for reasoning and initiating corrective action. The process of coding the acquired knowledge is called the **symbolic modelling** The knowledge engineer is responsible for symbolic modelling of Machining sequence and the machining process.

2.5 SYMBOLIC MODELLING OF DHMS

Once the domain of the problem is identified, then the domain has to be represented in such a way that the symbolic computation can be carried on. The process of representing the domain in the form of symbols is called symbolic modelling.

2.5.1 STRATEGY FOR SYMBOLIC MODELLING:

The domain of the problem, Deep hole machining system, is a complex set of functional and non functional components. A strategy is essential to achieve the symbolic modelling of such a complex system. The strategy thus adopted can be summed up as:

1. Domain decomposition
2. Identification of necessary elements of a DHMS from the control point of view.
3. Identification of the events which bring about changes in the conditions as stated above.
4. Identification of the conditions of these elements at a given instance of time or sequences.

5. Identification of the operation to reflect all of the above.

2.6 DOMAIN DECOMPOSITION:

A logically reconfigurable topology has to be chosen to meet the requirements. Logically reconfigurable means the elements of the decomposed dom… can be configured by the programmer to reflect the actual system requirement whenever the system demands the same. The entire problem domain, DHMS, has to be decomposed into sub groups and be represented in such a way that it reflect the real relation between various components of the DHMS in their hierarchy. To achieve this, the DHMS structure is represented in the form of a TREE or graph encompassing the most important character of such a representation namely, inheritance. The graph or tree representation of DHMS has the following relation.

$$A = (X^0, R^0)$$

in which the nodes $x_0^i \in x^0, (i = 1,\bar{n})$ represents different elements of DHMS. For every node we can formulate a separate control problem which should be oriented to achieve the general functioning of DHMS. Branch of the graph represents the parents-offspring relation namely Inheritance and Evolution. From above definitions it follows that symbolic presentation DHMS structure consists of the set of symbols corresponding to each node of a graph structure which reflects the relation of one node with other nodes and thus defines the **Symbolic model of DHMS Structure.**

2.7 ELEMENTS OF DHMS

DHMS is an assembly of various elements as shown in Figure 2.6. The coordinated actions of these elements in a proper sequence constitute the deep hole machining operation. A condition Parameter (CP) can be ascribed to each element of the system. Status of such an element or the status of a condition parameter can be defined as a functional value as follows:

0 - The element is not ready to carry-out its function.

1 - The element is ready to carry-out its function.

2 - The element is in abnormal condition or has failed.

Thus a complete description of an element can be represented symbolically in two fields the first field representing the CP and the second field representing the status. Complete symbolic representation of condition parameters is shown in Table 2.1.

2.8 EVENTS:

Various **events** constitute a machining cycle. The events can be 'instant' or 'interval'

Table 2.1: Condition Parameters for DHMS

#	Code	Description
1	ESC	Power supply condition
2	TEC	Tool Entry Condition w.r.t Starting Bush
3	CSC	Coolant Supply condition
4	RWC	Rotational Condition - Workpiece or Tool
5	PFC	Present Feed Condition
6	QRC	Quick Return Condition
7	FCC	Feed Change Condition
8	SCC	Speed Change Condition
9	CCS	Change Coolant Supply condition
10	IFC	Idle Feed Condition - Tool Head changing
11	CWC	Change Workpiece Condition
12	CBC	Change Boring Bar Condition

events. An event can be defined as a particular action by one of the elements of the DHMS. For example 'switching ON' the power supply to the deep hole drilling machine is an event. Since the process of switching on the power is almost instantaneous it is called an instant event. So an **Instant event** can be defined as an event wherein a condition parameter (CP) goes through a momentary change of value in its status for an insignificant amount of time. Table 2.2 lists instant events in a DHMS.

Once an instant event takes place, for example, switching ON the power supply to the DHMS, the power supply continues till the process is finished or until it is requested by the expert controller. This event of continuous power supply takes place for a finite period of time between two corresponding instant events. Such events are called interval events. **Interval events** can thus be defined as events which take place between two corresponding Instant events over a finite period of time. It is interesting to note that duration of every operation in DHMS is the time interval between two instant events. Also the duration of the process is the duration of interval events. Hence interval events can be defined as events during which the CP does not change its value. For example, the machining processes in DHMS can be described by interval events beginning with instant events STWF and ending with instant event STWS. A complete list of interval events are shown in Table 2.3.

Table 2.2: Instant Events for DHMS

#	Code	Description
1	SESS	Start power supply to control system
2	SETB	Start - tool head entry to starting bush
3	SFET	End- tool head entry to starting bush
4	SCCS	Start - coolant supply
5	SWTR	Start - Workpiece/tool rotation
6	STWF	Start - Tool/ Workpiece feed
7	SFRR	Allow - Feed rate control
8	SSRR	Allow - Speed control
9	SCSR	Start - Coolant supply control
10	SWTS	Stop - Workpiece/Tool rotation
11	STWS	Stop - Tool/Workpiece feed
12	STCS	Stop - Coolant supply
13	SQRB	Start - Quick return to initial position on S.B.
14	STQR	Stop - Quick return
15	SIFC	Start - Idle feed control
16	STIF	Stop - Idle feed control
17	SEST	Stop - Electric power supply.

Table 2.3: Interval Events

#	CODE	DESCRIPTION
1	PSD*	Power supplying to DHMS
2	TSB*	Tool entering to starting bush
3	CSD*	Coolant supplying to DHMS
4	RTW*	Rotating tool/workpiece
5	FTW*	Feeding tool/workpiece
6	QRT*	Quick Returning tool/workpiece
7	CST*	Changing speed tool/workpiece
8	CFT*	Changing feed tool/workpiece
9	CRT*	Changing flow rate of coolant
10	ICT*	Idling of the cutting tool
11	CWP*	Changing the workpiece
12	CBB*	Changing the boring bar

2.9 SEQUENCE OF EVENTS:

Events in a DHMS operation occur in a designed sequence, as illustrated in Figure 2.7, and hence at any given instance of time, information about some events which are completed, occurring or future events relative to given time interval can be reflected with the help of combined events. **Combined events** we define as events which take place only under conditions when some logical relation between events is valid. ("within the other events (IOE)" "Before the other event (BOE)" or "After the other event (AOE))".

For example the information about the changing of coolant supply conditions (events CRT*) allowed only during machining (events FTW*) defined by relations within the events (IOE) and can be written in logical form as:

$$\mathbf{CRT^* \cup FTW^* \cup IOE}$$

Identifiers (set of symbols reflecting the name of events) of instant, interval & combined events can have one of the three values:

'0' - Events will occur in the future

'1' - Events taking place currently

'2' - Events completed at this instance.

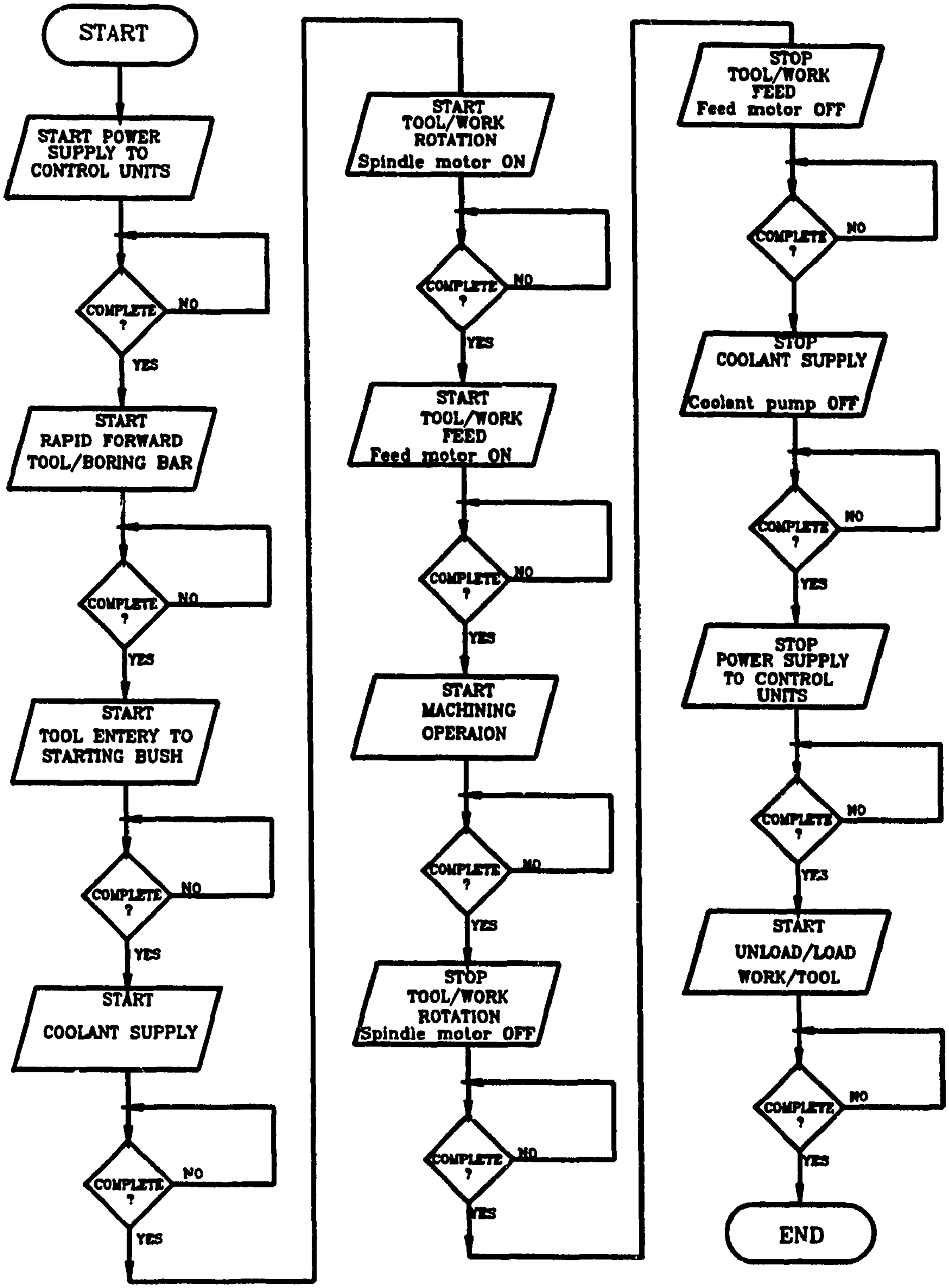


Figure 2.7: Sequence of DHMS operation.

Thus the **Symbolic model of DHMS Functions** we define as the set of Instant, Interval and Combined events. **Situation** in a symbolic model can be defined as a set of events. Each situation can be written in conjunctive normal form with the value of CP. For example the situation indicating the DHMS is ready to carryout machining consists of 6 (six) completed (Logical value '2') instant events:

SESS - Start power supply to DHMS

SETB - Start tool head entry to starting bush.

SFET - End - tool head entry to starting bush.

SCCS - Start - coolant supply.

SWTR - Start - Workpiece/tool rotation.

STWF - Start - tool/workpiece feed.

Description of Situation, DHMS is 'ready for machining' can be written as:

$$(SESS=2)\cup(SETB=2)\cup(SFET=2)\cup(SCCS=2)\cup(SWTR=2)\cup(STWF=2)$$

This situation provides the verification for the expert controller before starting the machining process. Further verification to decide the machine is 'ready for machining' is provided by the symbolic model of condition parameters(CP) as shown below:

$$(ESC=1)\cup(TEC=2)\cup(CSC=1)\cup(RWC=1)$$

Situation in DHMS which describe only one element of conjunction is called an

elementary situation. Situation with more than one element in conjunctive description is called a **complex situation**.

2.10 OPERATION

A complete and comprehensive representation of the deep hole machining operation is necessary for symbolic modelling and temporal reasoning. This comprehensive representation of the entire process enables to identify and model the relation between the various events and the identification of the beginning and ending of various events as represented by the beginning of the sequentially next event. Operation chart of the DHMS is shown in Figure 2.8.

2.11 ALGORITHMIC MANIPULATION OF EVENTS

2.11.1 REQUIREMENTS OF ALGORITHM AND DATA STRUCTURE

Having defined various events of deep hole machining process and identifying various elements involved in such a process in order to obtain a symbolic model of DHMS, attention should be focused on a suitable data structure and algorithm which enables successful implementation of such a model in a working expert controller. The main requirements of such an algorithm and data structure are:

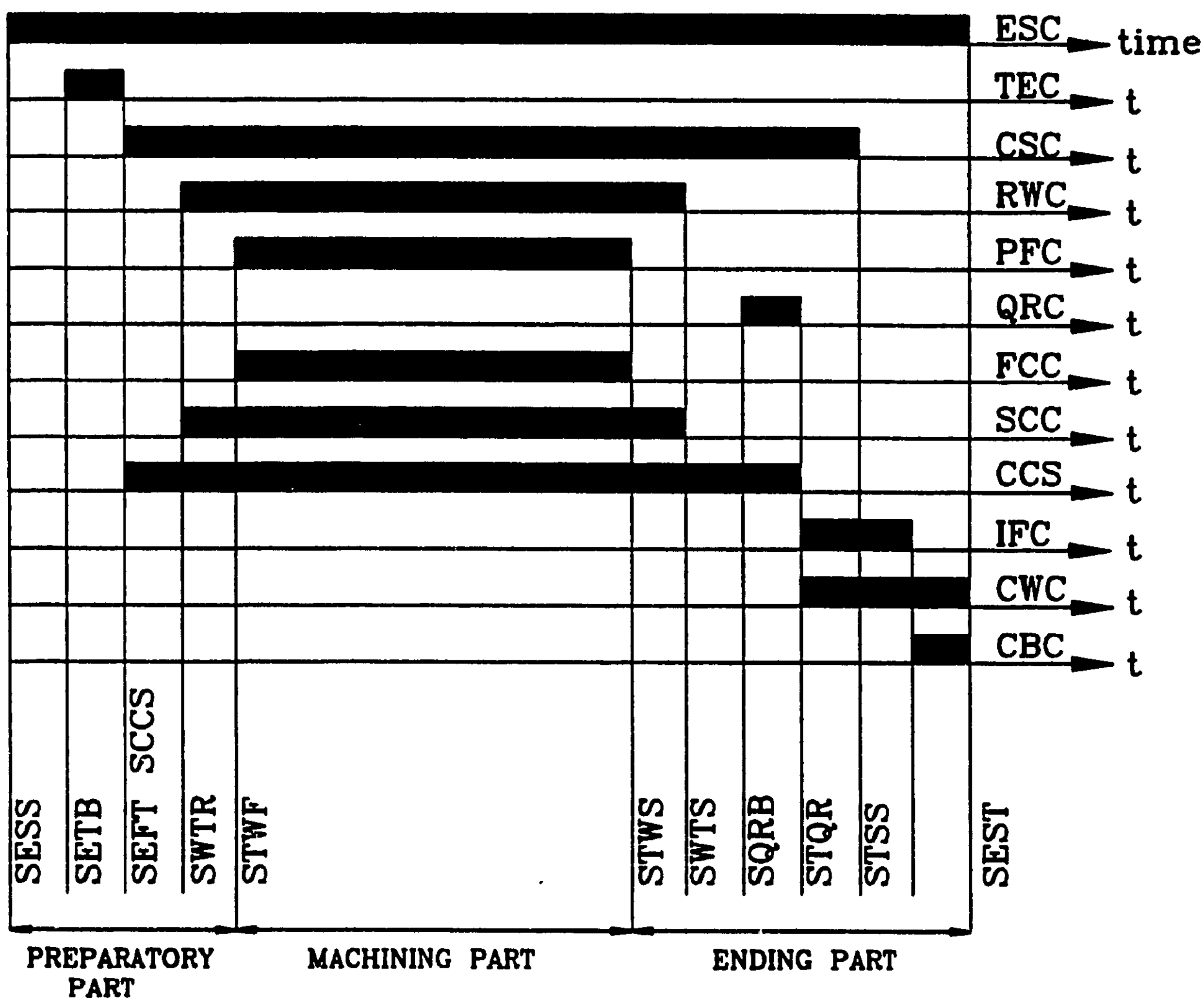


Figure 2.8: DHMS operation chart.

1. Robustness : The algorithm and the data structure should be robust to withstand continuous operation with simple instructions for adding and deleting items.

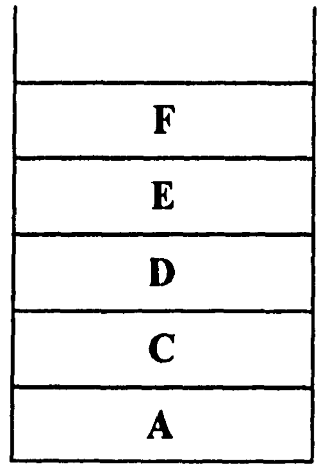
2. Speed: Since the algorithm becomes a part of the real time control system, faster the algorithm, better the suitability

3. Fool proof: Inherited capability to maintain the sequence of operations involved in DHMS and thus providing an in built safety jacket.

4. Dynamic: The program resulting from the algorithm should be memory efficient and hence be created and destroyed dynamically.

5. Simplicity: The concept should be simple, for simplicity is the greatest of all virtues.

A data structure which fulfils all of the requirements stated above is STACK . Stack is one of the most powerful concepts of computer science and is deceptively simple and very robust. A stack is an orderly collection of items into which a new item can be inserted and from which items may be deleted at only one end called the **top** of the stack. A stack can be pictured as shown below:



Unlike the most popular data structure called ARRAY, the definition of the stack provides for insertion and deletion of items so that a stack is a dynamic, constantly changing object. The Open end of the stack is distinguished as the top of the stack and in the figure above F is the top of the stack and any items added to the stack are placed on the top of F and if any item is to be deleted, F which is at the top is the first to be deleted as it is the only element accessible in the stack. This attribute of the stack is the most important of a stack, that the last element inserted into the stack is the first item deleted. Thus F is deleted before E because F is added after E and hence a stack is a last IN first OUT list. This is analogous to Lay-off practices employed in any organization. This property of the stack should be over emphasized as it is the property required to maintain strict sequence of operation of DHMS. For example the instant Event SESS (Power supply) is the FIRST item to be activated in the machining process and the LAST operation to be carried out and hence adopts 'First IN last Out' rule.

2.11.2-OPERATIONS ON A STACK:

Two operations are possible on a stack. When an item is added to a stack it is PUSHED onto the stack and when it is removed, its POPPED from the stack. Given a stack 's', and an item 'i', performing an operation 'push(s,i)' adds the item i to the top of the stack s. Similarly, the operation 'pop(s)' removes the top element and return it as a function value thus the assignment operation i= pop(s) removes the top element of s and assigns its value to i. Although 'push' operation can be applied to any stack 'pop' operation cannot be applied to the empty stack because such a stack has no elements to pop. Operation empty(s) determines whether or not a stack is empty. This gives an other important safety jacket for controlling DHMS operation which will be evident elsewhere in the thesis.

2.11.3 STACKS IN EXPERT CONTROLLER:

Three stacks 0, 1 and 2, are assigned for storing the condition parameters (CPs) during various stages of machining. Each stack representing the status of such a parameter respectively. Three stacks, I0, I1 and I2 are assigned for instant events. Letter part of the name representing instant event while integer part representing the status. Three stacks are assigned for Interval events. They are named 'Y' 'T' and 'F' representing 'Yet to take place', 'operation is Taking place' and ' operation is Finished' respectively. Figures 2.9, 2.10 and 2.11 illustrate the stacks before, during and at the end of machining operation. The algorithm is activated at the START command from the expert controller and the END command will reset the stacks to its initial state. This unique way of representing the different events gives the advantage of cross checking different stacks to see if the sequence of operations is properly done and also provides a powerful safety lock from common mistakes which could have been included in the knowledge base. For example, consider faulty knowledge base with 'quick return' command in the middle of machining process. With stack implementation this mistake will not result in the quick return instead the expert controller indicates a faulty knowledge coding to the user. This is because of the inherited property of stacks not allowing any random addition or deletion of items from it making this a fool proof system. Figure 2.12 illustrates the data flow in sequence control using stacks.

2.12 PROCESS KNOWLEDGE:

Knowledge of the machining process (metal cutting) forms the core of the knowledge base as it forms the frame work for the decision making during actual metal cutting operation in deep hole machining operation. In order to evaluate and control this machining process knowledge it is essential to identify the condition of the process

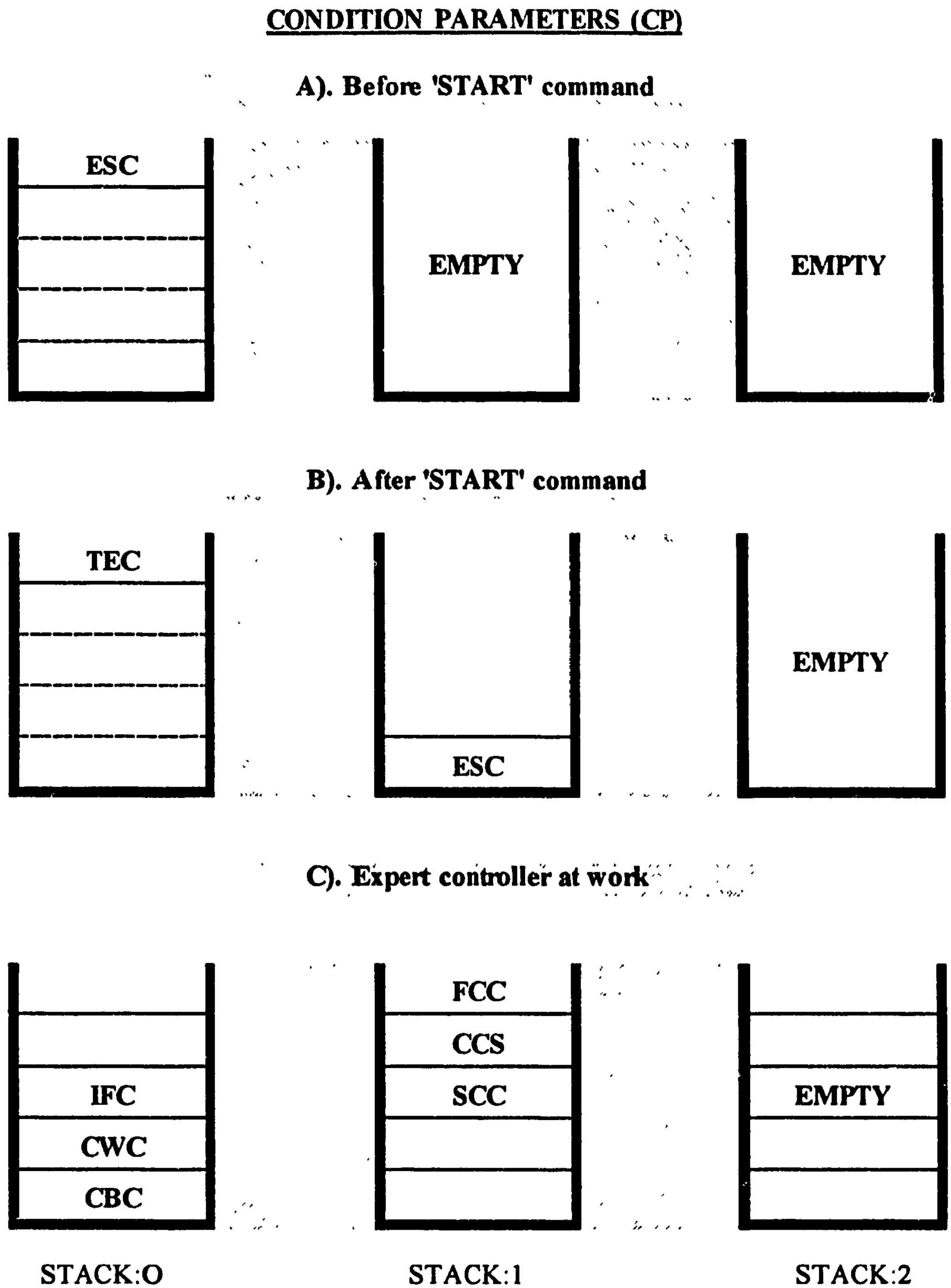


Figure 2.9: Stack implementation of CPs

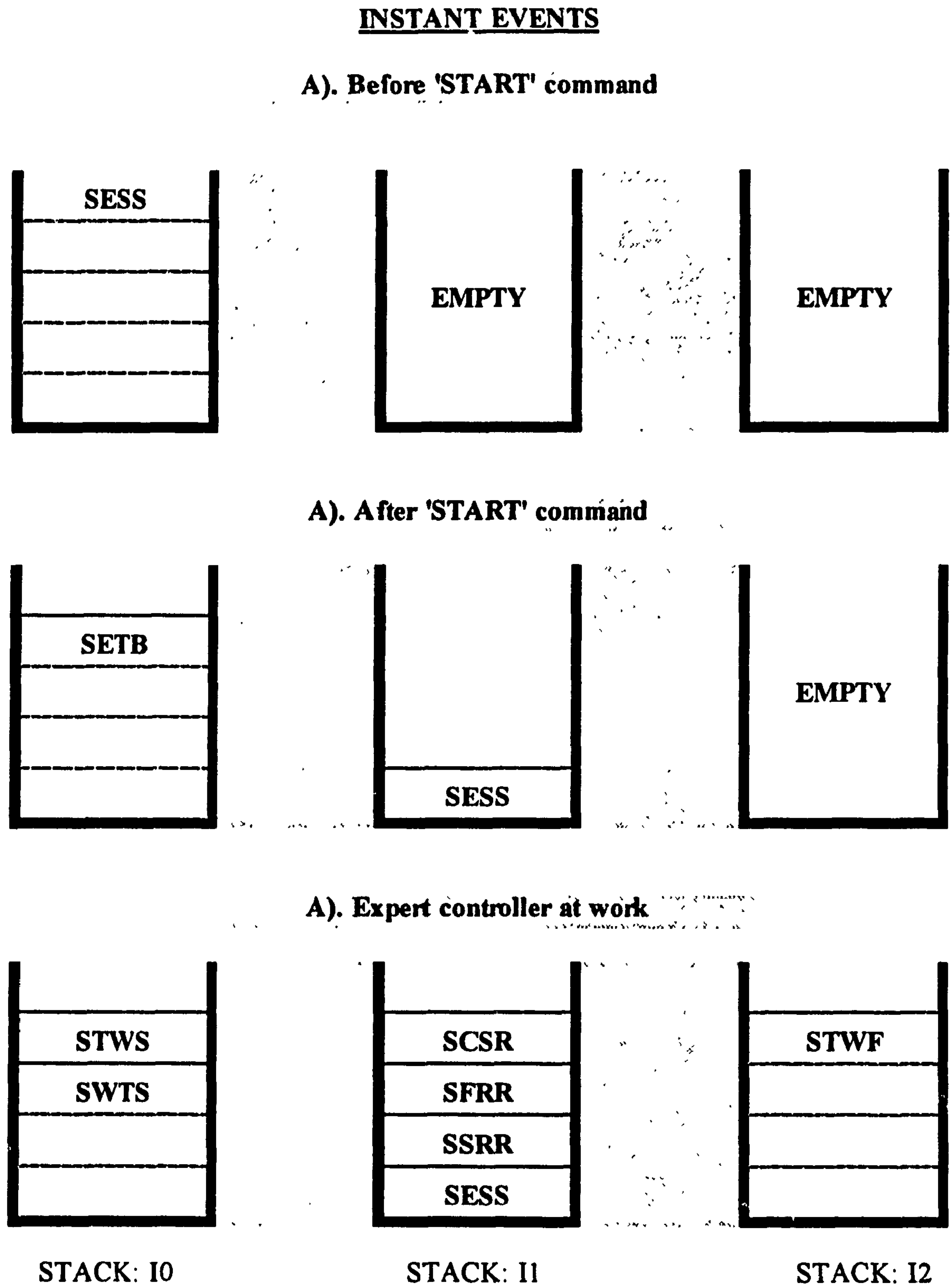


Figure 2.10: Stack implementation of instant events

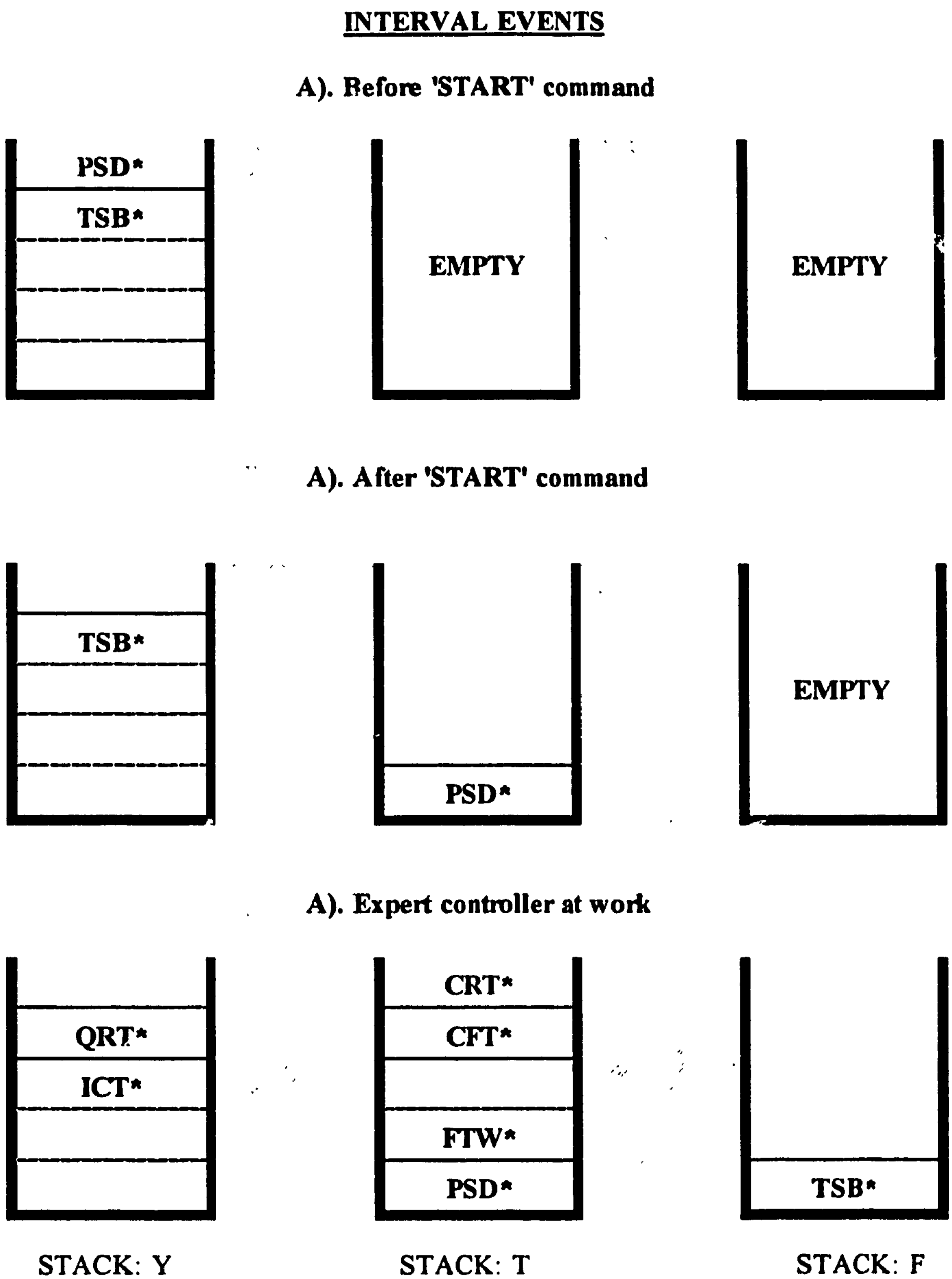


Figure 2.11: Stack implementation of interval events.

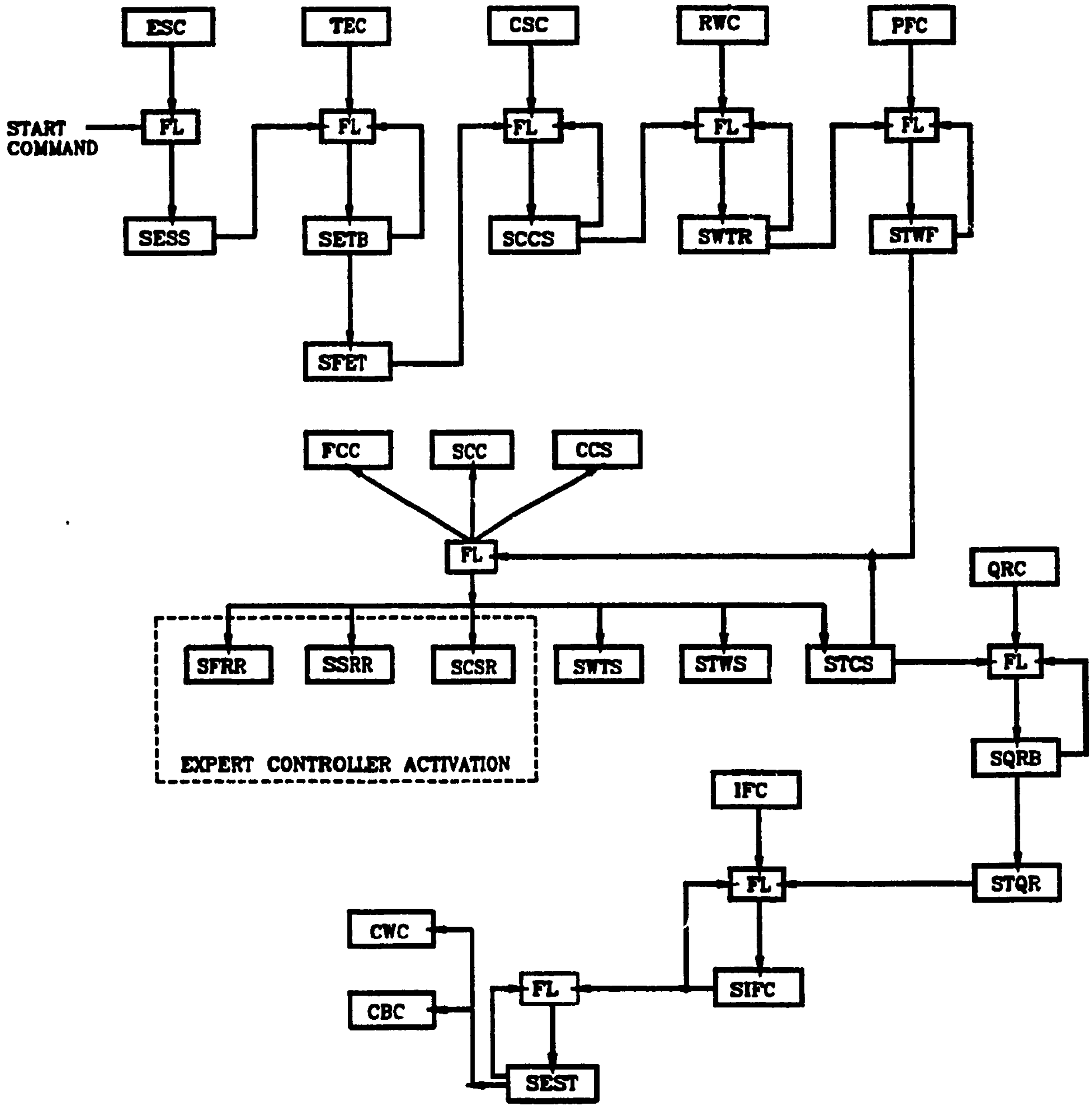


Figure 2.12: Data flow in sequence control.

and then initiate corrective action, when required, to continue the process according to the criterion defined for such a control. The identification of the process condition is provided by signals from the DHMS and these signals are transformed into symbols for decision making by the expert controller. The machining responses which describes the condition of the machining process at any given time are referred to as the *process descriptors*. The knowledge required for decision making, when a description of the process is provided by the descriptors, constitute the *process expertise*.

2.12.1 PROCESSOR DESCRIPTORS:

The metal cutting process in deep hole machining results in various machining responses. These machining responses manifest themselves in different forms such as torque, thrust, vibration of the tool-boring bar assembly, fluctuations of spindle motor speed, fluctuation of current drawn by the spindle and feed motor, acoustic emission from the cutting zone, and heat generated at various points on the machine etc [Owen, 1993]. Even though all of these descriptors enable to determine the machining condition at a given instance, it is the goal of the knowledge engineer to select the appropriate descriptors which are necessary and sufficient to describe the process. This criterion of minimum number of descriptors which are necessary and sufficient for the description of machining process should enable decision making without redundancy.

Peter Streicher [1975], Chandrashekhar [1984] and Griffiths [1982], have studied the various machining responses in deep hole machining operation and have suggested that the 'Torque' and 'Thrust' in deep hole machining has a direct correlation with the machining condition. The results published by these researchers, with respect to torque and thrust in deep hole machining can be summed up as follows:

1. Amplitude of both Torque and Thrust increase when the machining process is unstable.
2. Torque and Thrust increases with tool wear
3. Tool failure is indicated by sudden increase and decrease of the amplitudes of torque and thrust.
4. Torque and thrust describe various phases of deep hole machining operation.
5. Torque and thrust levels can be controlled by appropriate cutting speed and feed rates. etc.,

Thus it can be concluded that the torque and thrust are very descriptors of the process condition. However, torque or thrust alone cannot be used for determining condition. Griffiths [1982] and Peter Streicher [1975] have shown that at the initial stages of machining, when the guide pads enter the workpiece, the amplitude of torque increases whereas the amplitude of the thrust force does not show any significant change. Thus the increase in the amplitude of the torque with no change in the amplitude of the thrust force indicates only the entry of guide pads into the workpiece and not any unstable machining or tool failure. Moreover, When crater wear on the tool is high the tool may have effective positive rake angle resulting in lower torque levels. This situation is described by observing thrust levels at the same time. Thrust levels cannot be used alone as the descriptor as the rise and fall in the amplitude are smaller than that of torque levels and hence is difficult to monitor continuously.

Additional descriptors are required to verify the ease of chip removal, swarf blockage etc. These are secondary descriptors as they do not describe the actual condition in the machining zone. Lukic [1987] has demonstrated that the power spectrum of the swarf impinging on a chip sensor, can be used to classify the chips into various groups of sizes. Thus the chip factor obtain by the power spectrum indicates the size of the chip. Smaller the size of the chip easier the chip removal. The blockage of swarf can be red by absence of coolant flow into the machining zone as the chips blocking the swarf flow break the coolant circuit.

Thus, it is imperative that the following necessary and sufficient parameters can be used to determine the machining operation in deep hole machining:

FO - Axial Drilling Force

FF - Amplitude of Axial Drilling Force

MT - Drilling Torque

MF - Amplitude of Drilling Torque and

CF - Chip Factor.

2.12.2 SENSING THE PROCESS DESCRIPTORS:

Force: Strain gages and piezoelectric sensors are the traditional force monitors. The force sensors can be located in such a way that they bear the full cutting load without significantly affecting the machine stiffness. The resolution is high enough to monitor tool breakage and wear even on finishing operations. The piezoelectric force sensors passes on 90% of the force flow with no loss of static or dynamic machine stiffness [Owens, 1993].

Previous research [Griffiths, 1982] on gun drilling shows that the axial force component of the cutting force is a very good indicator of machining process. Amplitude signature of the axial force reflects the stability of the machining process and tool failure. In increasing amplitude of the axial cutting force towards the upper limit indicates the instability and impending tool failure, enabling the controller to take corrective action to prolong the tool life by bringing back the machining process into stability. When the tool fails or wears out, amplitude of the signal overshoots the upper limit. The ease and effectiveness of the axial force signature makes is most suitable for monitoring deep hole drilling process.

Torque: The piezoelectric force sensor used for force sensing can also be used for monitoring the drilling torque. Together with the axial force signals, the torque signals can be effectively used to classify tool entry, tool exit, tool failure and instability in the machining condition. Since the axial force and the drilling torque can be sensed using the same sensor, it is more economical and more efficient for monitoring the deep hole drilling operation.

Chip sensor: A design of chip sensor is illustrated in Figure 2.13. The swarf coming out of the boring bar is directed on to a thin metallic membrane. The strain gauges on these membrane are connected to a power amplifier and then to the data acquisition system. The FFT of the signals reveal the chip factor(CF) [Lukic, 1987].

2.12.3 PROCESS EXPERTISE

2.12.3-1. PRODUCTION RULES:

Production rules or rule-based systems represent the underlying technology for many of today's artificial intelligence (AI) systems. Emil Post[1936], whose primary goal was to study computability in formal systems or logic, originated the work on production systems. The post production allowed for rewriting of expressions where a rewrite corresponds to forming a conclusion from a set of premises. Later it was

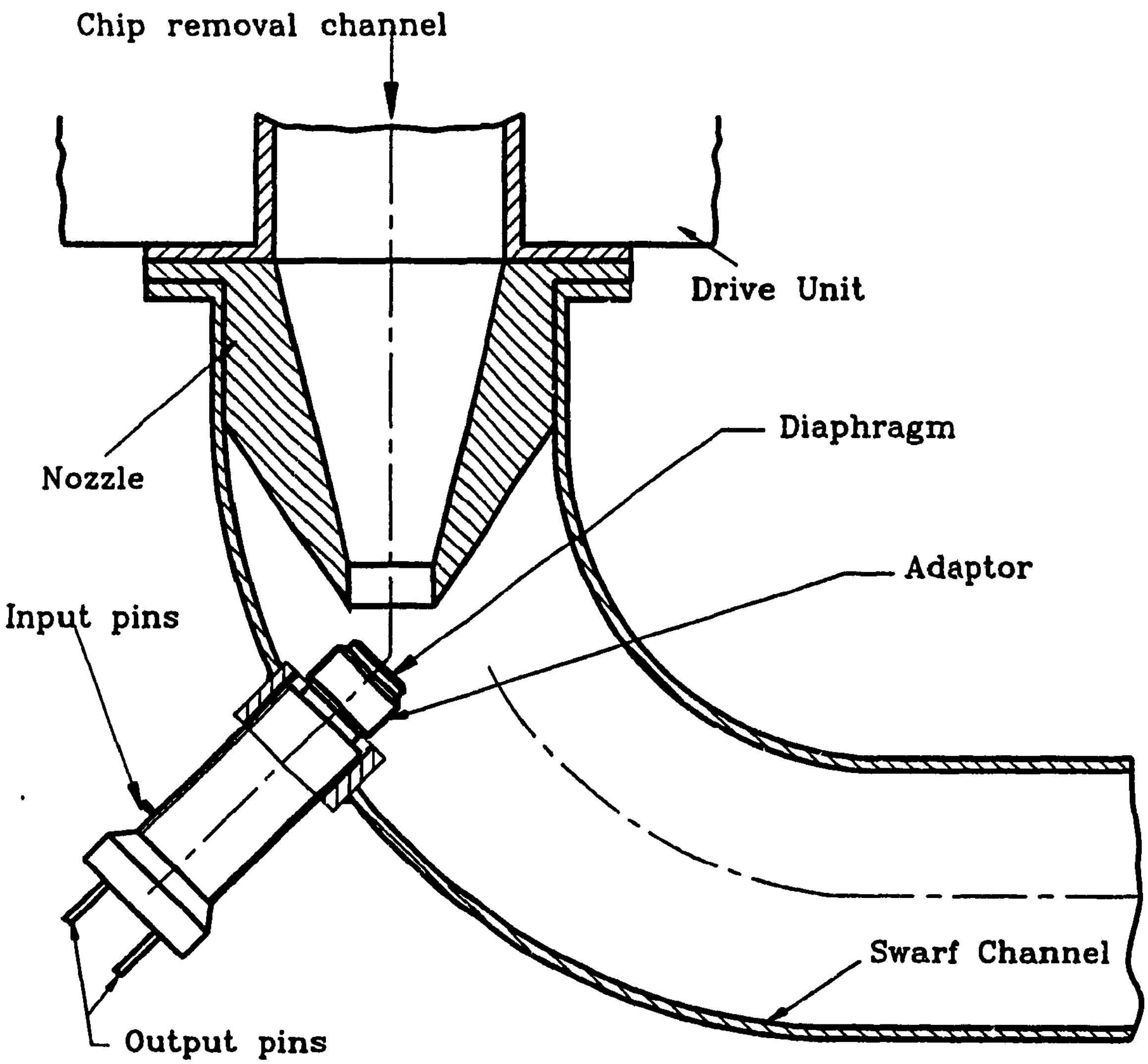


Figure 2.13: Chip sensor.

shown that production systems were equivalent in power to Turing machines[2]. Although current production systems employed in AI are considerable different from Post's original notions, the fundamental ides, i.e., the notion of **premise-conclusion** pairs, remains the same.

Rules have been used extensively in AI because rules are simple to work with and because each rule can be considered independent of the others. The latter fact allows for incremental construction of AI programs. Many believe that rules closely represent the way human beings deal with day-to-day problems. In fact, production rules were used with moderate success in the modelling of simple cognitive skills [Newell and Simon 1972] of humans.

Production rules, or IF-THEN rules, are statements of the form:

PREMISE → CONCLUSION

where PREMISE determines the conditions or situations that must be satisfied for the rule to be applicable and CONCLUSION is the action(s) that must be taken once the rule is applied. For simplicity, rules are written a more English-like manner by using the construct IF..... THEN... form. For example, the following are a few of the rules governing the expert control of deep hole machining system.

[2]The concept of Turing machines was introduced by A. Turing as a simple model of a computing device. Despite their simplicity, Turing machines are considered to have the capabilities of today's general-purpose computer concept.

2.12.3-2 FORMATION OF PREMISE:

Five parameters have been identified for the complete description of the machining conditions. These five parameters provides the necessary and sufficient conditions for decision making by the expert controller. Thus these five parameters should represent the premise of the rule. In order to completely define the premise, the signals have to be transformed into symbols. The status of these parameters, according to the signals provided by the sensors, is classified into 'Normal', 'Less than Normal' and 'Above Normal' conditions These conditions in the form of logical conjectives forms the **symbolic models of verification**. Symbolic models of Verification of operating conditions of DHMS provides a tool to realize the status of controlling parameters. Status of each controlled parameter is assigned three logical values for model representation.

'1' - The value of controlled paramcter less than normal.

'2' - The value of controlled parameter is normal.

'3' - The value of controlled parameter exceeds normal.

The interval between the lower bound and the upper bound values which determine the logical values of controlling parameters is referred to as **Tolerance intervals**. Table 2.4 is a listing of tolerance intervals for various controlling parameters as defined by

Table 2.4: Expert Control Table

Controlling parameter	Unit	Less than normal	Normal	Greater than Normal
FO	N			
FF	N			
MT	NM			
MF	NM			
CF	Piece/min.			

the experts. Each of these five parameters along with their status forms the symbolic model of the machining condition representing a complex situation. The knowledge base has, thus, to accommodate (5!*3!÷3) = 240 non identical and 3 identical and hence a total of 243 complex situations to describe the deep hole machining operation in its entirety. These situations enable the expert controller to establish the condition of the DHMS for further diagnosis and to initiate control actions. These premises are supplied to the experts for their advise on the action for each situation explained by the premises. The conclusion or the actions provided by the experts along with the premise form a complete rule. This complete rule is termed **Symbolic diagnostic model** of DHMS.

Control symbolic model of DHMS is a set of control decision which corresponds to certain situation. These models can be referred to as procedural rules. These models provide the necessary control actions when the symbolic diagnosis models finish their diagnosis. Thus it is evident that the premise of a production rule is the diagnosis provided by the symbolic diagnosis model. Whereas the conclusion part of the production rule is provided by the control symbolic model. A complete set of symbolic diagnosis models together with all the Control symbolic models, the structure and the function defines the **symbolic model of DHMS**. It should be mentioned that information set of every particular model contain some common elements. Therefore these sets intersect and the formal determination of each model can be done only by symbolic language. The formal structure of the symbolic model is shown in the Figure 2.14. A complete of listing of all the rules and symbolic models are provided in Appendix 1.

2.13. INFERENCE ENGINE

2.13.1. INFERENCE MECHANISM

A simple forward chaining alg..rithm is used for inferencing. The basic function of the inference engine can be explained in 3 simple steps:

a. Prove a Rule : Get the current situation from data acquisition system, match the current situation with the premise clause of the rule, if the current situation matches the premise of the rule, activate the action which the conclusion of the rule. Otherwise go to **b**.

b. Prove another Rule: Repeat **a** and **b** for the next rule in the knowledge base until all the rules in the knowledge base is tried.

c. Resolve Conflict : If more than one rule matches, then resolve conflict. go to **a**.

Forward chaining mechanism is illustrated in Figure 1.18 (Chapter 1).

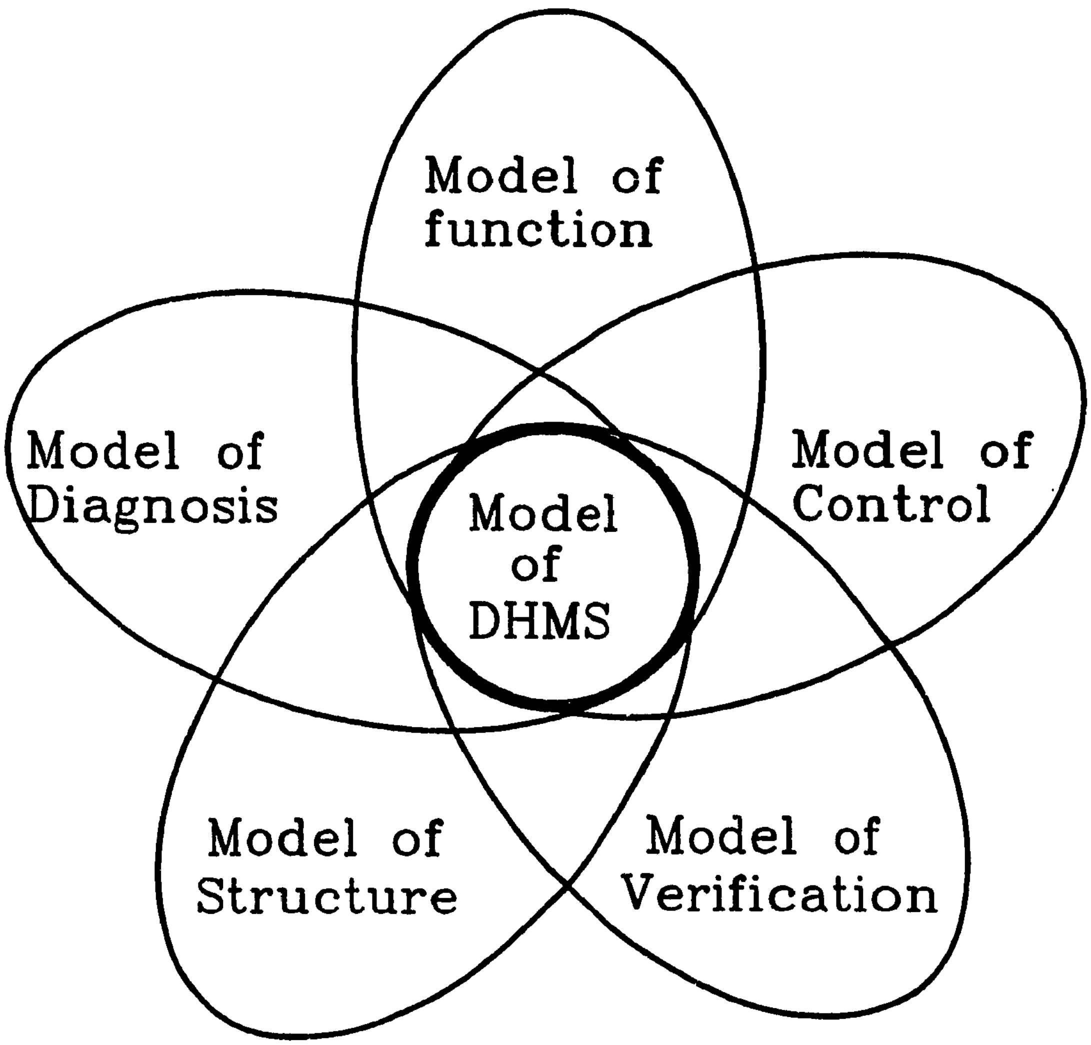


Figure 2.14: Structure of DHMS symbolic model.

2.13.2 IMPLEMENTATION OF FORWARD CHAINING:

Simplicity in appearance of the forward chaining algorithm is deceptive. Actual implementation of this algorithm with conflict resolution in a real time system is extremely complicated and beyond the scope of this theses. However a brief comprehensive implementation methodology of forward chaining mechanism is described.

Stack, as explained earlier, forms the basic building block of this algorithm. Two stacks are defined for this process. The first stack is a repository of all the rules and follows 'Last In First Out' principle. The second stack , following the same principle, contains the rules which have been successfully activated before reaching the current situation. The second stack gives an important reference for the current situation and is often visited by the inference mechanism either for conflict resolution or as requested by some rules which suggest an action based on previous rule premises. The second stack forms the history of the machining till the present time and its importance should be emphasized. The contents of the second stack, or the history of the process so far, is extremely useful in anticipating the future situation and hence the system can forecast the future and initiate necessary actions. It is also important to note that tagging all the rules which have visited the second stack will indicate the frequency of firing or activation of rules. This knowledge is used in the learning cycle to reordering the knowledge base sequence and hence making it more efficient with respect to execution time over a period of time or training cycle.

Utilization of stacks takes away a fixed amount of storage even when the structures are using a smaller amount or possible not storage at all, further, no more than that fixed amount of storage may be allocated thus introducing the possibility of overflow. If the expert control system uses two stacks for rules in two separate arrays one for the rules and one for the rules which are successfully fired as is normally used in general purpose expert systems, each of these array has to have space for all the rules i.e., 243 rules. Despite the availability of space for 243 rules, neither can grow beyond 243 items. Consider an example wherein several rules have to be fired more then once in a deep hole machining operation. Then there is a possibility that the second stack containing the fired rules in sequential form may have to grow beyond 243 elements. There is no way of determining the maximum size of the stack required in such an operation. One solution to this problem is ordering the items of the second stack in an explicit manner wherein each element contains the address of the next element without being bounded by any dimensions. Such a structure is called linked list and is a powerful dynamic data structure used by computer scientists [Tenenbaum et al, 1990]. Combination of stacks and linked lists or representing the stacks in the form of linked lists is the only way out of this infinite storage requirement problem. Hence the inference engine utilizes stacks represented in the form of linked lists.

2.14 CONFLICT RESOLUTION:

Expert knowledge is subjective as it is the manifestation of their understanding of the facts in suggesting the goals. Experts are known for their difference of opinion and some times for a given set of facts the goals suggested by different experts may be different or in some cases opposite. This results in multiple rules or the same facts. In other words more than one rule matches the current facts. The knowledge engineer has design an inferencing system which is capable of resolving such conflicts. For example consider the following set of rules formed by the knowledge engineer after eliciting knowledge by the experts:

EXPERT 1	EXPERT 2
IF FO = NORMAL FF = NORMAL MT = NORMAL MF= NORMAL CF = NORMAL	**IF** FO = NORMAL FF = NORMAL MT = NORMAL MF= NORMAL CF = NORMAL
THEN: INCREASE CUTTING SPEED	**THEN:** INCREASE CUTTING SPEED & INCREASE FEED RATE

EXPERT 1	EXPERT 2
IF FO = NORMAL FF = GRT. THAN NORMAL MT = NORMAL MF= GRT. THAN NORMAL CF =LESS THAN NORMAL	**IF** FO = NORMAL FF = GRT. THAN NORMAL MT = NORMAL MF= GRT. THAN NORMAL CF = LESS THAN NORMAL
THEN: DECREASE CUTTING SPEED & DECREASE FEED RATE.	**THEN:** DECREASE FEED RATE & ANALYSE PREV. SITUATION.

When multiple rules match the facts or current situation of machining, conflict arises in selecting the rule to be activated. In order to implement conflict resolution it is necessary to view the inference mechanism, forward chaining in this case, as a three step cycle:

1. **Match:** Match all the rules against the fact base and determine the successful matches.
2. **Resolve Conflicts:** If more than one rule matches the fact or the current situation, choose either one, some, or all of the matching rules based on a conflict resolution strategy.
3. **Act:** Add the conclusion(s) to selected rule(s) to the action queue.

2.14.1 CONFLICT RESOLUTION STRATEGY

Designing and developing a conflict resolution strategy is one of the most important and difficult task of an expert system builder. In case of expert controller for DHMS it is imperative that a conflict resolution strategy thus developed should converge to a decision within a given time as required by the real time controller. With this constraint in mind the following strategy is adopted.

1. DO ONE:

Choose the first rule that matches the current situation. If the goal of the expert controller is successfully achieved then keep the rule in the position. Otherwise move the rule to the bottom of the group. Now the second rule becomes the first rule. Continue the same till current situation match the premise of another rule in the knowledge base.

2.DO ALL:

Apply all the matching rules in one Batch. For example,

rule #L

IF(FO = X, FF = X, M=X, MF=Y, CF=Z) THEN (Increase V) ....expert 1

rule #M

IF(FO =X, FF=X, M=X, MF=Y, CF=Z) THEN (Increase S) ....expert 2 then this conflict resolution method increases both V and S till a new situation is reached so that the control action can be handled by other rules some where down the sequence.

3. DO IN SEQUENCE:

Apply all the matching rules one by one in sequence so that the new current situation proved by a rule can be used to establish the premise of the rules else where in the rule base.

4.DO THE MOST RECENT:

In this method, the current set of premise is compared with the last rule activated which resulted in generating the current situation. The difference in the premise explains the current situation or fact then look for a rule which will act on those premise which made the difference. For example, rules matching the current situation:

IF(FO=X, FF=Y, M=X,MF=X,CF=X) THEN (Reduce S, Increase V) ...expert 1

IF(FO=X, FF=Y, M=X,MF=X,CF=X) THEN (Reduce S, Reduce V) ...expert 2

Rule which generated this situation (Previous rule)

IF(FO=X, FF=X, M=X, MF=X,CF=X) THEN (Increase S, Increase V)

comparison of the rule by expert 1 with the previous rule suggests that change in situation (FF=Y from FF=X) is a direct consequence of 'increase S', Hence the most suitable rule would be to neutralize this effect which is given by the first rule (expert 1).

In order to apply this conflict resolution methodology:

- The rules have to be grouped into class representing identical premise.
- Rank the groups according to the priority of decision required (1, 2 or 3).
- Within such a group assign ZERO confidence for all the rules.
- Increase the confidence level of a rule which has been successfully activated. This will enable the elimination of some of the rules, after a long run, if their confidence remains ZERO.

A flow chart illustrating the application of this conflict resolution strategy is shown in Figure 2.15.

2.15 DATA FLOW

The flow of data from the DHMS and back to the actuators is illustrated in Figure 2.16. The signals from the sensors are converted to symbols and then these symbols are processed by the meta rules. The meta rules decide whether high speed decisions are required for the present situation described by the symbols. If the situation does not describe an emergency then the rule base will search for a matching rule to be

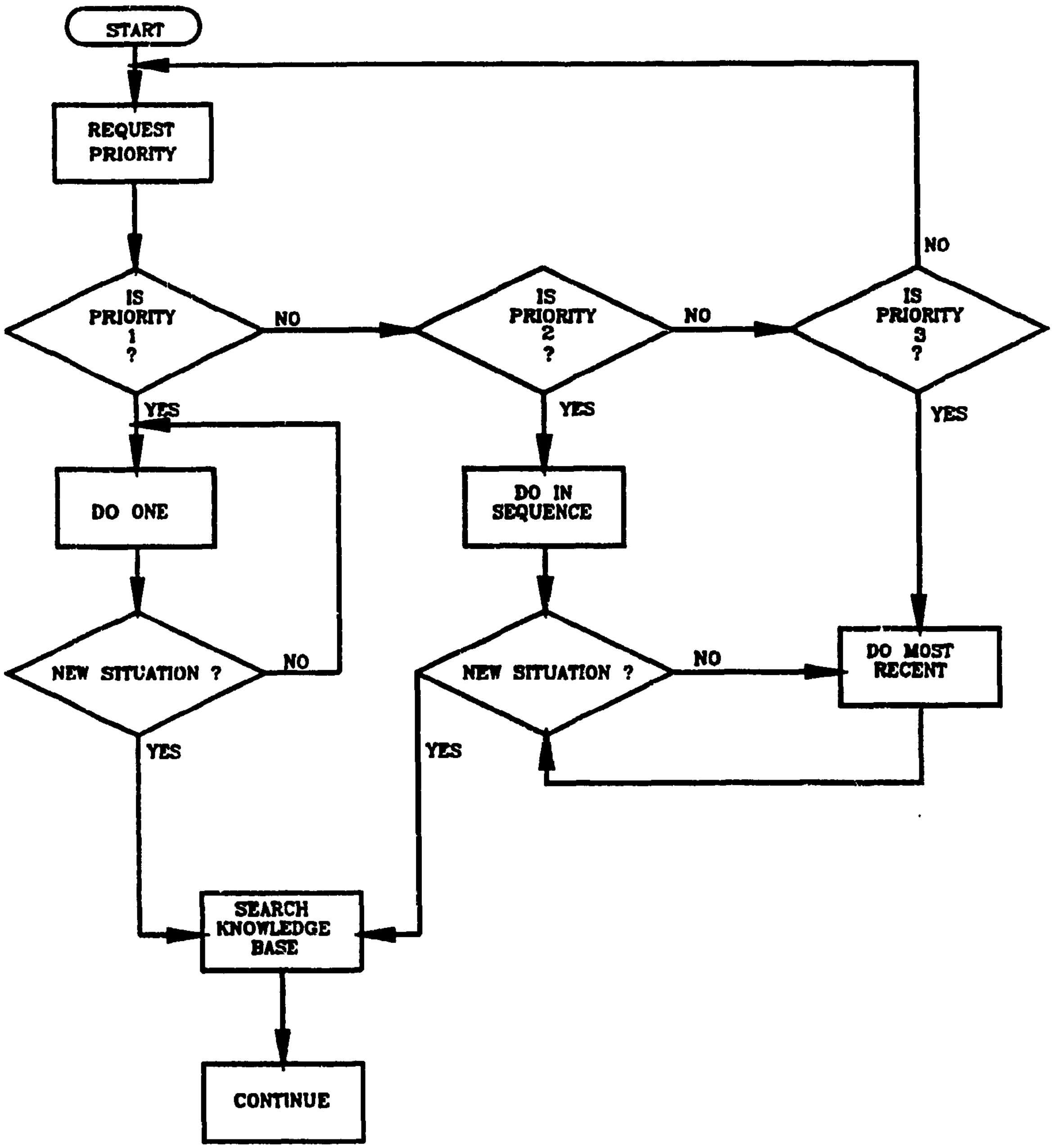


Figure 2.15: Conflict resolution strategy.

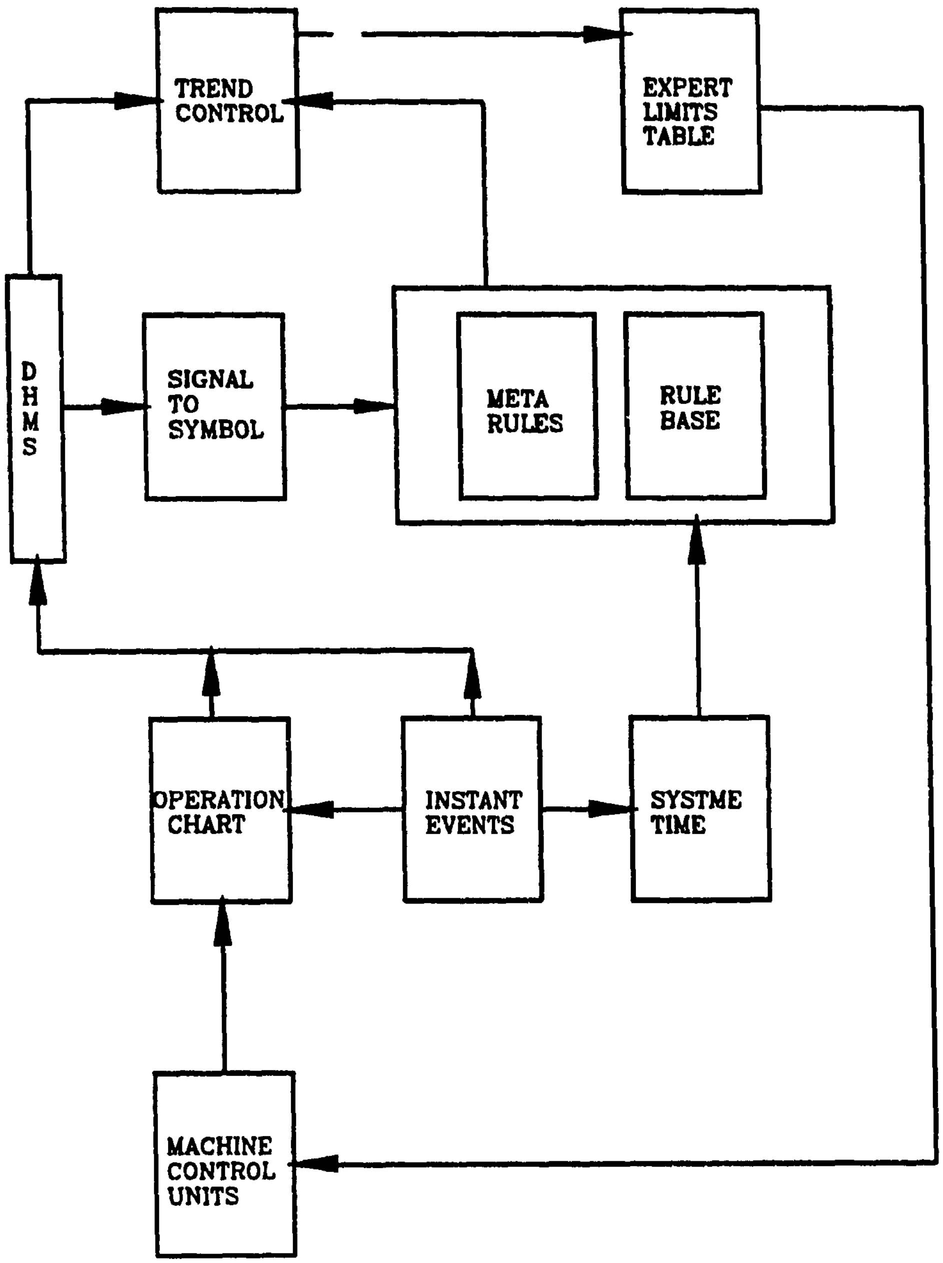


Figure 2.16: Data flow in the expert control of DHMS.

fired. The system time block will time stamp the rule. Depending on the time stamp on the rule and request for trend control from the expert system the trend control block calculates the future values of cutting speed and feed. The time stamp provides the details of current situation in the operation chart and the instant event at that instant. The values from the trend control block are converted to output voltages and fed into the machine control unit. In case of action by the rules alone, the parameters are varied until a new situation is reported by the expert system. The further action depends on the new situation.

2.16 TREND CONTROL BLOCK (TCB)

The activity of the trend control block can be considered as a forecasting method. The trend control block is used to calculate the future values of the cutting speed and feed etc to be implemented in order to obtained the expected machining response by utilizing various mathematical and statistical knowledge of the process.

2.15.1 GENERAL APPROACH TO TREND CONTROL

For a given system of data (x_i, y_i), $i=1,n$ where x_i is the system time and y_i is the current value of controlled parameter and n is the number of elementary controlling acts defined as tendency of changing controlled parameter as:

$$y = f(x_i, a_1, \ldots, a_m) \tag{2.1}$$

where a_1, a_m are constants.

It is necessary to define a forecast value y_F at a given moment of time x_F.

2.16.2 THE SOLUTION OF TREND CONTROL PROBLEM

The data base of the expert controller includes the mathematical or statistical knowledge about the process. These are generally in the form empirical mathematical models. Let us consider the models as shown below.

$$y = ax + c \tag{2.2}$$

$$y = ax^b + c \tag{2.3}$$

$$y = ae^{bx} + c \tag{2.4}$$

$$y = \left\{ \frac{1}{(ax+b)} \right\} + c \tag{2.5}$$

the solution of trend control problem can be obtained by proceeding along the following steps:

1. Define preliminary value of constant a_i; $i=1,3$ in the equations (2.2)

through (2.5).

2. Define secondary a_i, i=1,3

3. Choose the appropriate mathematical model for (2.2)-(2.5).

4. Calculate the forecast value of the controlled parameter.

If the data (x_i,y_i); i=1,n does not have an error (errors of measurement, acquisition, transmission etc) and chosen mathematical model precisely reflect the variation of controlled parameter, then we can find the constant a_i; i=1,n in (1) from the solution of the following:

$$\begin{aligned} y_1 &= f(x_1, a_1, \ldots, a_m) \\ y_2 &= f(x_2; a_1, \ldots, a_m) \\ &\ldots\ldots\ldots\ldots \\ y_m &= f(x_m; a_1, \ldots, a_m) \end{aligned} \qquad (2.6)$$

but in practice the system (2.6) due to errors of measurements, transmission,acquisition and due to the difference between the reality and the mathematical model the system (2.6) can be indeterminate.

The preliminary values of a,b,c can be found for the couple (x_i,y_i) under two elementary act of controlling n=1 and n=m. Substituting the corresponding values off (x_i,y_i) i= 1,..s, in the model (2.2) - (2.5) we have for every model the system two (three) equation and by solving we get a,b & c. For example, for linear model (2.2) we have

$$a = (y_1 - c) \;/\; x_1,$$
$$c = \left(y_1 y_m - y_{m x_1}\right) \;/\; \left(x_m - x_1\right) \qquad (2.7)$$

From model (2.3)

$$c = \left(y_1 y_m - y_s^2\right) / \left(y_1 + y_m - 2y_s\right)$$
$$b = \frac{\log\left[(y_m - c)/(y_1 - c)\right]}{\log(x_m/x_1)} \qquad (2.8)$$
$$a = (y_1 - c)/x_1^2$$

from equation (2.4)

$$c = \left(y_1 y_m - y_s^2\right) / \left(y_1 + y_m - 2y_s\right)$$
$$b = \frac{\log\left[(y_m - c)/(y_1 - c)\right]}{\log(x_m - x_1)} \qquad (2.9)$$
$$a = (y_1 - c)/e^{b x_1}$$

from equation (2.5)

$$c = \frac{(y_s(y_1 + y_m) - 2y_1 y_m)}{(2y_s - y_1 - y_m)}$$
$$b = \frac{1}{(y_m - c) - a x_m} \qquad (2.10)$$
$$a = \left(\frac{1}{y_1 - c}\right) - \left(\frac{1}{(y_m - c)}\right)$$

For defining the secondary value of the parameter it is necessary to calculate the residual in (2.2)

$$r_i^{(j)}=y_i-f\left(x_i;a^{(j)},b^{(j)},c^{(j)}\right),i=1,n, \tag{2.11}$$

where j represents the order of approximation. The secondary values of parameters can be represented as

$$a^{(1)}=a(0)+\alpha \tag{2.12}$$

$$b^{(1)}=b^{(0)}+\beta \tag{2.13}$$

where $a^{(0)}$, $b^{(0)}$ are the preliminary values and $a^{(1)}$, $b^{(1)}$ are the secondary values. α and β are the corrections.

The value of $c^{(0)}$ corresponds to the value of parameter before the machining of new work piece. This value changes after machining of a new workpiece and up dated in the database.

Substituting (2.12) - (2.13) in (2.1) we have:

$$y=f(X;a\pm\alpha,b+\beta,c) \tag{2.14}$$

Expanding the equation (2.14) using Taylors series w.r.t. corrections α and β and leaving only the first ordered terms we get

$$y = f(x;a,b,c) + f_a^1(x;a,b,c)\alpha + f_b^1(x;a,b,c)\beta \tag{2.15}$$

substituting;

$$\begin{aligned} f(x_i;a,b,c) &= f_i \\ f_a^1(x_i;a,b,c) &= \left(f_a^1\right)_i \\ f_b^1(x_i;a,b,c) &= \left(f_b^1\right)_i \end{aligned} \tag{2.16}$$

we have,

$$y_i = f_i + \left(f_a^1\right)\alpha + \left(f_b^1\right)\beta \tag{2.17}$$

substituting equation (2.17) in (2.6)

$$\begin{aligned} y_1 &= f_1 + \left(f_a^1\right)_1\alpha + \left(f_b^1\right)_1\beta \\ y_2 &= f_2 + \left(f_a^1\right)_2\alpha + \left(f_b^1\right)_2\beta \\ &\cdots\cdots\cdots\cdots \\ y_n &= f_n + \left(f_a^1\right)_n\alpha + \left(f_b^1\right)_n\beta \end{aligned} \tag{2.18}$$

In general case if n > 2 system (2.18) is indeterminate. This type of system known as conditional system and α and β can be defined by the method of Least Squares

Least Squares method can be stated as the procedure of m.nimizing the equation below.

$$S(\alpha,\beta)=\sum_{i=1}^{n}\left[f(x_i;a+\alpha,b+\beta,c)-y_i\right]^2 \tag{2.19}$$

or;

$$S(\alpha,\beta)=\sum_{i=1}^{n}\left[\left(f_a^1\right)_i\alpha+\left(f_a^1\right)_i\beta-\gamma_i\right]^2 \tag{2.20}$$

using sufficiency conditions for minimum functions:

$$\begin{aligned}\sum_{i=1}^{n}\left(f_a^1\right)_i^2\alpha+\sum_{i=1}^{n}\left(f_b^1\right)_i\left(f_a^1\right)_i\beta&=\sum_{i=1}^{n}\gamma_i\left(f_a^1\right)_i\\ \sum_{i=1}^{n}\left(f_a^1\right)_i\alpha\left(f_b^1\right)_i+\sum_{i=1}^{n}\left(f_b^1\right)_i^2\beta&=\sum_{i=1}^{n}\gamma_i\left(f_a^1\right)_i\end{aligned} \tag{2.21}$$

when the determinant of system (2.21) is not zero i.e., when the unique solution of (2.21) exists then:

$$\alpha=\frac{\sum_{i=1}^{n}\gamma_i\left(f_a^1\right)_i-\sum_{i=1}^{n}\gamma_i\left(f_b^1\right)_i\sum_{i=1}^{n}\left(f_a^1\right)_i\left(f_b^1\right)_i/\sum_{i=1}^{n}\left(f_b^1\right)_i^2}{\sum_{i=1}^{n}\left(f_a^1\right)_i^2-\left[\sum_{i=1}^{n}\left(f_a^1\right)_i\left(f_b^1\right)_i\right]^2/\sum_{i=1}^{n}\left(f_b^1\right)_i^2} \tag{2.22}$$

$$\beta=\frac{\sum_{i=1}^{n}\gamma_i\left(f_b^1\right)_i-\sum_{i=1}^{n}\gamma_i\left(f_a^1\right)_i\sum_{i=1}^{n}\left(f_a^1\right)_i\left(f_b^1\right)_i/\sum_{i=1}^{n}\left(f_a^1\right)_i^2}{\sum_{i=1}^{n}\left(f_b^1\right)_i^2-\left[\sum_{i=1}^{n}\left(f_a^1\right)_i\left(f_b^1\right)_i\right]^2/\sum_{i=1}^{n}\left(f_a^1\right)_i^2} \tag{2.23}$$

The secondary values of $a^{(1)}$ and $b^{(1)}$ can be determined by substituting for α and β in (2.12) and (2.13). From (2.2) - (2.5) and with secondary values of $a^{(1)}$ and $b^{(1)}$ the residuals can be calculated as:

$$\gamma_{ij}^{(1)}=y_i-f_i\left(x_i;a^{(1)},b^{(1)},c^{(0)}\right),i=1,\overline{n},j=1,\overline{4} \tag{2.24}$$

The mean square error can be obtained by:

$$\rho_j=\sum_{i=1}^{n}\left[\gamma_{ij}^{(1)}\right]^2,j=1,4 \tag{2.25}$$

The final choice of mathematical model is made by minimum value of ρ_j.

Thus, the predicted value of the parameter y_F can be obtained by:

$$y_F=\hat{f}\left(X_F;a,b,c\right) \tag{2.26}$$

where f is chosen from mathematical models (2.2) - (2.5)

In (2.26) the forecasting time $X_l = X_i + \tau$ where X_j is current time τ is the action time (time required to take action including time for processing actuator response etc).

2.17 CONCLUSION

Implementation of expert control involves building an expert system. Knowledge engineering deals with the details of such a design for implementation and the knowledge engineer is responsible for the developing strategy and stages necessary for building a successful expert controller.

The first stage of the development is the domain specification and decomposition. The entire domain has to be represented in a form suitable for symbolic manipulation in a computer. This is done by symbolic modelling of the deep hole machining system. To represent the interaction between the various elements of the machining system with constitute the machining operation, the machining operation is divided into machining sequence and machining process. The machining sequence is controlled by identifying the various events as Instant and Interval events and then representing the situation of machining by means of symbolic model of DHMS functions. Within a machining cycle, the machining process, i.e., actual metal machining, is described by the process descriptors. Five process descriptors, namely, Torque, Torque fluctuations, Thrust, Thrust fluctuations and Chip factor are identified as the necessary and sufficient machining responses to describe a machining condition. These five processes descriptors can generate a combination of 243 situations. These 243 situations form the premise of the rules used for expert control. The conclusions, which describes the corrective actions to be initiated by the expert controller, are supplied by the experts in the field of deep hole machining. In addition to the expert knowledge represented in the form of rules, additional rules, called temporal rules, are formed to keep tab on the time sequence. Procedural rules direct the changing of controlled parameters by the expert controller (slow, gradual, sudden, etc). Meta rules are the rules located at the top of the knowledge base and are designed to handle emergency situations, namely, tool failure, missing tool, etc. Trend control block makes use of the mathematical/statistical knowledge available for determining the cutting parameters when there is sufficient time for making decisions. Thus, the expert controller is capable of using the human expertise for high speed decisions and mathematical and statistical models in addition to human knowledge for slow speed decision making.

CHAPTER 3

MACHINING PERFORMANCE EVALUATION AND CONTROL

PREVIEW

Five necessary and sufficient parameters for identifying the machining condition in DHMS were discussed in the previous chapter. In this chapter a detailed study of these parameters and their significance in determining the machining condition is carried out.

3.1. INTRODUCTION.

Eyes and ears monitored cutting tool performance until the 80's, watching or listening for tooling faults and stopping the machine, usually after problems occurred. "The machinist watched the curl and colour of the chips," recalls a controls engineer at Giddings & Lewis Automation Technology [Owens, 1993]. "He heard the chatter in an end mill or the squeal of an insert. Then he cranked up the feed and speed to suit cutting conditions, tweaking the overrides occasionally when heavier stock or harder material came along."

Modern production plants with higher automation, exotic tooling made of carbide, ceramic and industrial diamond could not rely on the operator to respond to tool breakage, wear etc as the cost of a broken tool and damaged workpiece could send spirals of bad parts down the automated lines. In case of deep hole machining, this factor is exaggerated by the cost of damaged workpiece which may run couple of meters long along with the drilling tool which is very expensive. In addition to aforesaid losses, damage to the machines and machine downtime should not be under emphasized. Most important of all the losses may be the danger of "implanting hidden quality problems in the process or product that come back later to haunt the manufacturer". It is imperative that the DHMS should have "non human eyes and ears" or sensors to detect the machining condition and to pass on this message to the expert controller for corrective action. This system of estimating the machining condition by means of feedback from the cutting edge and the machine through various sensors operate on a simple principle: ***All machining actions, including breaking tools, have distinct and repeatable patterns or signatures that sensors can pick up and measure in some way***. Changes in these patterns signal a change in the present condition of the tool. Once optimum tool action is known, it can be programmed into the system. It can be argued that real-time measurement and control, rather than tool changing based on averages, greatly extends tool life and saves money. Rest of this chapter is dedicated to the detailed study of various machining signatures such as cutting force, torque etc., which can be sensed and adopted for controlling deep hole machining system.

3.2. TOOL CONDITION EVALUATION

Large amount of research has been done in relating the cutting force and torque in deep hole machining to describe the cutting tool condition in the machining zone [Peter Streicher, 1975; Chandrashekhar, 1984; Griffiths, 1982; Owens, 1993, Osman and Latinovic, 1979; Latinovic and Osman, 1975]. Analysis of the available knowledge shows that:

1. For a given speed and feed rate in deep hole machining,
   - i. Mean torque increases with the tool wear
   - ii. Mean thrust increases with the tool wear
   - iii. Torque value reaches a maximum at the end of tool life.
   - iv. Thrust value reaches a maximum at the end of tool life.
   - v. A sudden rise and fall in the amplitude of torque and thrust indicate tool failure.
2. For a given tool-workpiece combination, Amplitude of the torque and thrust indicate the stability of machining condition.

These results enable the knowledge engineer to infer that the torque and thrust are a prominent indicators of the tool and machining condition in deep hole machining and thus be monitored for evaluating the situation in the machining zone.

The knowledge that torque and thrust are important indicators of tool and machining conditions makes it necessary to obtain further knowledge for the selection of limiting values for a given tool-workpiece combination.

It is very important to recognize the phases of deep hole machining. The process of deep hole machining involves drilling/boring and burnishing. At the beginning of machining, only the drilling/boring phase is initiated. As this process progresses the burnishing pads gradually engage the machined hole. Normal deep hole machining process is set in motion when the entire length of the burnishing pads enter the work piece. At the end of the workpiece, the cutting edges of the tool come out of the workpiece while the burnishing pads are still at work. Figure 3.1 illustrates the phases of deep hole machining and Figure 3.2 illustrates the corresponding change in cutting force and torque. Figures 3.3 and 3.4 illustrate a clear distinction in the force and torque signatures for stable and unstable machining conditions respectively. The signatures of torque and thrust in unstable and stable machining provides the upper and lower limits of corresponding signatures to be employed in the expert controller. Torque and thrust values are controlled by varying the speed and feed Variation of

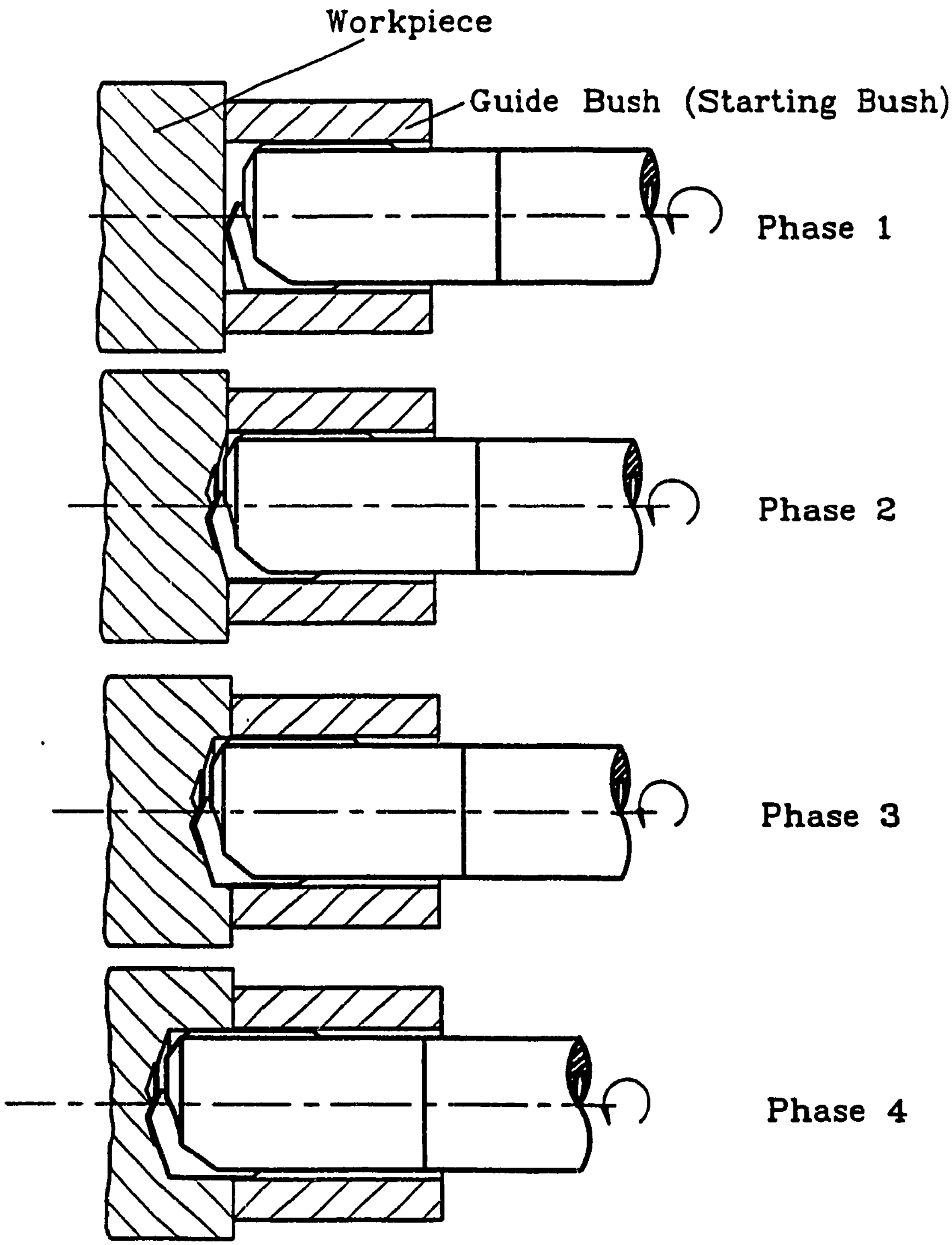


Figure 3.1: Phases of deep hole machining[36].

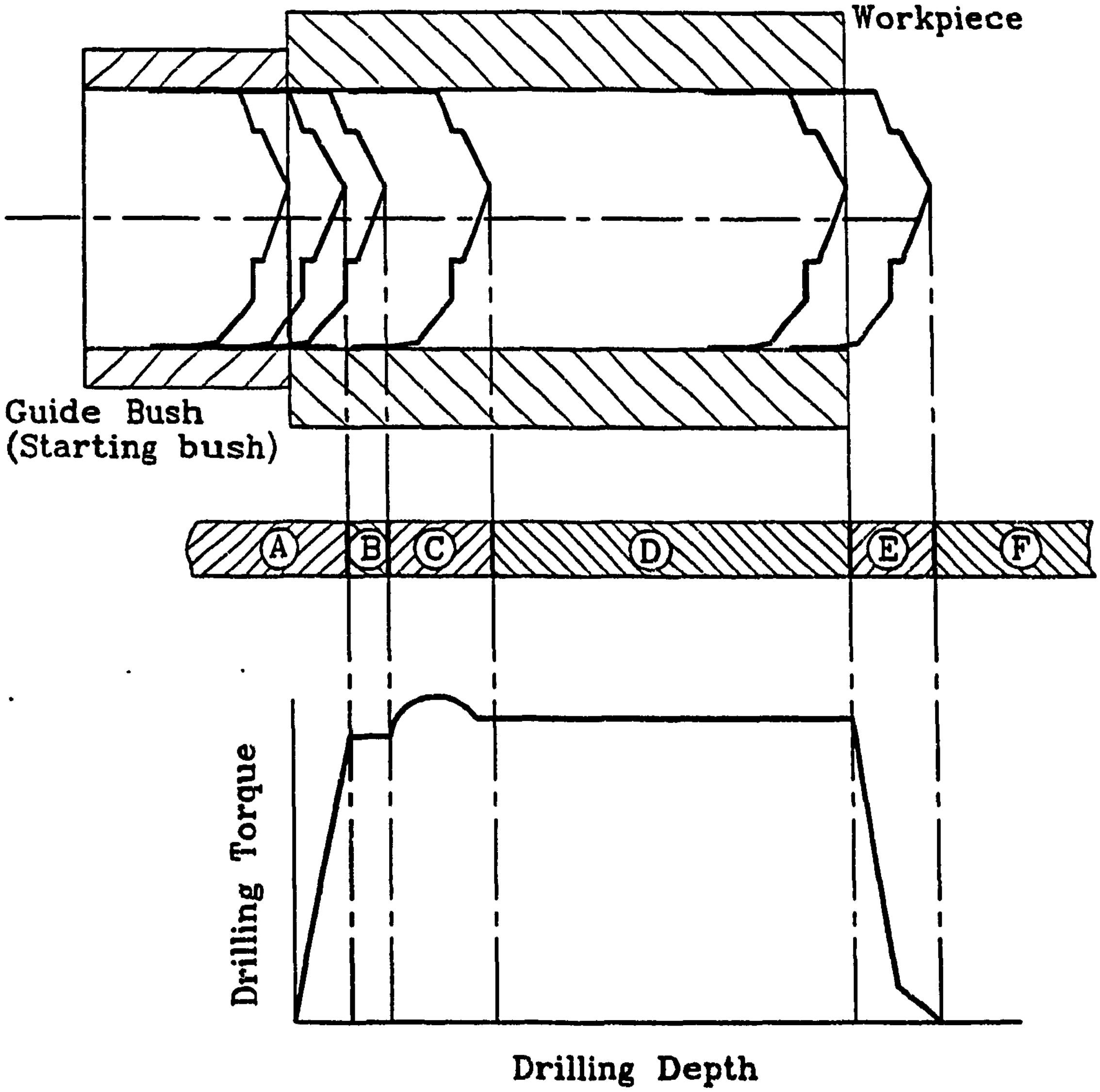


Figure 3.2: Torque at different phases of deep hole machining[36].

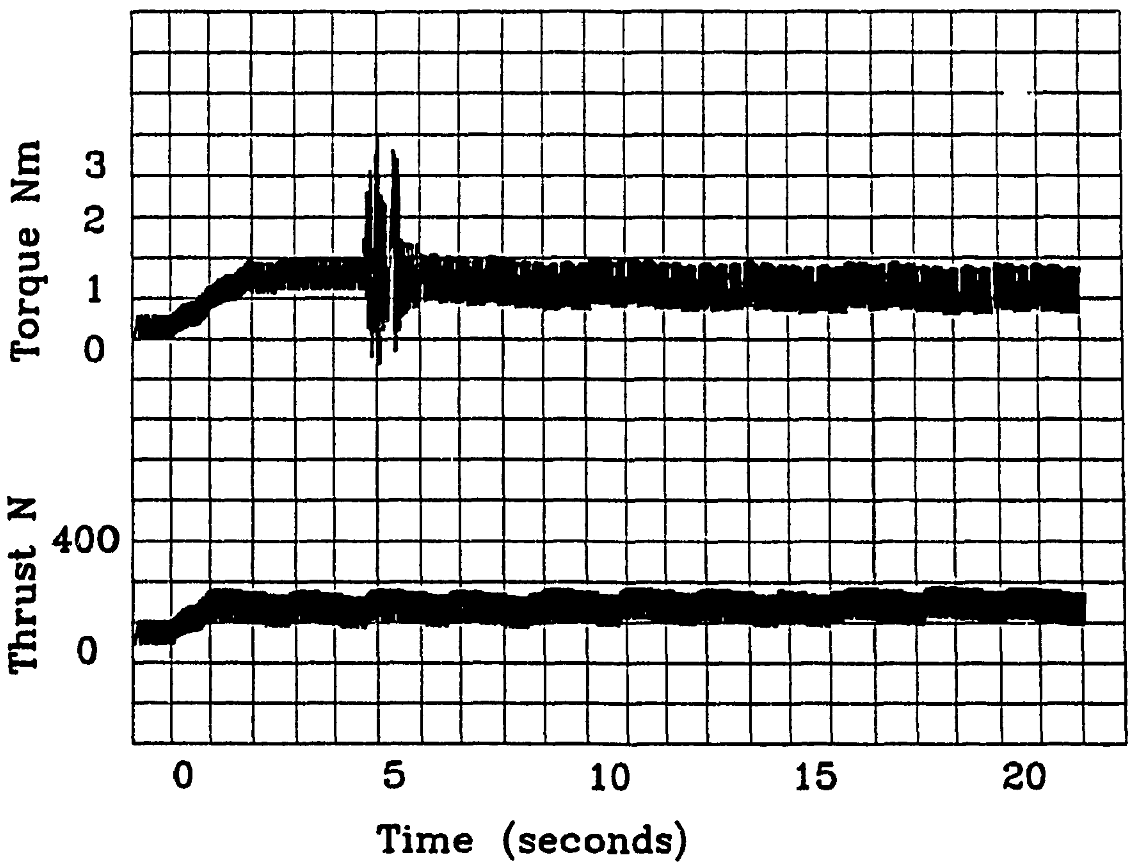


Figure 3.3: Torque and thrust during stable machining[67].

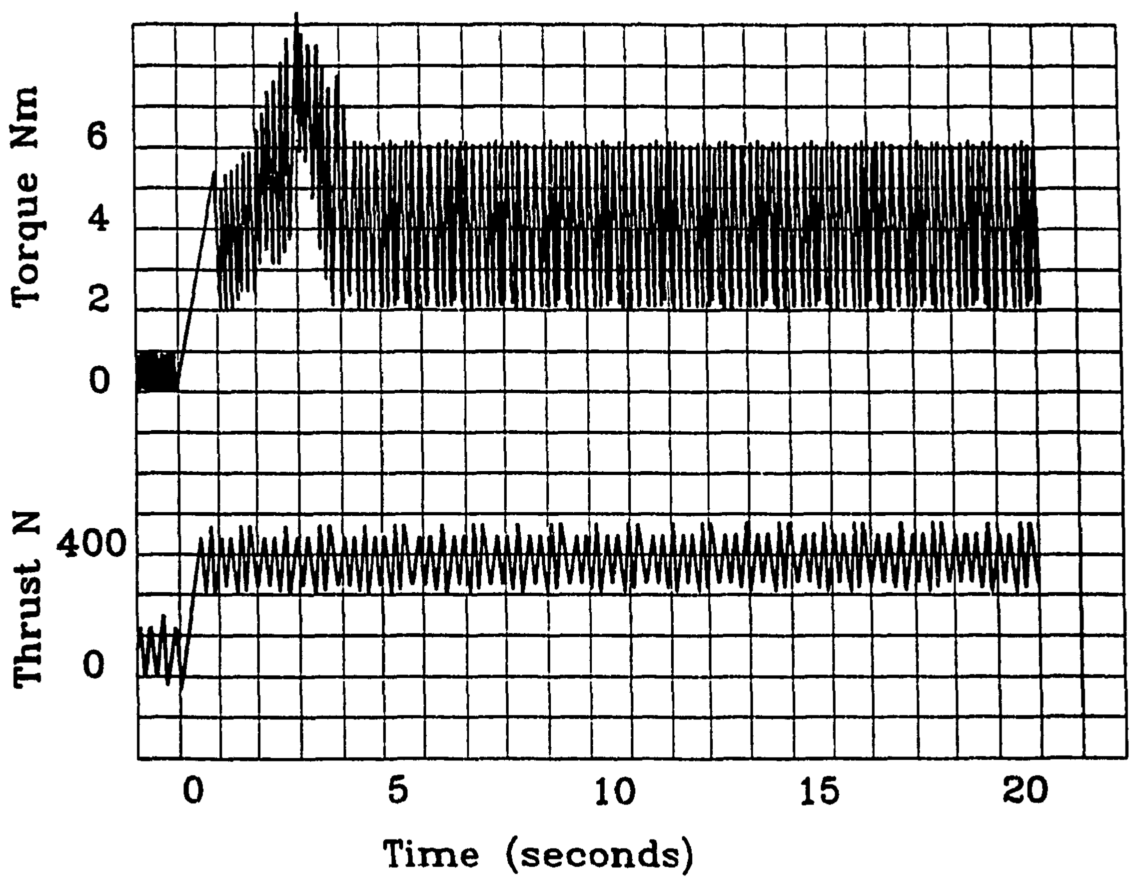


Figure 3.4: Torque and thrust during unstable machining[67].

torque and thrust with respect to cutting speed and feed rate are illustrated in Figure 3.5. Figure 3.6 illustrates the torque and thrust values for new and worn tools at various feed rates. Figure 3. 7 illustrates the percentage variation of torque and thrust with respect feed rates and thus define the bounds of torque and thrust values. Maximum values of the torque and thrust indicated in Figure 3.7 signals the wear limit of the tool. These limitations are clearly illustrated in the form of limit graphs in Figures 3.8 and 3.9. The amplitude of torque and thrust provide an important signature off the tool condition. Tool failure is signalled by an instant raise in the amplitude above the upper limit followed by a sudden drop to the missing tool limit [Owens, 1993] as shown in the Figure 3.10. The Missing tool limit indicates 'No Tool' situion with cutting tool is not present at the end of the boring bar. Table 3.1 [Chandrashekhar, 1984] summarises the relationship of torque and thrust with cutting speed and feed thus providing a basis for controlling the torque and thrust by manipulating the speed and feed.

3.3. WORKPIECE CONDITION MONITORING:

3.3.1. QUALITY OF THE HOLE:

It is important to establish the relation between the controlling parameters namely, Speed and Feed with the integrity of the surface produced. B.J. Griffiths [1982] has done extensive studies on this subject and has documented his findings. The aspects

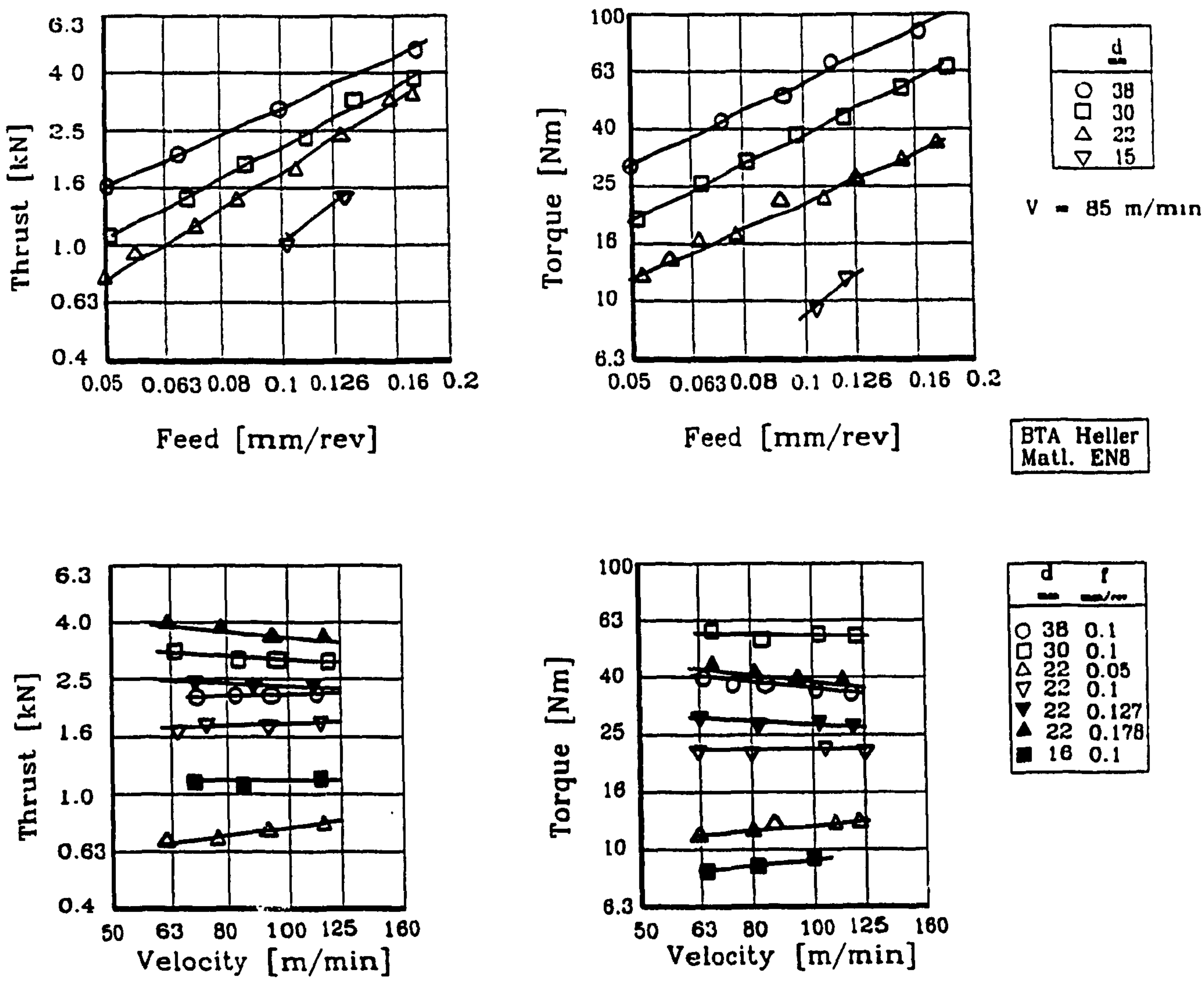


Figure 3.5: Torque and thrust Vs cutting parameters[36].

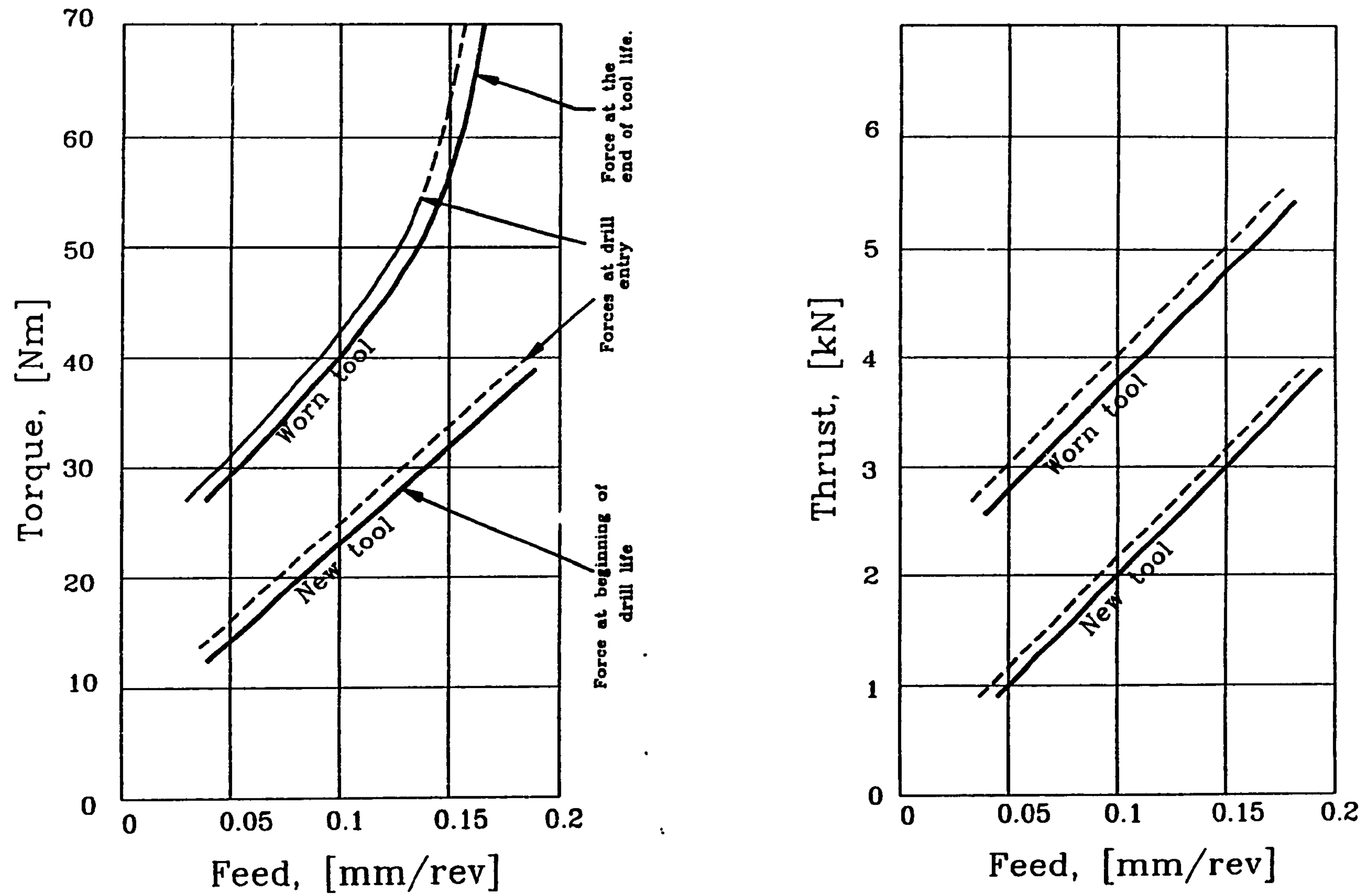


Figure 3.6: Effect of tool condition on torque and thrust[36].

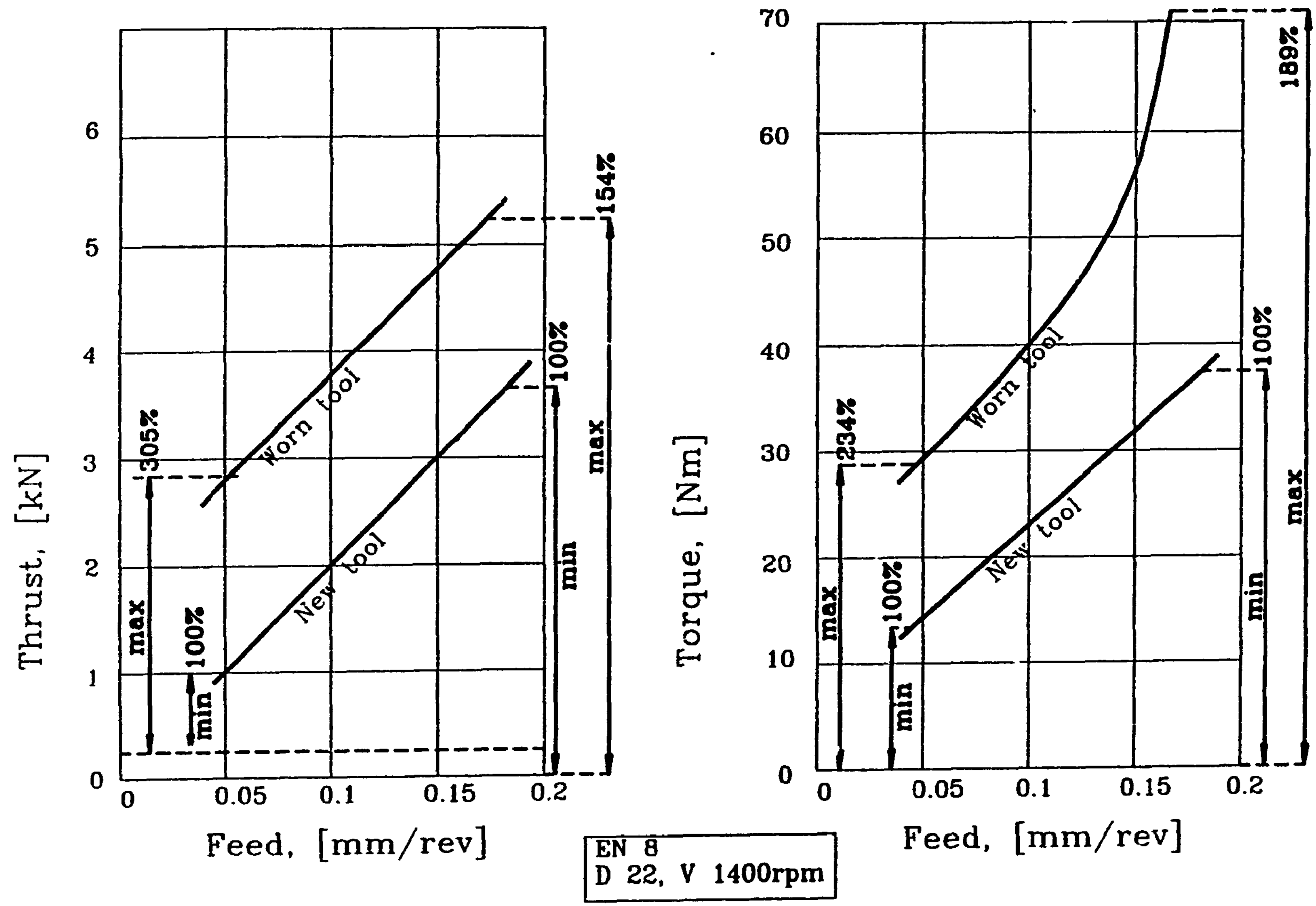


Figure 3.7: Percentage change in torque and thrust with tool wear[36].

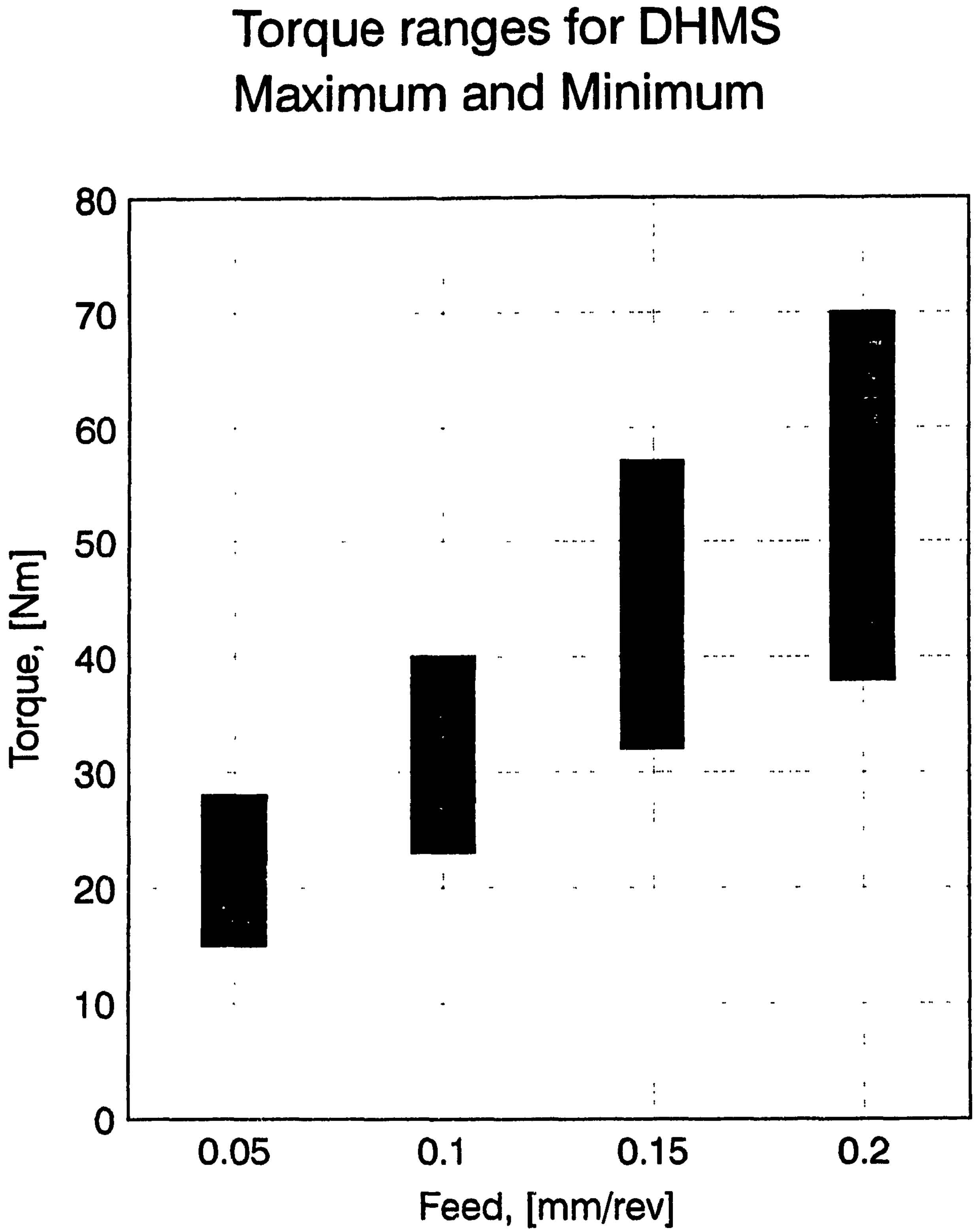


Figure 3.8: Limit graph for Torque.

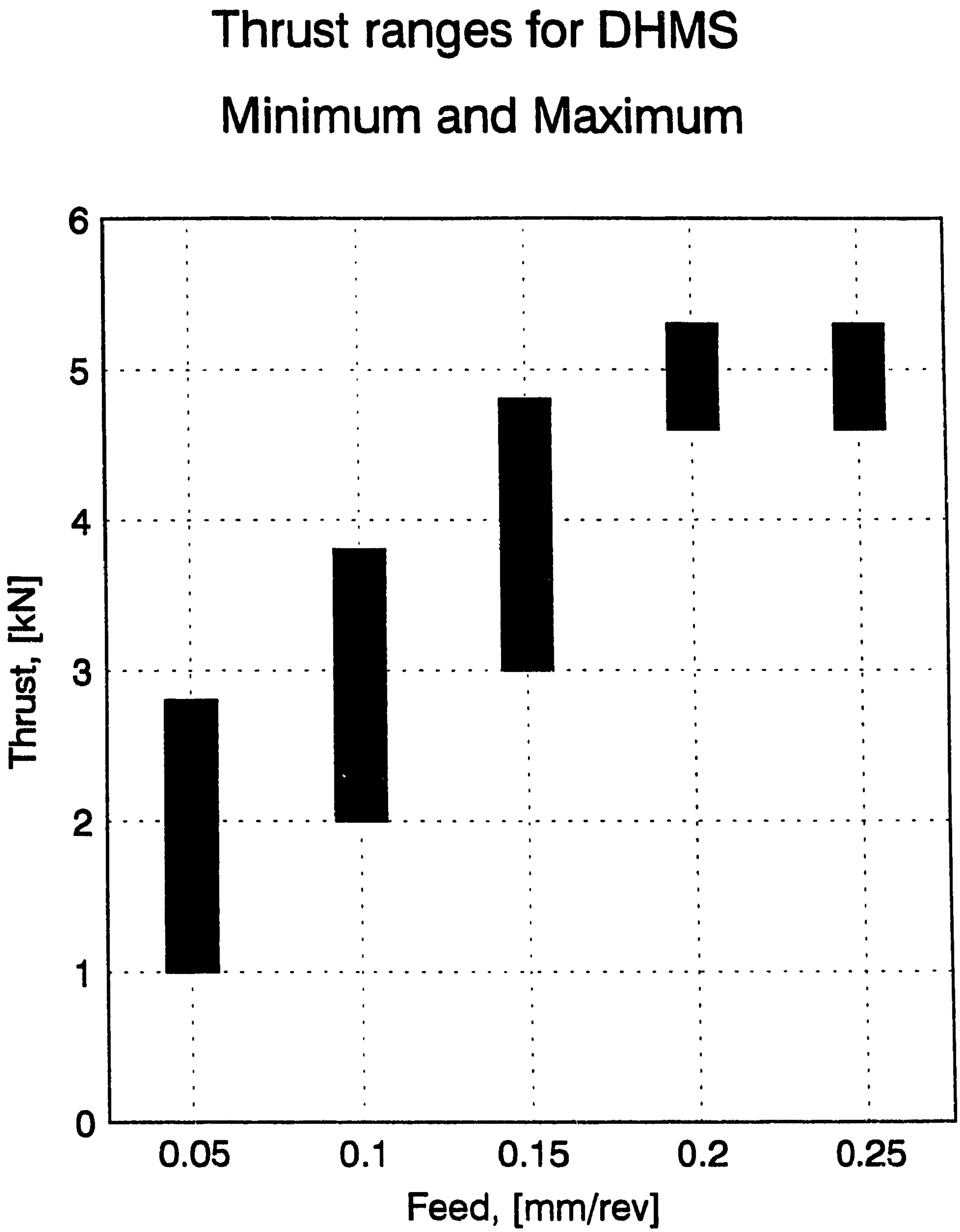


Figure 3.9: Limit graph for Thrust

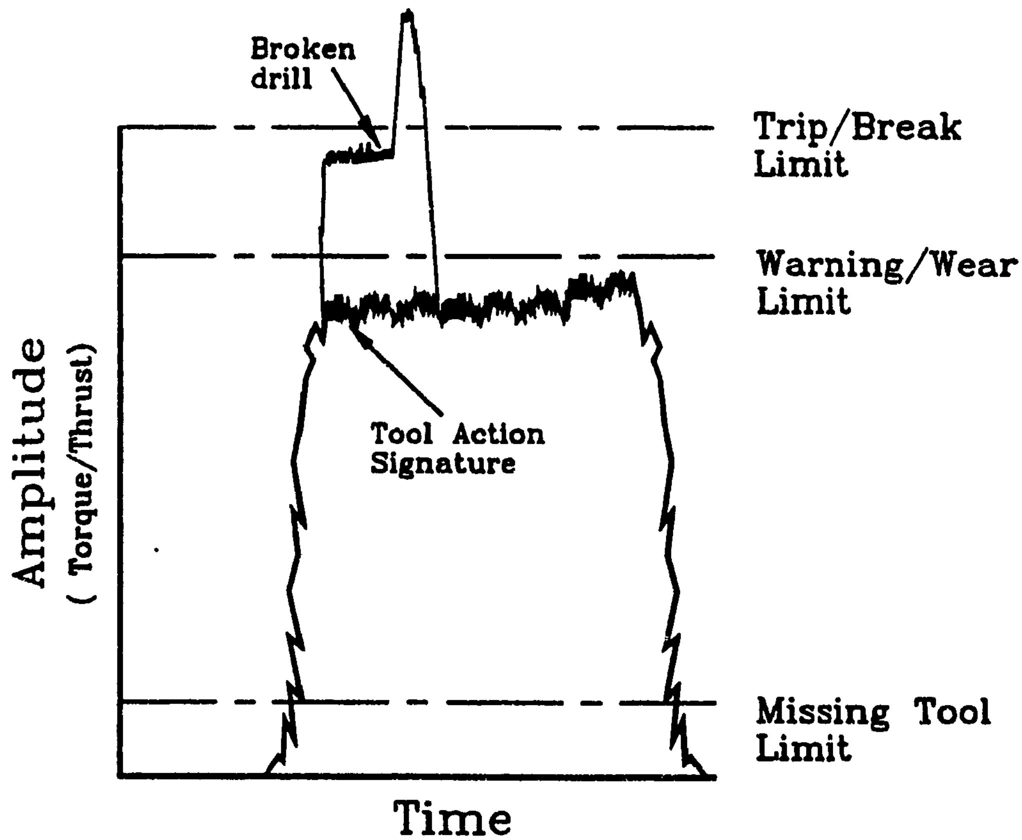


Figure 3.10: Force signature for tool failure[64].

of surface integrity [El-Khabeery, 1991] are:

Table 3.1

TOOL	MEAN VALUES	PEAK VALUES
B.T.A HELLER	Torque = $0.713*F^{0.95}*D^{1.81}$	Torque=$1.88*f^{0.929}*d^{1.516}$
	Thrust = $1.912*f^{1.06}*D^{0.78}$	Thrust=$2.67*f^{0.679}*d^{1.15}$
B.T.A SANDVIK	Torque= $0.785*f^{0.95}*d^{1.48}*V^{0.19}$	
	Thrust = $0.465*f^{0.97}*d^{0.79}\ V^{0.23}$	
EJECTOR	Torque = $0.743*f^{0.96}*d^{1.49}*V^{0.2}$	
	Thrust=$0.405*f^{1.03}*d^{0.8}*V^{0.27}$	

(Matl. EN8, Thrust kN, Torque Nm, d mm, f mm/rev, V m/min)

- Surface topography
- Mean Groove depth
- Surface finish
- Deformation depth
- Microstructure

A deep hole machined surface generally consists of grooves and the spacing between these grooves corresponds to the feed rate. B.J. Griffiths [1982] concludes that this surface topography did not alter during the various combinations of speed and feed. The relationship between the mean groove depth and feed rate is illustrated in Figure 3.11. A linear relationship appears to exist. No correlation was found between mean groove dept and drilling speed [Croniager et al., 1979]. This indicates that the variation of feed brings about changes in the surface topography of drilled surface. The bore surface finish at various speeds and feeds were presented [Sakuma et al. 1982] in the Figure 3.11. The researcher also concludes that the drilling speed has little effect on the surface finish. The study of literature published by various researchers [Griffiths. 1982; Sakuma et al., 1980, 1982; El-khabeery,1991] can be used to draw following conclusions.

1. Increase of drilling **feed** results in

    a. increasing mean groove depth

    b. decreasing surface finish quality

    c. increasing depth of plastic deformation

2. Increase of drilling **speed** results in

    a. negligible change in mean groove depth

    b. negligible change in surface finish quality

    c increasing depth of plastic deformation

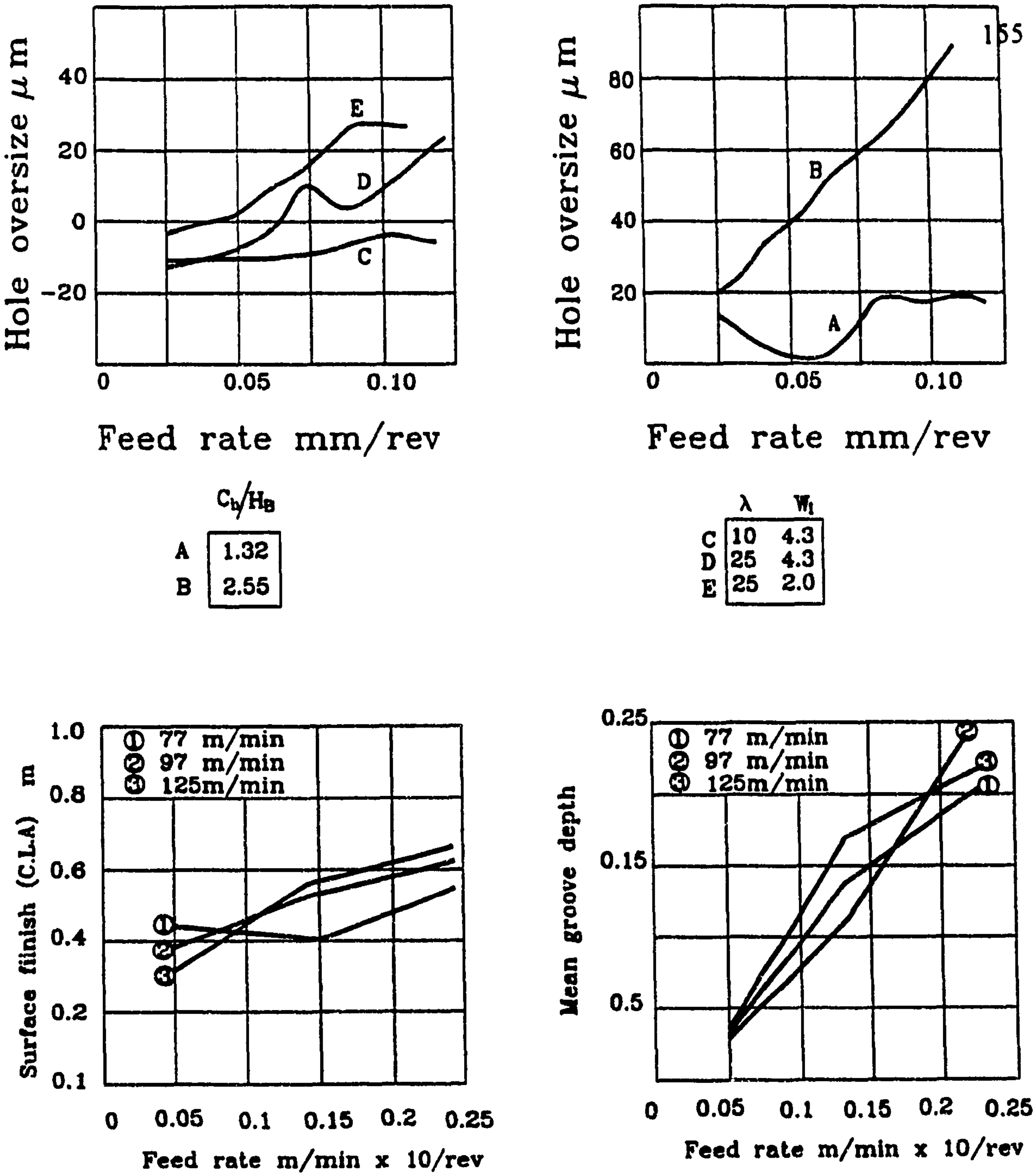


Figure 3.11: Variation of hole quality with feed[72,73].

3.3.2. DIMENSIONAL ACCURACY OF THE HOLE

Enlargement of shrinkage of machined hole with respect to the desired hole diameter is a common phenomenon in deep hole machining. This depends on several factors [sakuma,1980] e.g., machining condition, tool shape, work material etc., In order to obtain desired accuracy of the machined hole for a given workpiece and tool combination, those aspects of machining, namely the machining condition, which bring about a change in the desired hole size are to be considered for expert control of DHMS. Sakuma et. al,[1980] has shown that the hole oversize varies with the ratio of specific cutting force and hardness (material factor), width and chamfer angle of the guide pad(tool factor) and the feed rate. For a given combination of tool and workpiece the material factor and tool factor are constant and hence the feed rate forms the most important factor effecting the final dimension of the machined hole. The relation between feed rate and hole oversize is illustrated in Figure 3.11.

Thus, it is imperative that the tool condition, machining condition and the quality of the workpiece are described by the torque and thrust values. Values of torque and thrust depend on the cutting speed, feed and diameter of the tool. For a given tool, the cutting speed and feed are the determining parameters of torque and thrust and hence the condition of the tool, machining condition and the quality of the hole can be controlled by controlling cutting speed and feed in deep hole machining.

3.4 SUFFICIENCY OF THE AVAILABLE DATA

Even though the available knowledge looks sufficient for complete determination of the machining situation, One of the most important components of deep hole machining, namely, the coolant and its effect on the performance on deep hole machining has not been considered. Coolant in deep hole machining occupies more prominent position than in any other machining [Zwingmann, 1975]. The two parameters of coolant, quality and quantity, in the machining zone has a significant bearing on deep hole machining. The effect of the quality of coolant on the tool and workpiece have been studied by various researchers [Nicholson et al, 1979; Nicholson, 1979, Syrett et al, 1979]. It has been demonstrated [Syrett, 1979; Osman et al., 1981] that the cutting fluid flow is an important contributer to the performance of deep hole machining. Despite the importance of the cutting fluid flow, hardly any work has been done in this area and hence expertise is not readily available for building the knowledge base. This lack of available research regarding the effect of coolant 'quantity' on the performance of deep hole machining emphasizes the necessity for a detailed study of the coolant quantity in terms of rate of flow and pressure and its effect on the performance of deep hole machining in general and BTA and Ejector drilling in particular to build a complete and comprehensive knowledge base for the expert controller.

3.5. INVESTIGATION OF COOLANT FLOW AND PRESSURE IN BTA AND EJECTOR DRILLS

3.5.1 OBJECTIVES

The objectives of the investigation are to:

1. Establish a clear relation between the machining parameters, namely, the cutting speed and feed on the performance of coolant.
2. Determine the maximum (Q_{max}) and minimum (Q_{min}) flow required for economical machining.
3. Determine the optimal flow rate (Q_{opt}) for the maximum tool life.
4. Determine the flow losses at various conduits and develop corrective measures to maximize the efficienty of coolant delivery to the machining zone.
5. Evaluate the chip removal process by the coolant.

The coolant is pumped into the machining zone through the inlet in the connector and flows through the annular channel between the boring bar and the inner tube of in both the cases of BTA(Figure 3.12) and Ejector(Figure 3.13) drilling system. Most of the coolant is forced through the holes in the drill head for cooling and lubricating the cutting edges and the supporting pads. After cooling and lubricating, the coolant flows out of the machining zone carrying with it the chips produced during machining through the inner tube via ejector nozzles. This diversion produces the ejector effect, i.e., creates a partial vacuum in the inner tube. This vacuum causes the coolant from

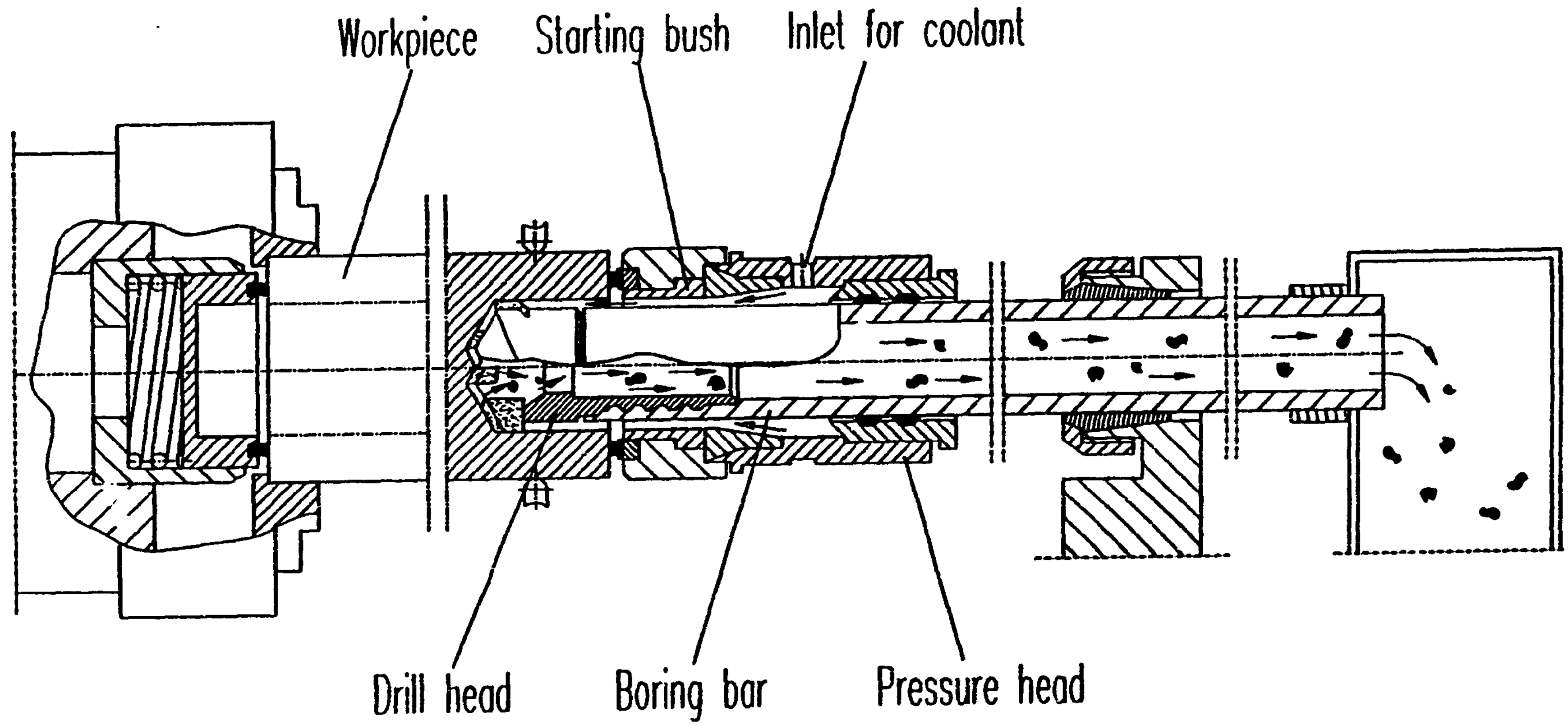


Figure 3.12: BTA drilling system

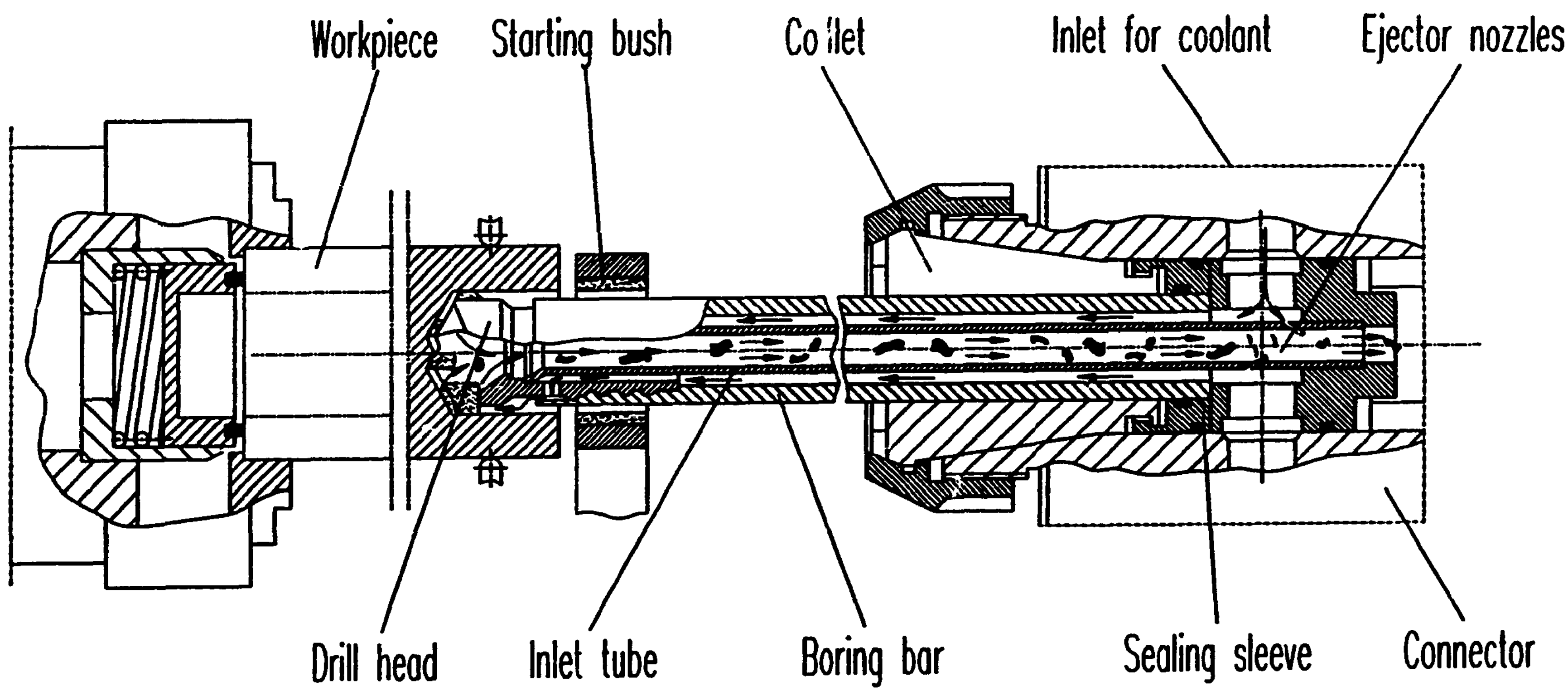


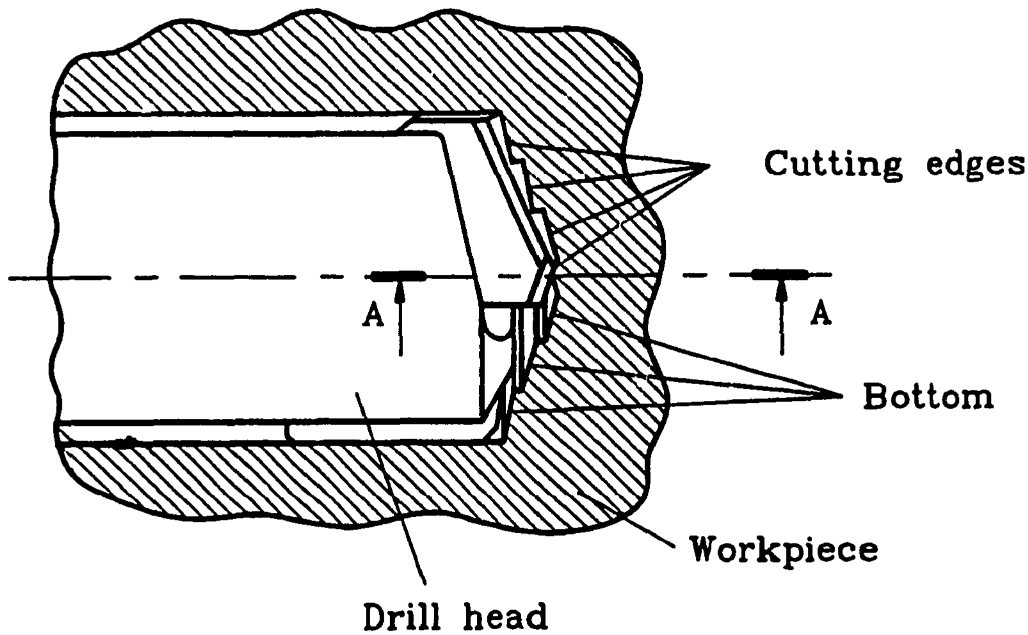
Figure 3.13: Ejector drilling system

the machining zone to be drawn into the inner tube along with chips into the outlet. Therefore, the coolant flow can be divided into three zones: inlet annular channel, machining zone, and chip removal channel.

3.5.2 COOLANT FLOW IN THE MACHINING ZONE

Due to the unique geometry of the drill head, the bottom of the hole produced in deep hole drilling has a special form as illustrated in Figure 3.14. The space between this special bottom surfaceof the workpiece and the flank surfaces of the drill head is termed 'bottom clearence'. The cutting fluid pressure in the bottom clearence has a major influence on the performance of deep hole machining operation as it determines the effectiveness of cooling and lubrication of the flank and rake contact areas of the drill head. Higher the pressure of the cutting fluid in the bottom clearence, better the penetration of the fluid into the narrow passages between the tool flank and the workpiece. It is thus imperative that the maintaining a high pressure in the bottom clearence is important to achieve better tool life and hence a more economical machining. Unfortunately little is known about the parameters which effect the coolant pressure at the bottom clearence.

Several experiments were performed to investigate the influence of various parameters on the cutting fluid pressure distribution in the bottom clearance using a single edge



A–A

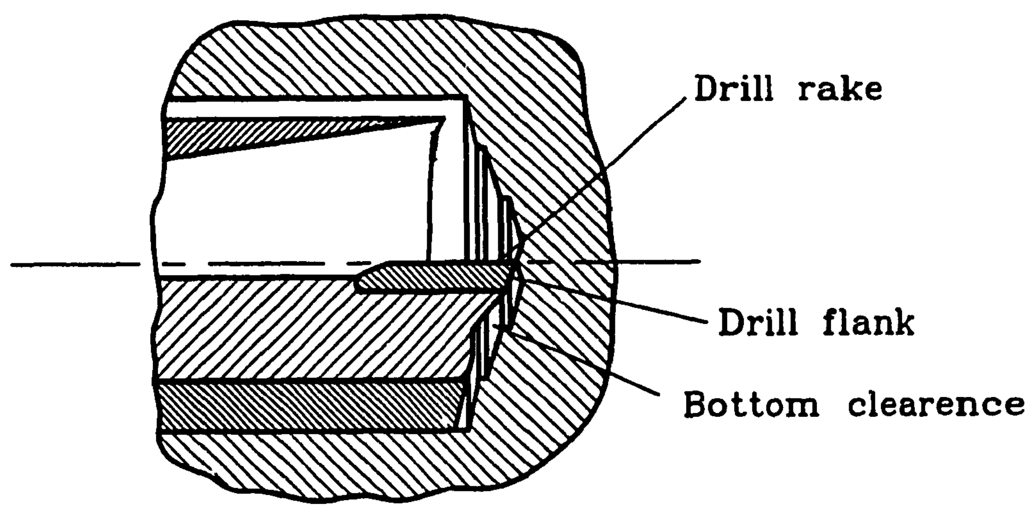


Figure 3.14: Details of bottom workpiece bottom profile.

B.T.A solid boring head (Drill "A", American Heller) and a multi-edge solid drilling head (Drill "B"). The diameter of each drill is 50.8 mm (2"). Some design parameters of the drill heads are shown in Figures 3.15 and 3.16.

3.5.3 STATIC PRESSURE DISTRIBUTION

The experimental set up consists of a standard deep hole boring machine (Figure 1.2) equiped with a variable coolant supply system. The standard basic deep hole boring machine is equiped with a pressure head, a boring bar and drill head. The variable coolant supply system includes chip traps and filters, precise turbine flow meter, variable flow control valve etc. The system is capable of delivering a flow up to 220 lit./min and generating a pressure of 4.5 MPa. The precision turbine flow meter (accuracy ±1%) was used to measure the flow rate. The main setup required for both the static and dynamic pressure measurement experiments is shown in Figure 3.17 and 3.18 respectively. Workpiece made of a cylindrical plexiglass of 75mm diameter and is held stationary by the chuck. The workpiece made of plexiglass is selected to visualize the movement of coolant in the machining zone. The workpiece is pre bored to the cutting tool diameter and the bottom of the workpiece is provided with bleed holes. Bleed holes are tapped to house the tube connections to the pressure measuring equipments. The position of the holes on the workpiece are illustrated in Figure 3.19 The workpiece is tightly held in the chuck such that the poly-tubes are still accessible

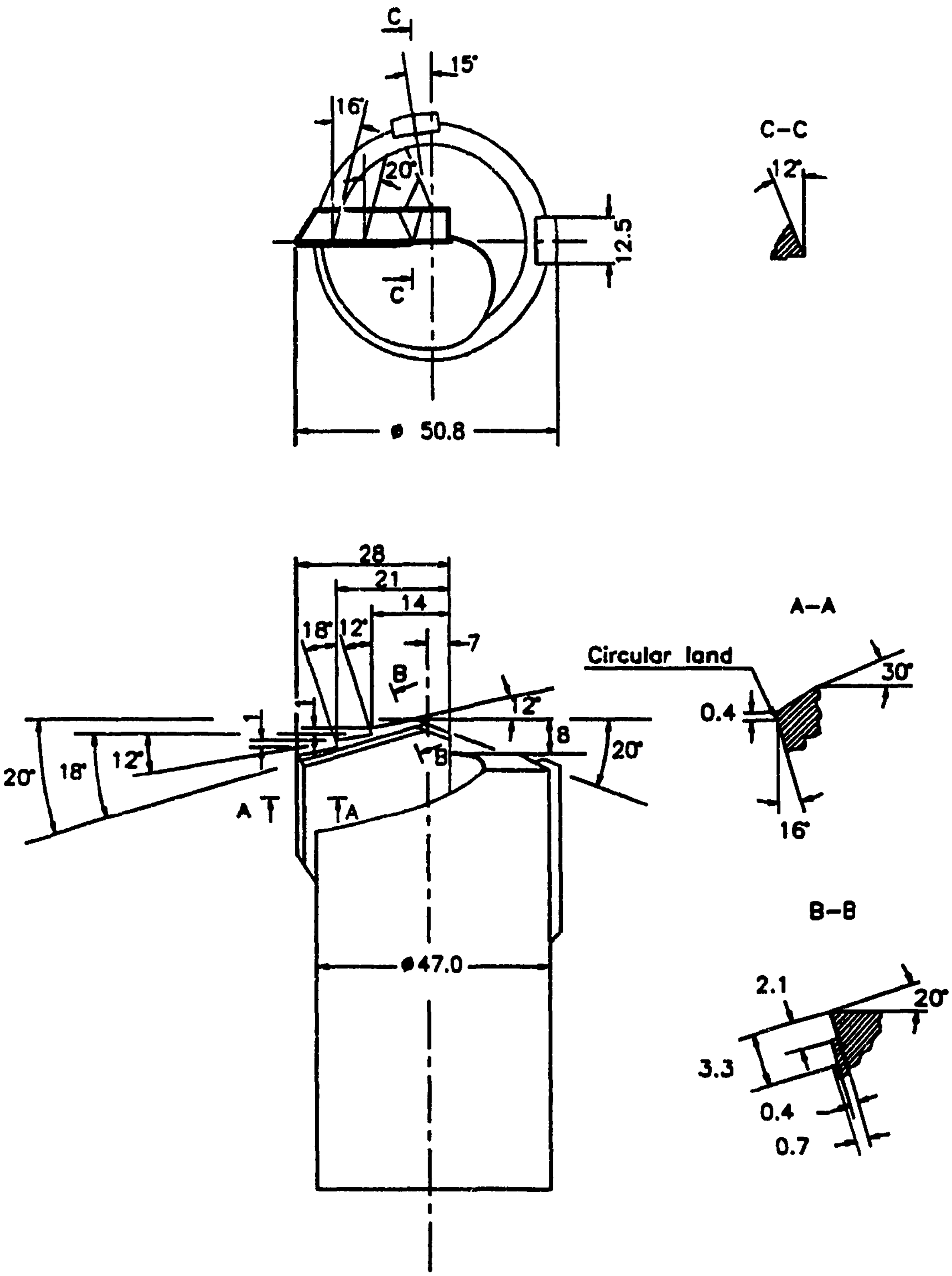


Figure 3.15: BTA solid boring head (drill 'A').

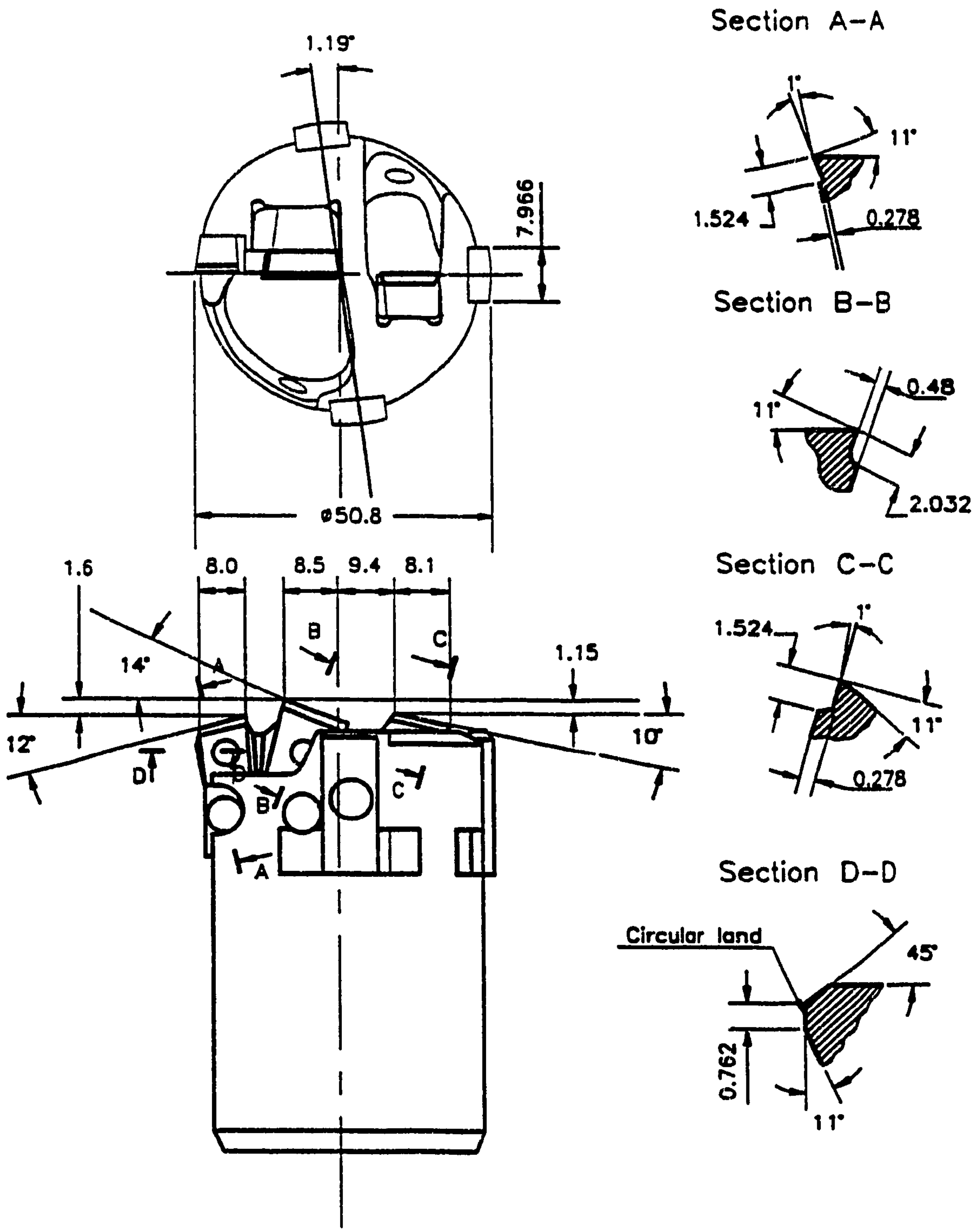


Figure 3.16: Multiedge solid boring head (drill 'B').

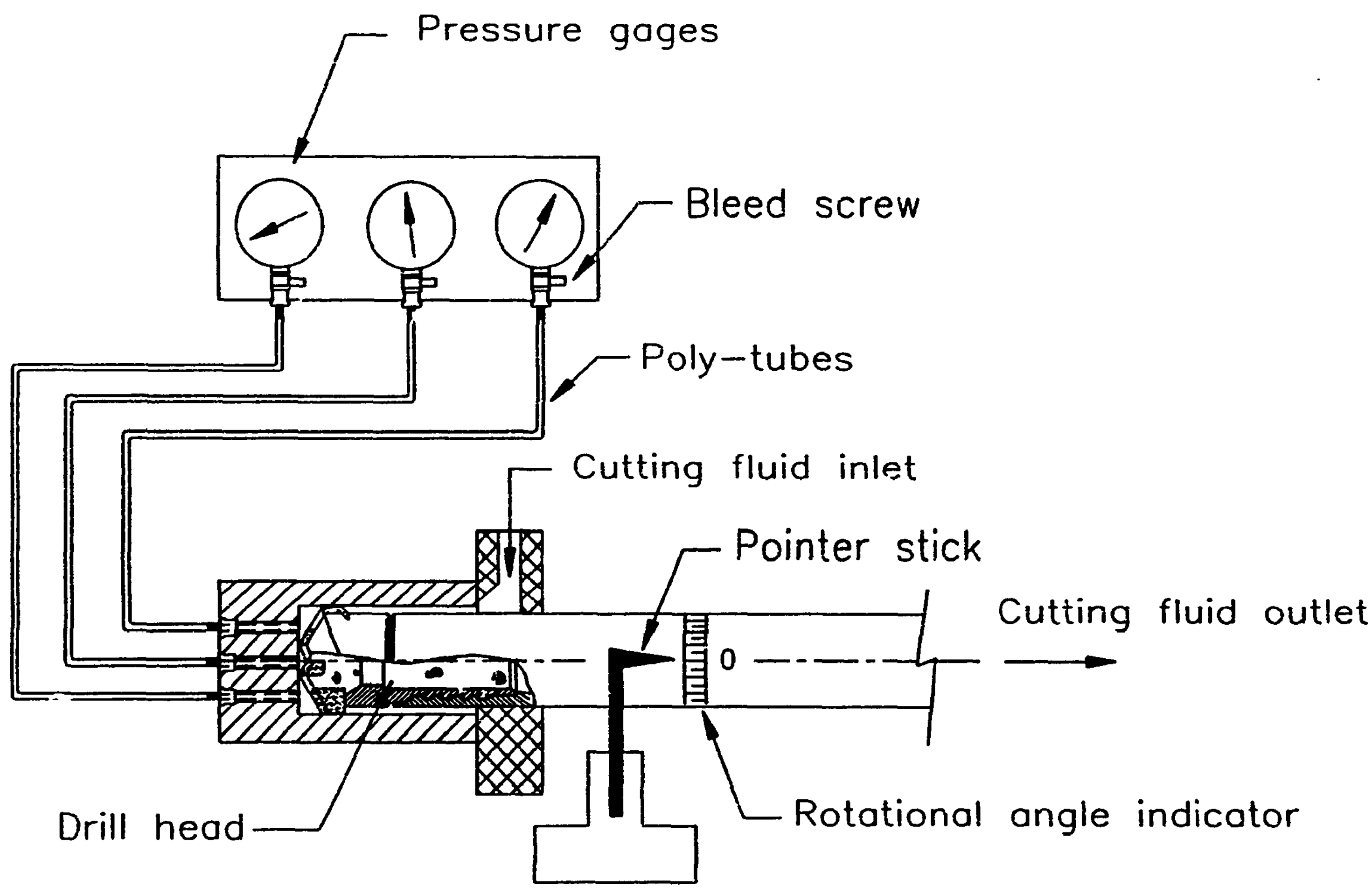
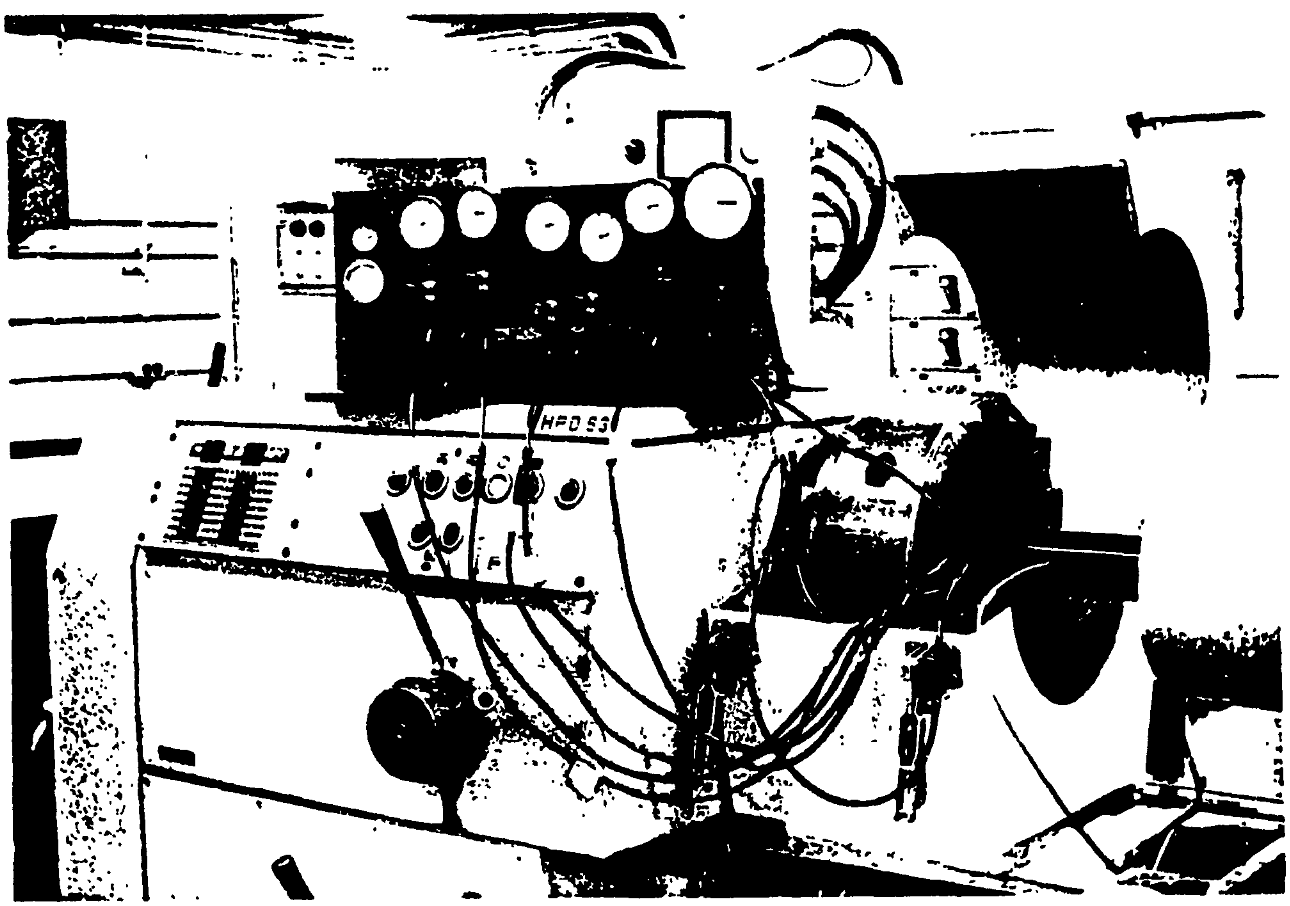
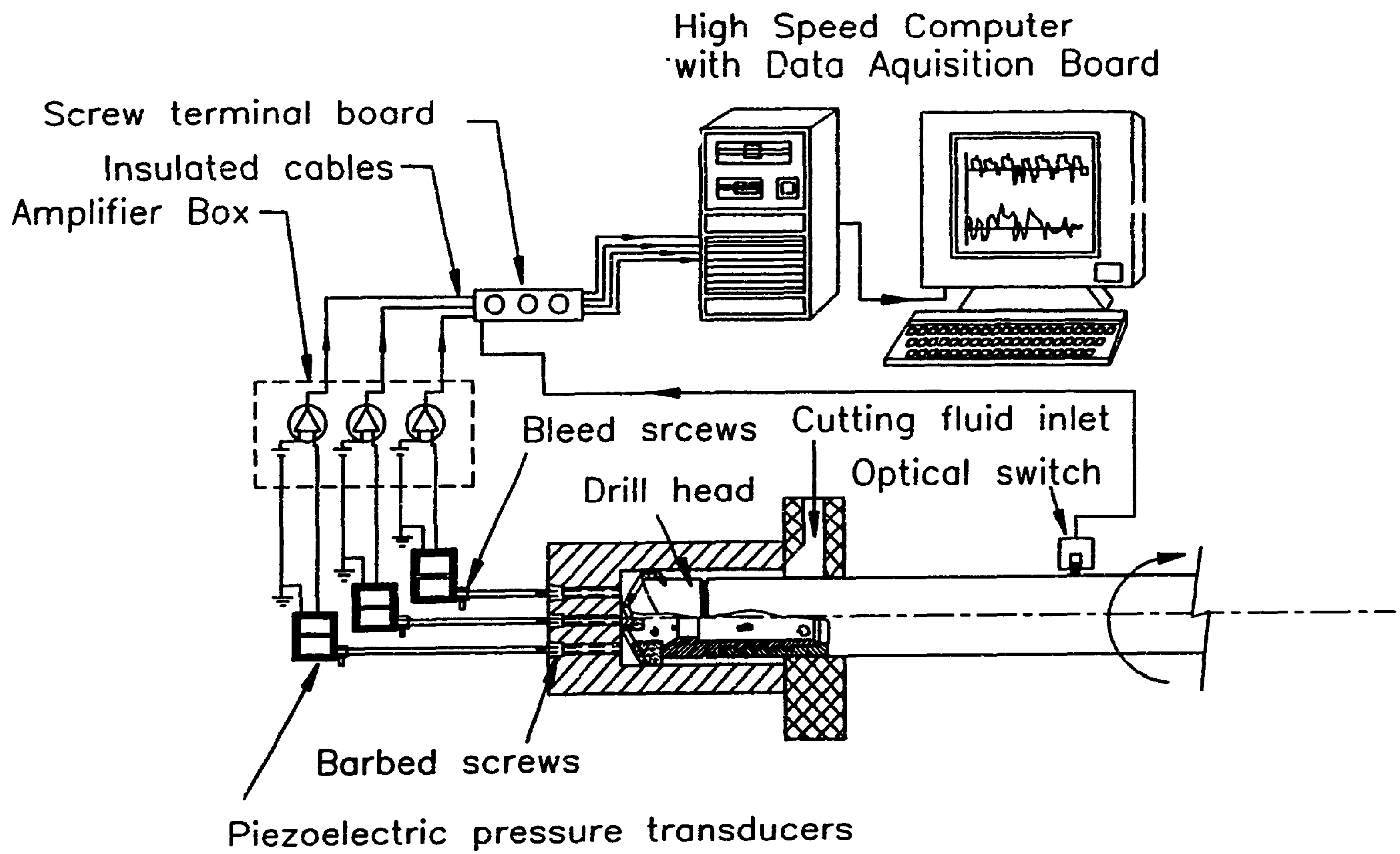


Figure 3.17: Schematic diagram of the experimental setup - static pressure distribution



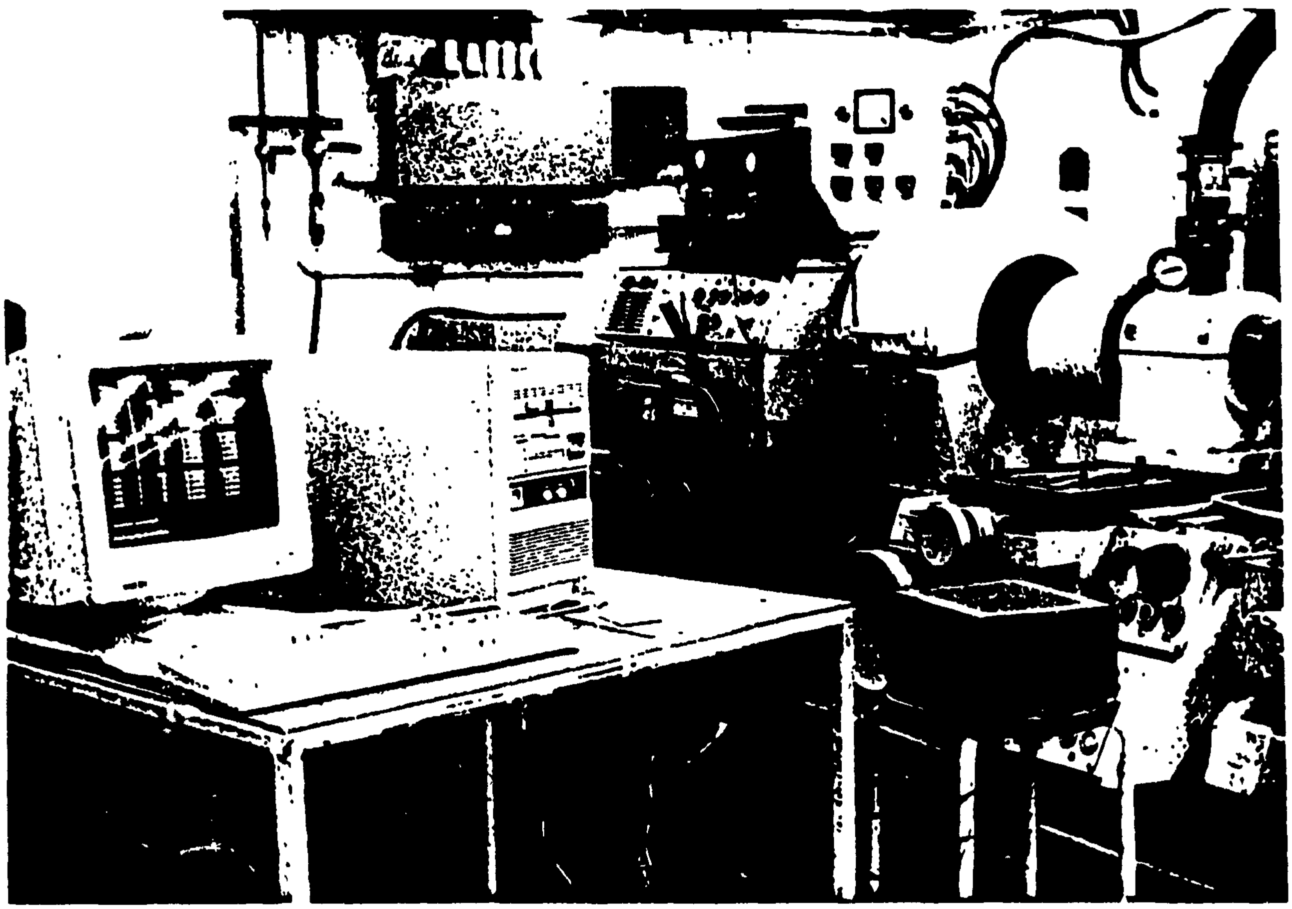
(B)

Figure 3.17: Experimental set up - Static pressure distribution



(A)

Figure 3.18: Schematic diagram of the experimental setup - dynamic pressure distribution



(B)

Figure 3.18: Experimental setup - dynamic pressure distribution

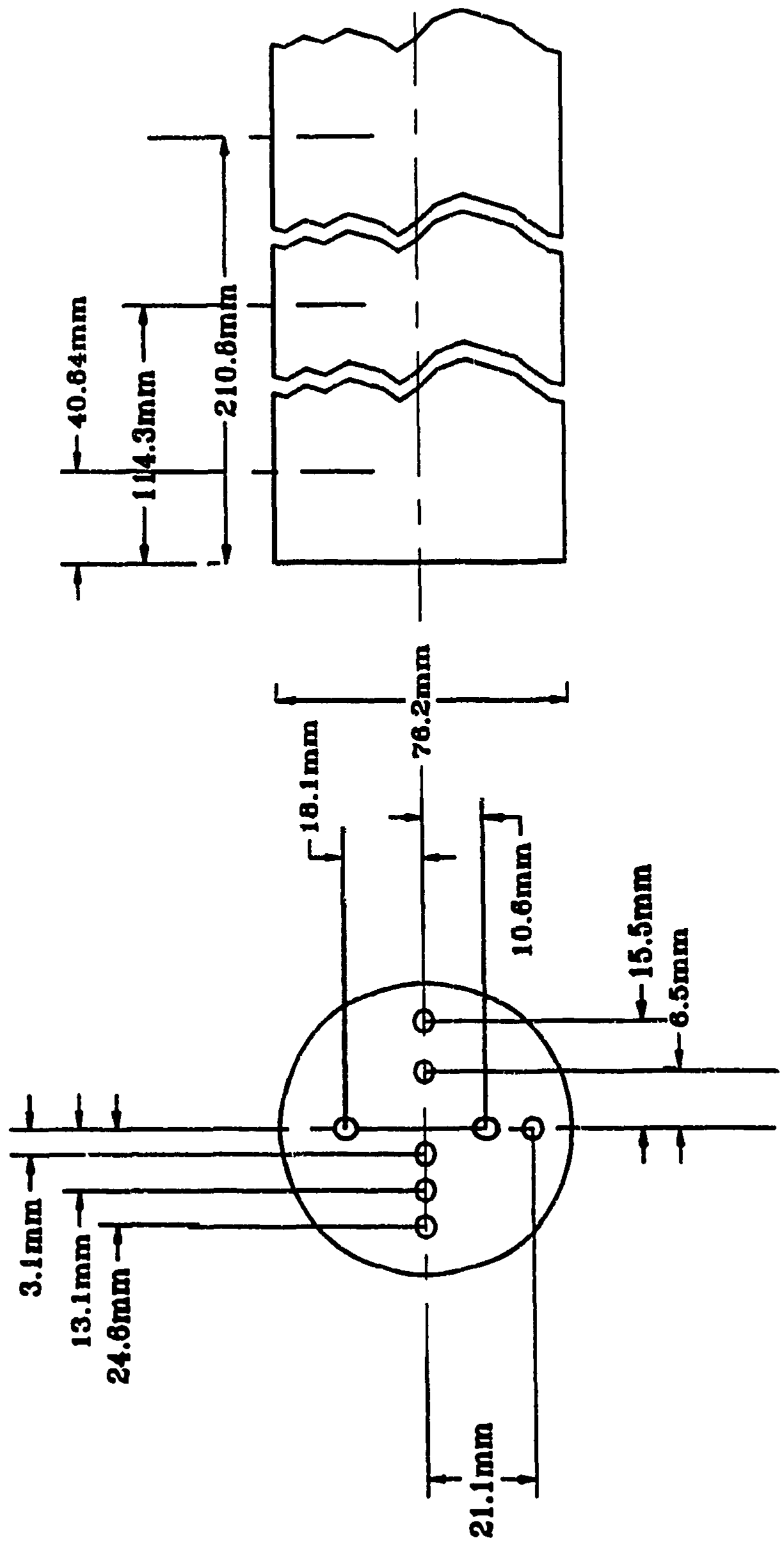


Figure 3.19: Plexiglass workpiece specimen

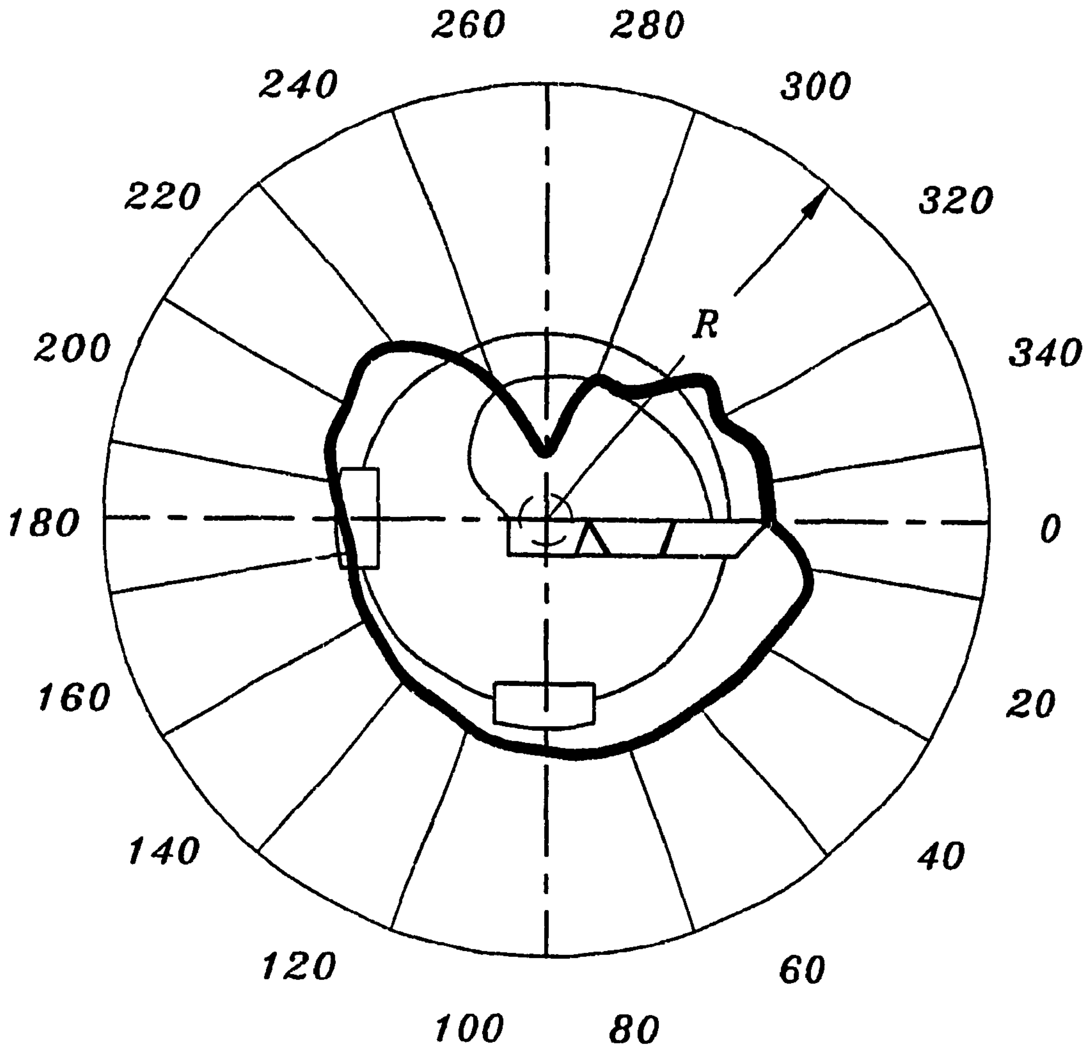
and the drilled holes are horizontally aligned. The drill head is attached to the boring bar. The pressure head is advanced toward the workpiece and is hydraulically pressed onto it. The boring bar is advanced till the cutting tool touches the bottom of the workpiece. Pressure gauges (0-600KPa) are connected to the end of the poly tubes via bleeding valves to relieve the trapped air in the cutting fluid reaching the pressure gauge. A position indicator to indicate the angle of rotation of the boring bar is mounted on the boring bar assembly.

The boring bar is gradually rotated till the cutting-tool's edge is in line with the pressure measurement holes on the workpiece and the position indicator is set to zero degrees. The coolant supply is initiated for a particular flow rate and the pressure readings are noted after removing all the entraped air from the measuring system using bleeding valves. Pressure is read for different angular positions of the boring bar for a complete cycle. This procedure is repeated for various flow rates from 115 to 170 lit./min. The results are plotted on cycle graphs as shown in Figure 3.20.

3.5.4 DYNAMIC PRESSURE DISTRIBUTION

A) Effect of speed:

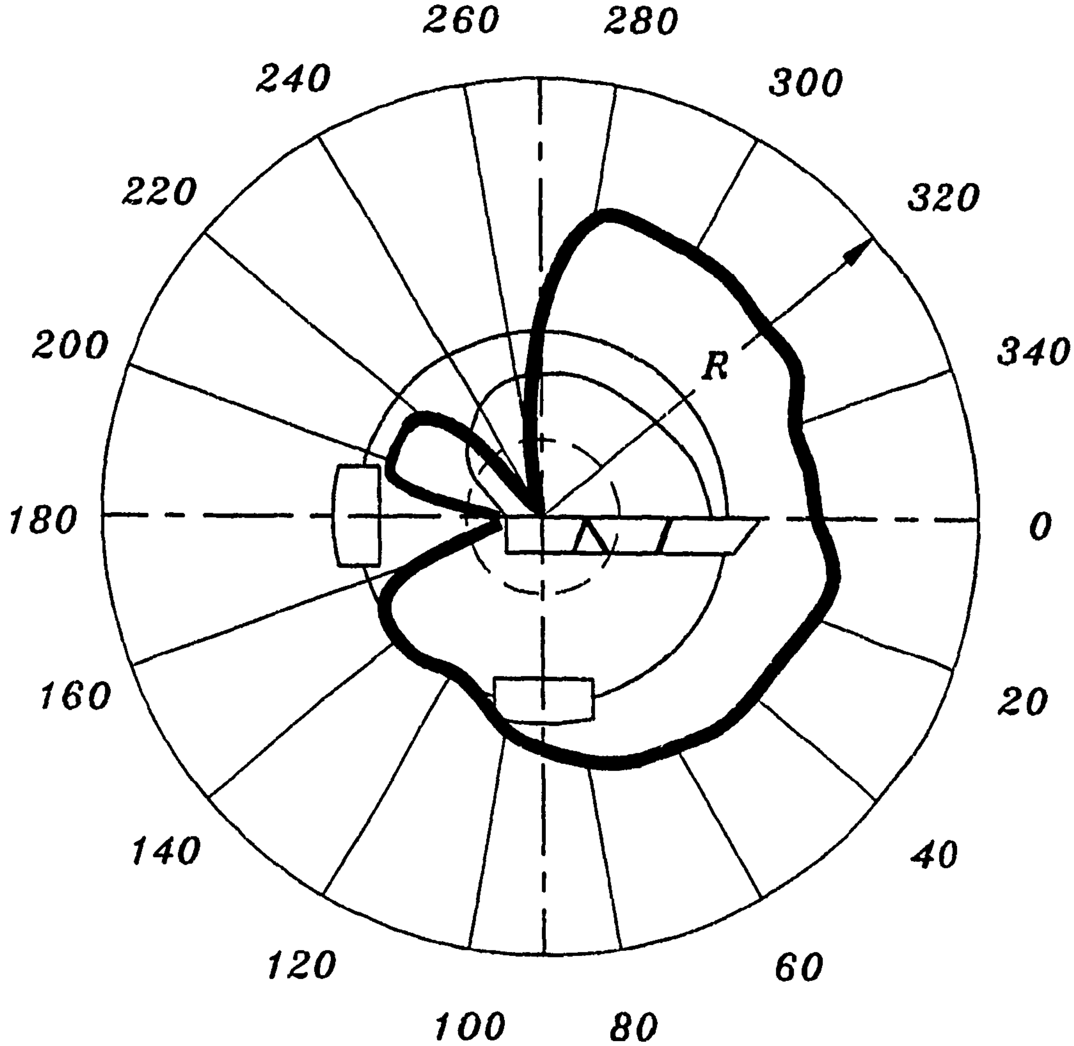
For measuring the dynamic pressure distribution, the previous experimental set up was modified to enable recording of dynamic pressure values. The pressure gauges were



R= 30KPa

Drill 'A' - Transducer at r=3.1mm

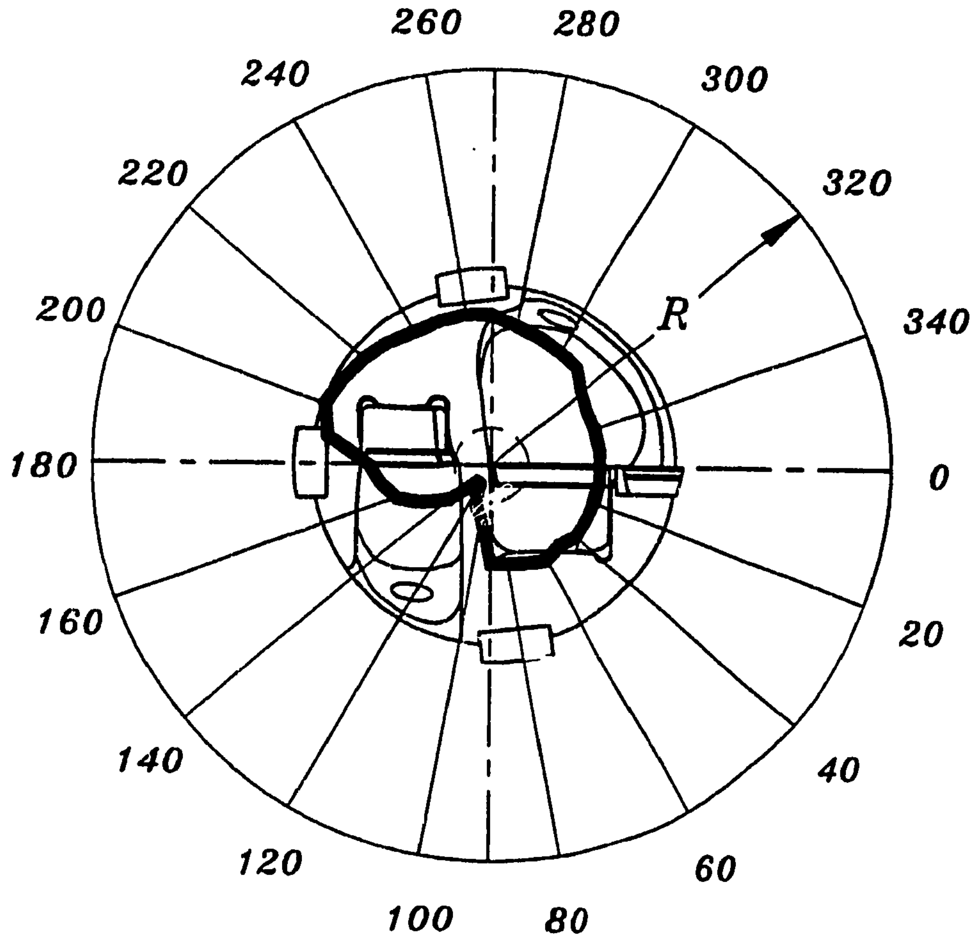
Figure 3.20: Static pressure distribution in the bottom clearance.



$R = 30KPa$

Drill 'A' - Transducer at r=10.6mm

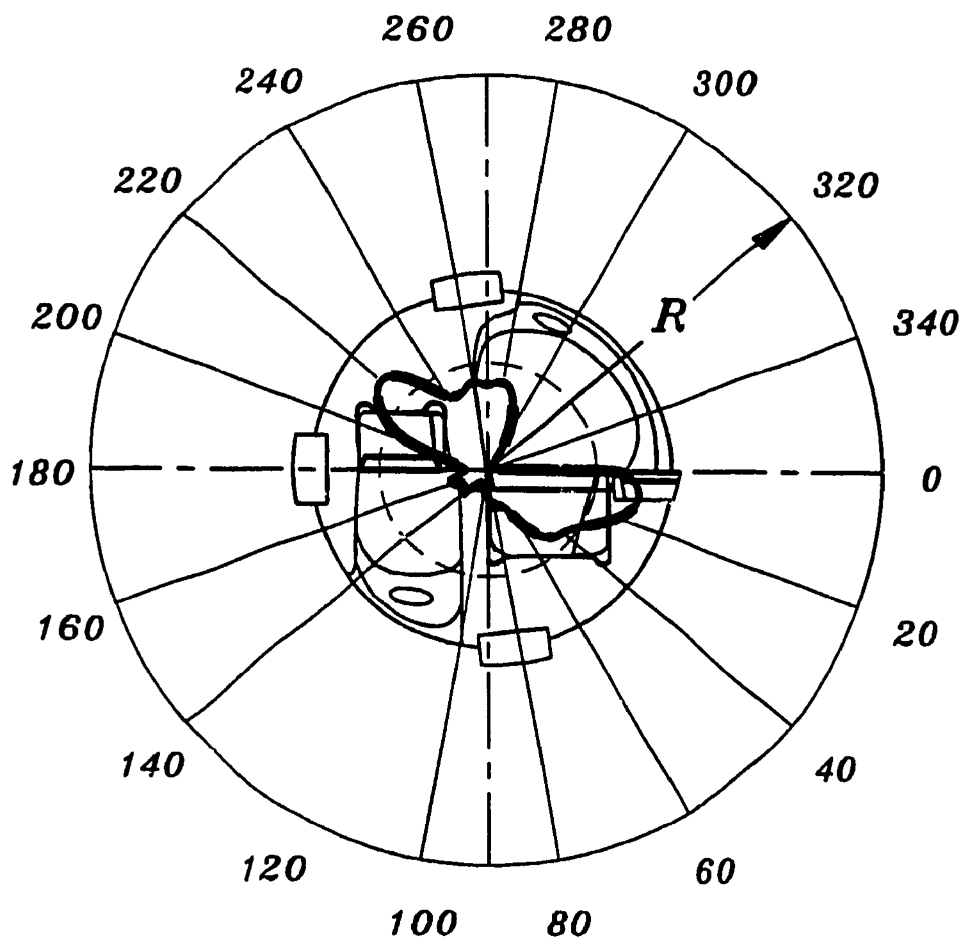
Figure 3.20: Static pressure distribution in the bottom clearance.



R=250 KPa

Drill 'B' - Transducer at r=6.5mm

Figure 3.20: Static pressure distribution in the bottom clearance.



R= 250KPa

Drill 'B' - Transducer at r=15.5mm

Figure 3.20: Static pressure distribution in the bottom clearance.

replaced with piezoelectric pressure transducers (DP15TL) and the signals from these transducers are recorded in a computer using a Data Acquisition System (OMEGA: DAS-20). An optical switch (OPB-804) is used to record each rotation of the boring bar from its initial position.

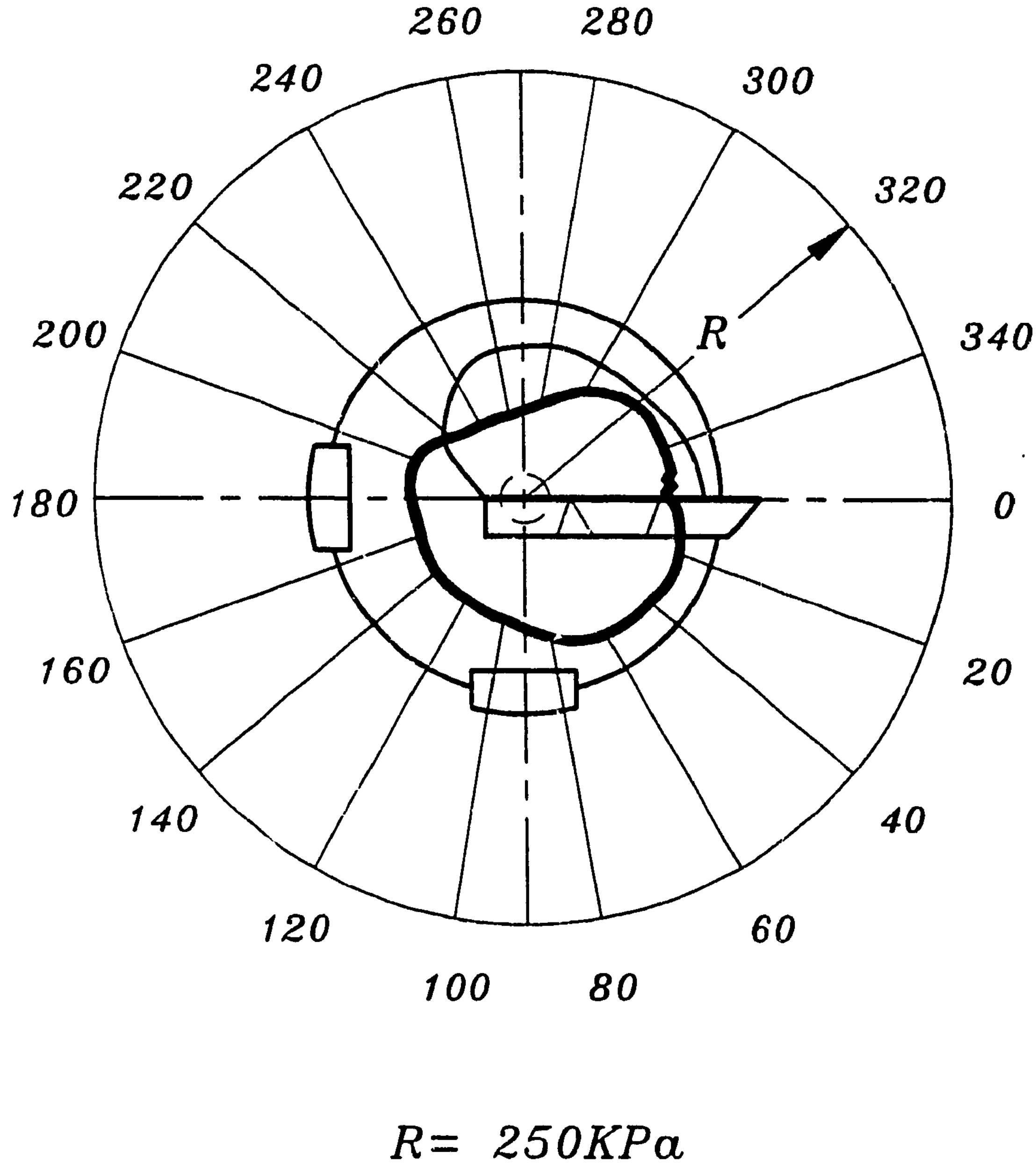
Entraped air from the feeders to the piezo electric transducers are removed using the bleed valves and the machine is run for couple of minutes for stabilizing. After the stabilizing period the voltage signals from the pressure transducer are amplified and multiplyed by a conversion factor to convert voltages into pressure reading in MPa. The collected data are then tabulated and plotted on a cycle graphs as shown in Figure 3.21.

B) Effect of feed:

The cutting tool is fed into the work piece at a rate of 0.05 to 0.1 mm/rev. The pressure readings are tabulatd and plotted on cycle graphs. Some of the plots are shown in Figure 3.22.

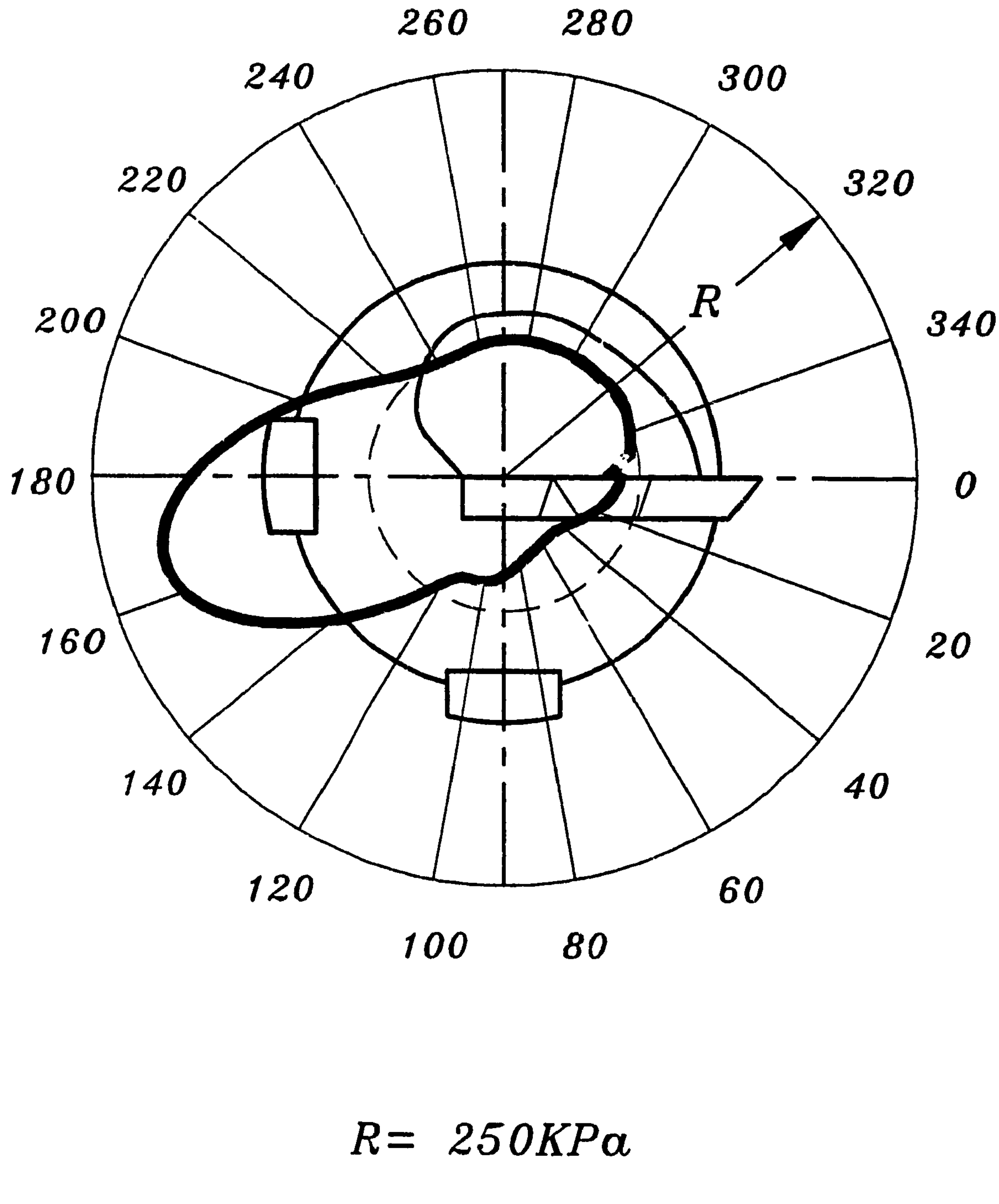
3.5.5 TOOL LIFE TEST

This test was designed to study the effect of coolant flow rates on the wear of the



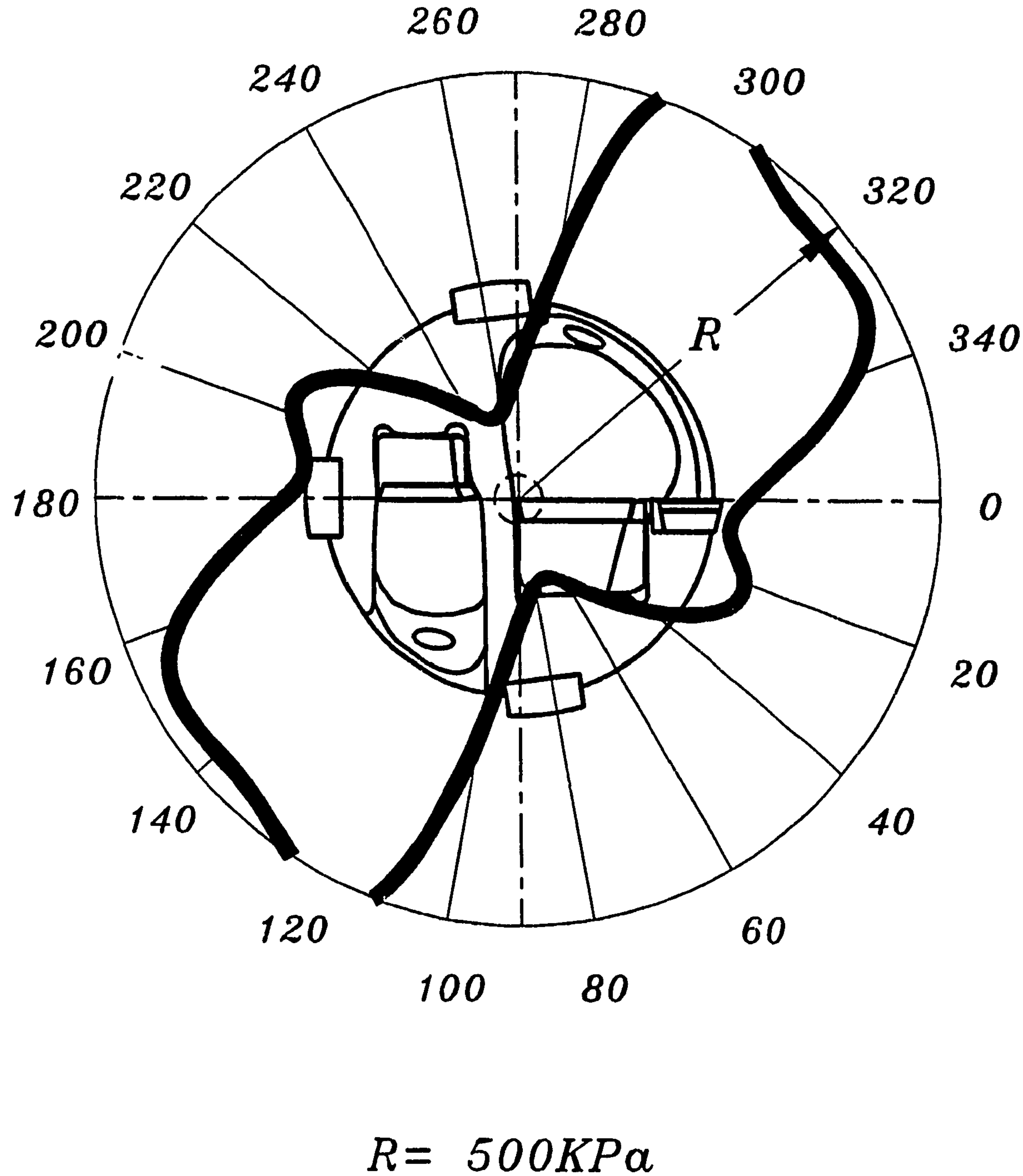
Drill 'A' - Transducer at r=3.1mm

Figure 3.21: Dynamic pressure distribution in the bottom clearance.



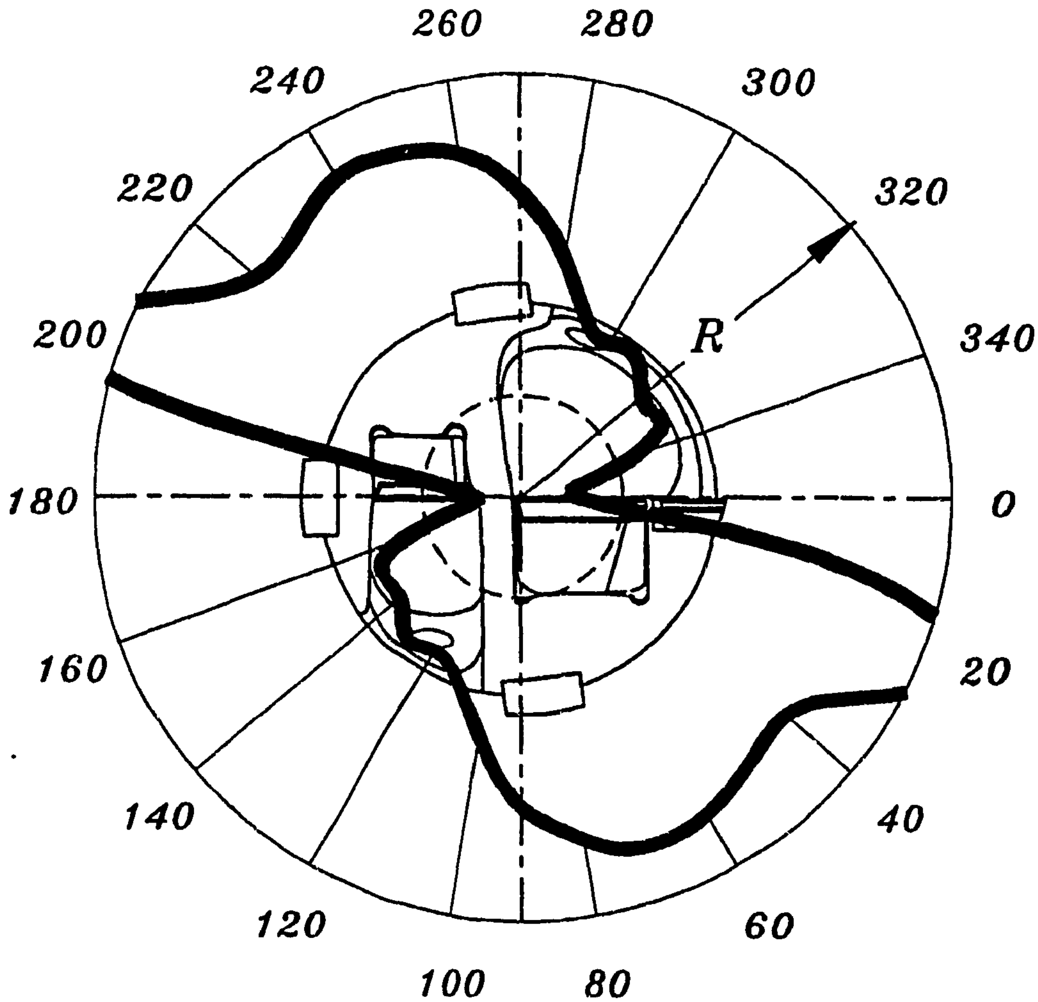
Drill 'A' - Transducer at r=13.1mm

Figure 3.21: Dynamic pressure distribution in the bottom clearance.



Drill 'B' - Transducer at r=3.1mm

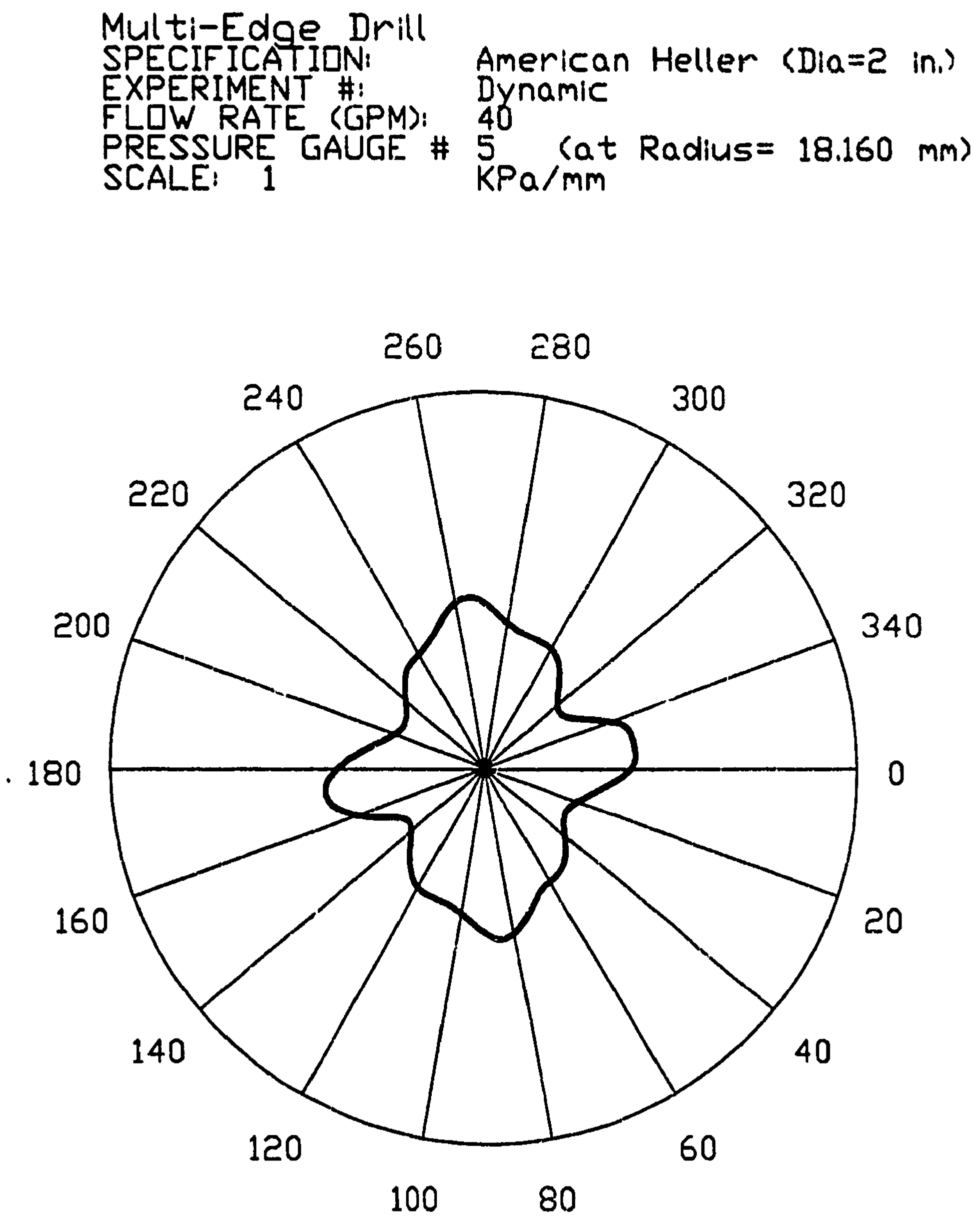
Figure 3.21: Dynamic pressure distribution in the bottom clearance.



R = 500*KPa*

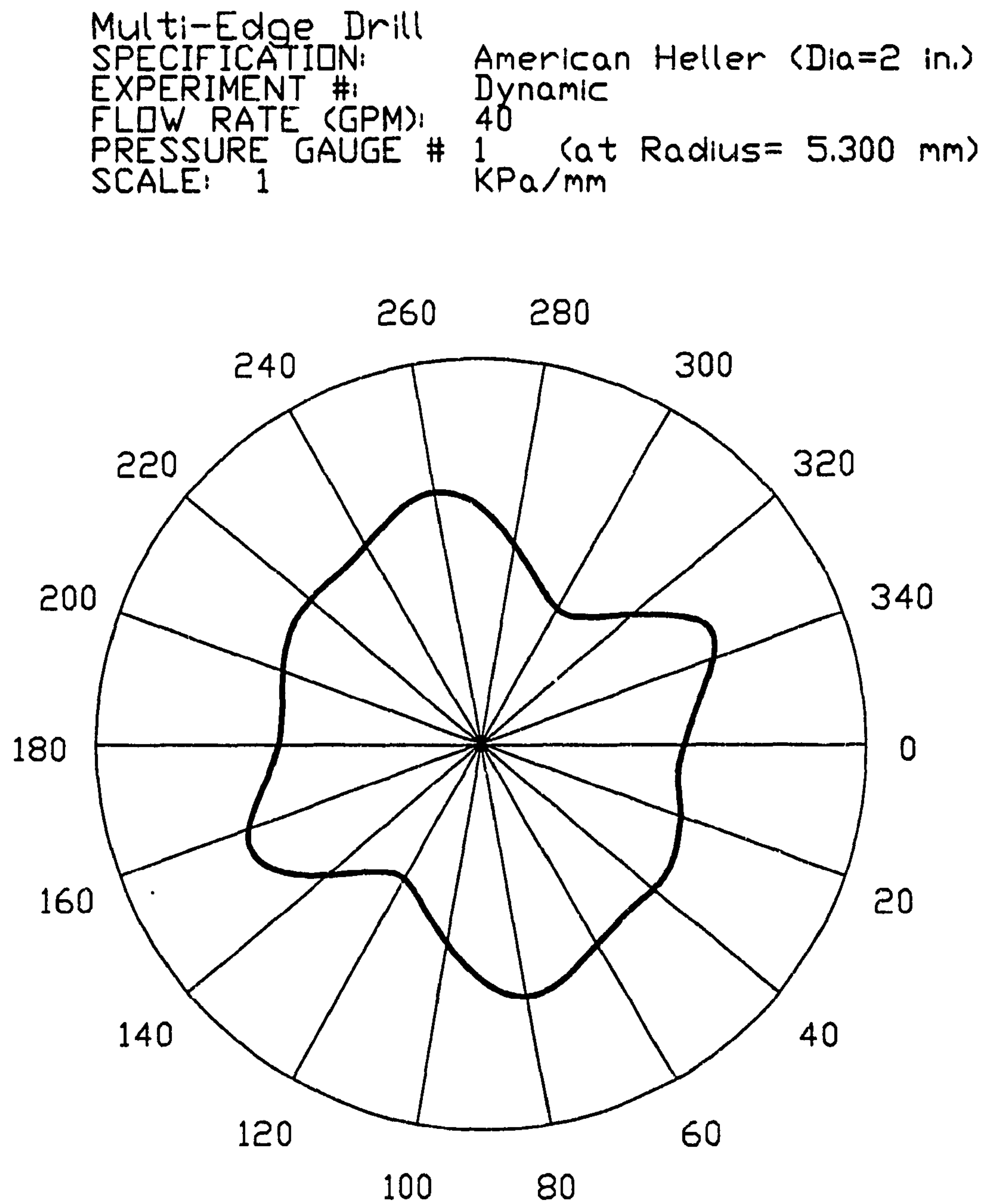
Drill 'B' - Transducer at r=13.1mm

Figure 3.21: Dynamic pressure distribution in the bottom clearance.



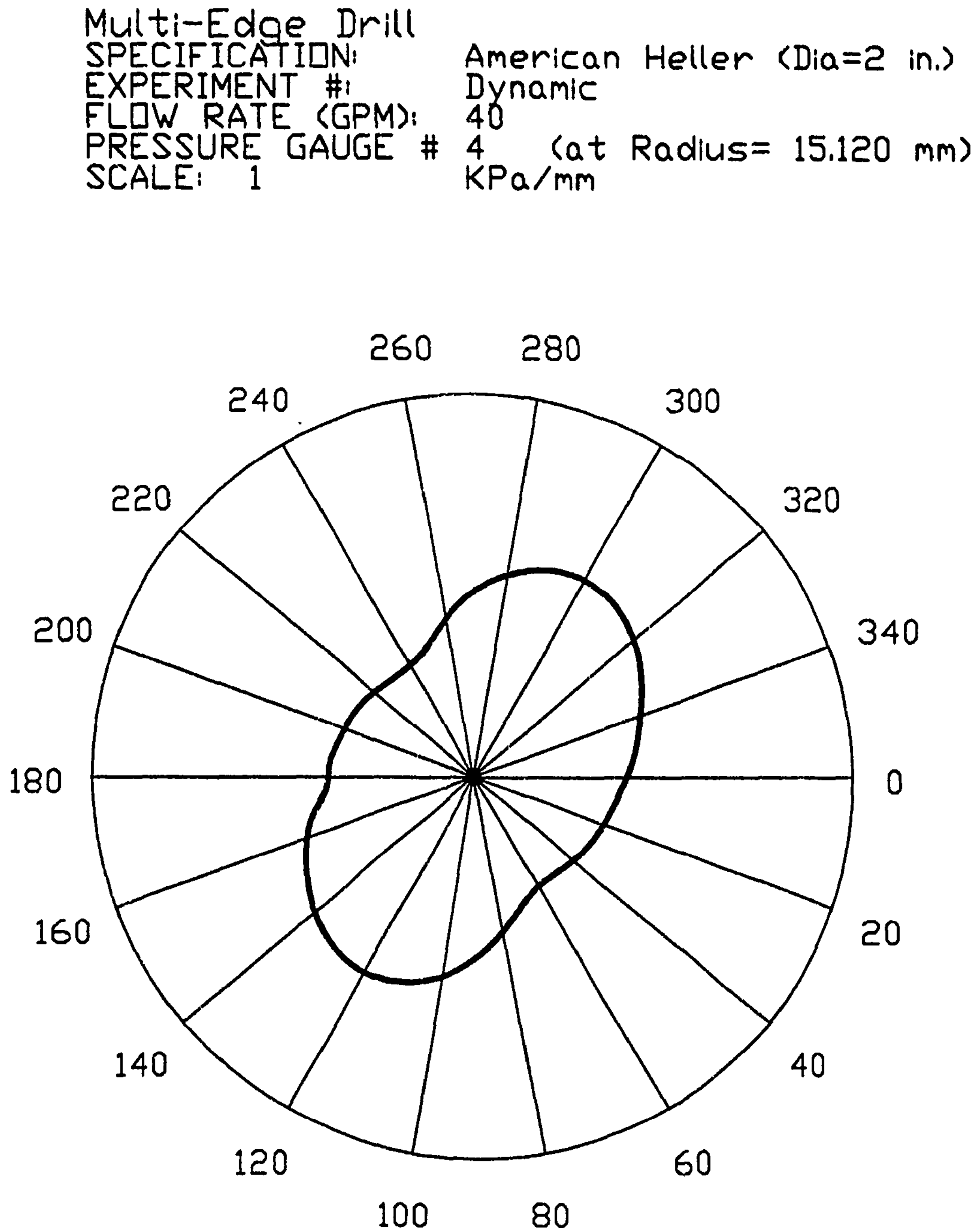
Feed = 0.05mm/rev.

Figure 3.22: Dynamic pressure distribution in the bottom clearance



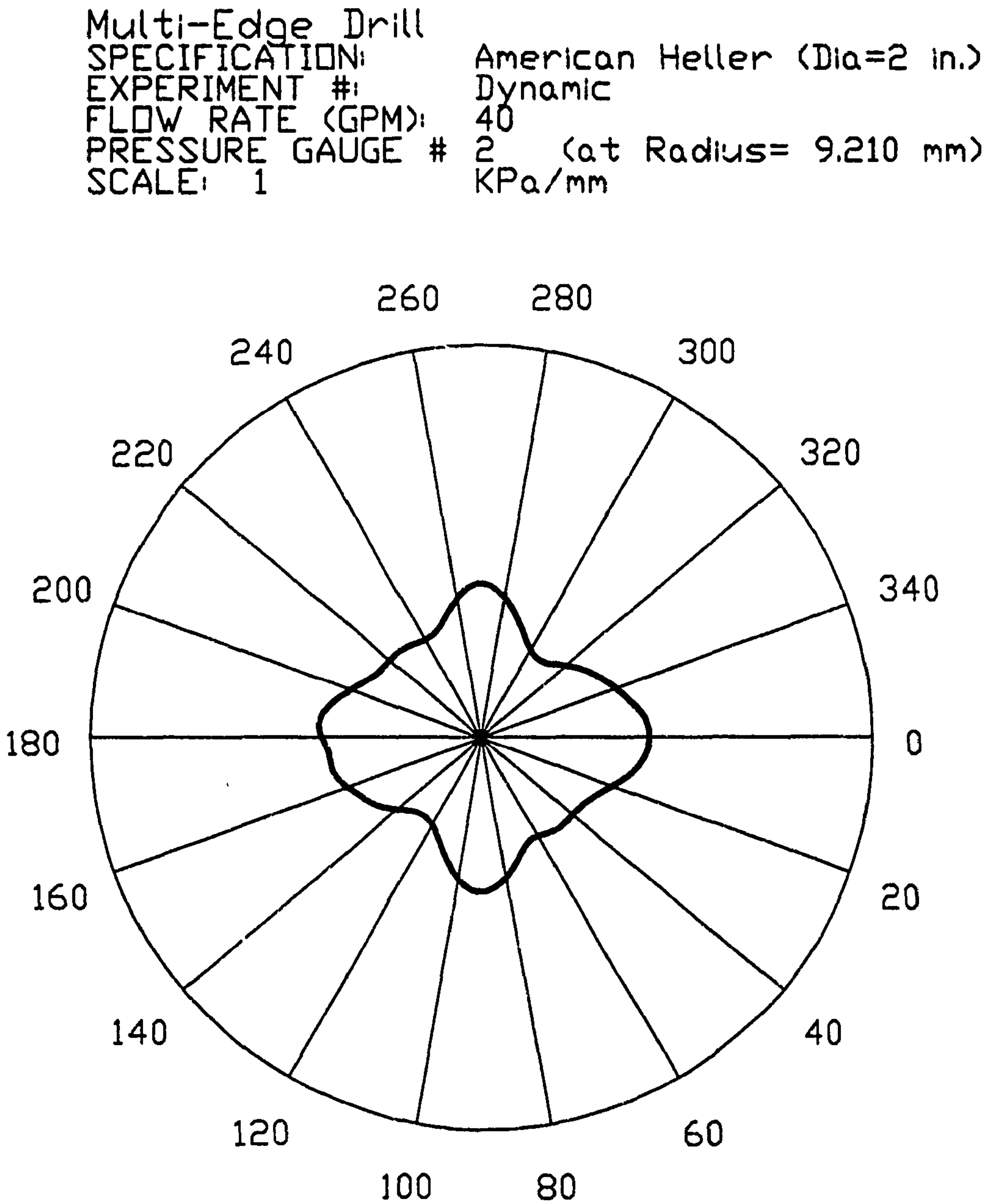
Feed = 0.05mm/rev.

Figure 3.22: Dynamic pressure distribution in the bottom clearance



Feed = 0.05mm/rev.

Figure 3..2: Dynamic pressure distribution in the bottom clearance



Feed = 0.05mm/rev.

Figure 3.22: Dynamic pressure distribution in the bottom clearance

deep hole drilling tools. A 22.225mm diameter solid BTA tool was used for testing the tool life. AISI 1045 steel was used as the workpiece material and 'Shell Garia H' was used as the coolant. Cutting tests were performed at a cutting speed of 75m/min and a feed of 0.1mm/rev. The workpiece held stationary and the rotating gundrill was fed into the workpiece. The tests were carried out for flow rates of $2x10^{-4}$ m^3/sec, $16x10^{-4}$ m^3/sec, and $12x10^{-4}$ m^3/sec. Width of the wear land was measured at regular intervals of cutting time and the results were plotted in Figure 3.23. Exper ·.ent was repeated and a maximum flank wear land width of 0.3mm was taken as the limiting wear for regrinding. Tool life is determined as time required to form a flank wear land of 0.3mm and the results are plotted in Figure 3.24.

Following observations were made from the recorded data.

1. The static as well as the dynamic pressure distribution in the machining zone are not uniform. Both distributions change significantly in the bottom clearance and there exist chambers with zero pressure (even vacuum chambers) in the machining zone under static and dynamic considerations.
2. The rotation of the tool leads to a significant change in the pressure distribution in the bottom clearance. The rotation makes the distribution more uniform in the case of drill "A" and less uniform in the case of drill "B".
3. Feeding the tool into the workpiece increases the uniformity of pressure

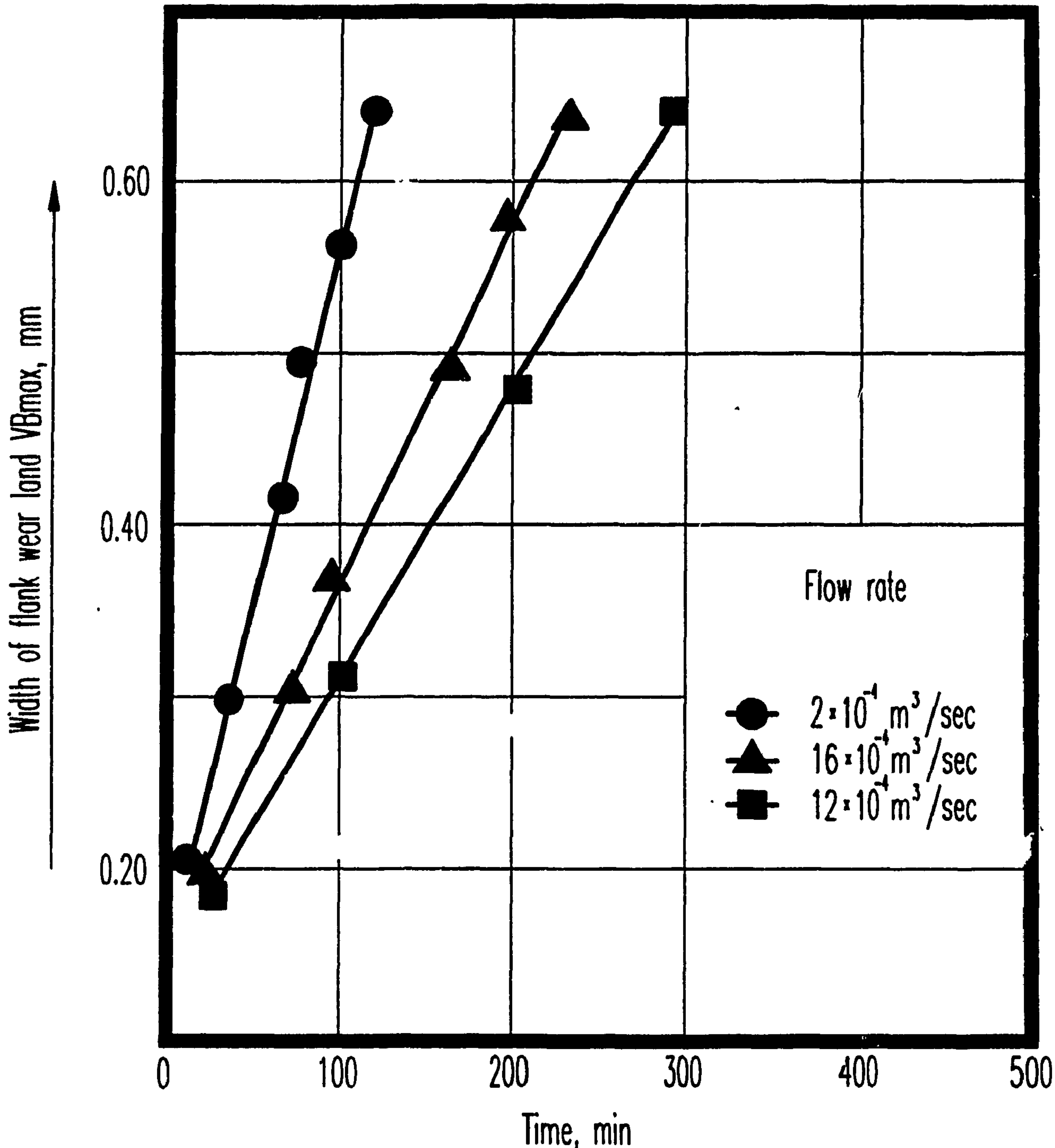


Figure 3.23: Relation between flank wear and coolant flow rate.

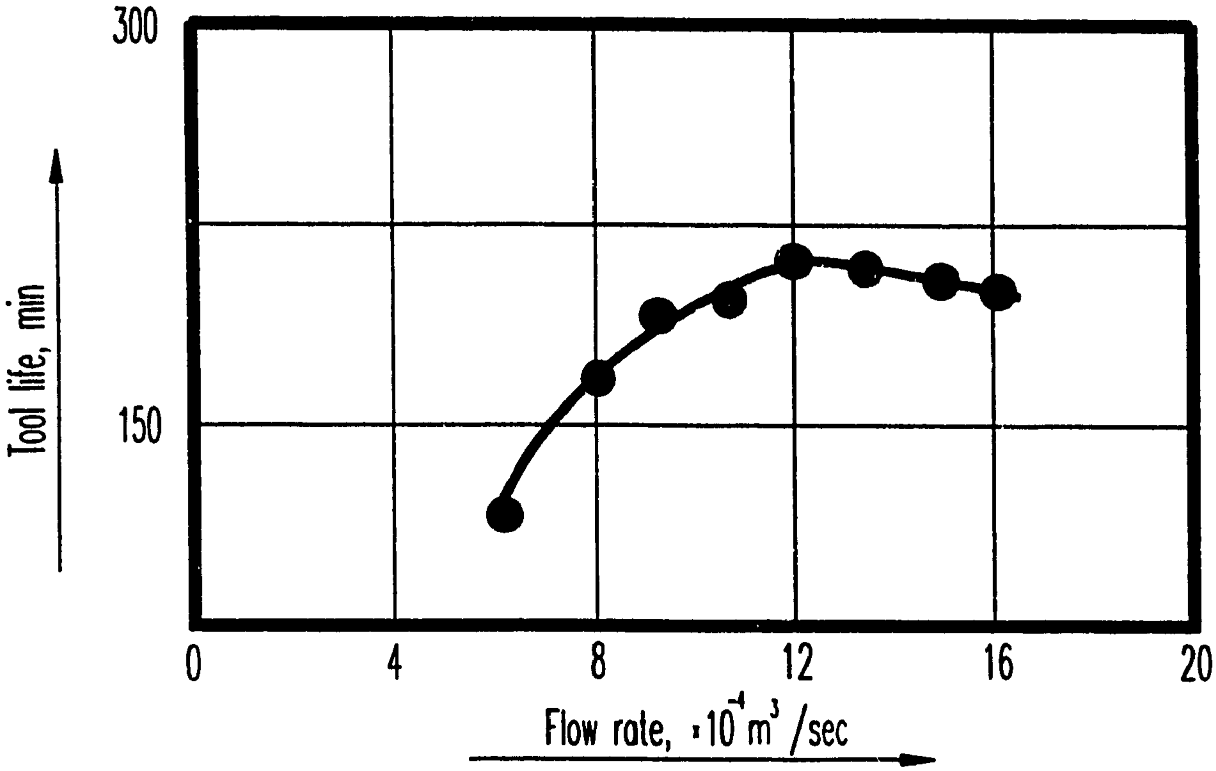


Figure 3.24: Relation between tool life and coolant flow rate.

distribution in the machining zone. Variation of feed rates has little effect on the pressure distribution in the bottom clearence.

4. The result of the experiment shows that drill "B" has a higher cutting fluid pressure in these areas and hence greater penetration of cutting fluid to the contact areas resulting in better cooling and lubrication of the contact areas. For drill "A", the zone of maximum pressure is shifted in the direction of the second supporting pad and hence plays no role improving the cooling and lubrication of the cutting edge.

5. The architecture of drill "A" provides better conditions for the chip removal process due to more uniform pressure distribution within the limits of the chip mouth. In the case of drill "B", sufficient conditions are achieved in the regions adjacent to the rake surface but close to the chip mouth limits the pressure is high. This phenemenon can be explained by the presence of the additional screw holes in the drill head, which are sources of additional cutting fluid jets directed toward the chip mouth. The existence of these jets can hinder the chip removal process along the chip mouth and throat, which is the most critical passage of the chip removal system.

6. The results also show that the architecture of the drill head can be changed in order to achieve better pressure distribution; avoid vacuum chambers in the bottom clearance; and provide the best conditions for cooling, lubrication and the chip removal.

3.5.6 CHIP REMOVAL

Reliable chip removal is one of the most important requirements of deep hole drilling. As mentioned above after performing its cooling and lubricating actions in the machining zone, the cutting fluid carries away the chips through the interior of the drill head and the boring bar. Therefore, past the machining zone, a two phase flow medium exists along the internal channels.

The most important factors affecting this two-phase flow medium are the following:

- The density and viscosity of the cutting fluid as a function of its temperature.
- The chip concentration (weight or volumetric), rheological properties and velocity of the cutting fluid-chip mixture (Swarf).
- The cross-sectional area and profile configuration of the chip removal channels.

Swarf flow through the chip removal channel may occur in different modes: heterogeneous with gliding chip layer, and pseud-ohomogeneous. Figure 3.25 illustrates the velocity and concentration profiles and the relationship between the swarf velocity and the friction loss along the chip removal channel for each mode.

The swarf velocity corresponding to the beginning of the heterogeneous mode is termed as "critical velocity" (v_{cr}). Our observations show that for if the mixture's

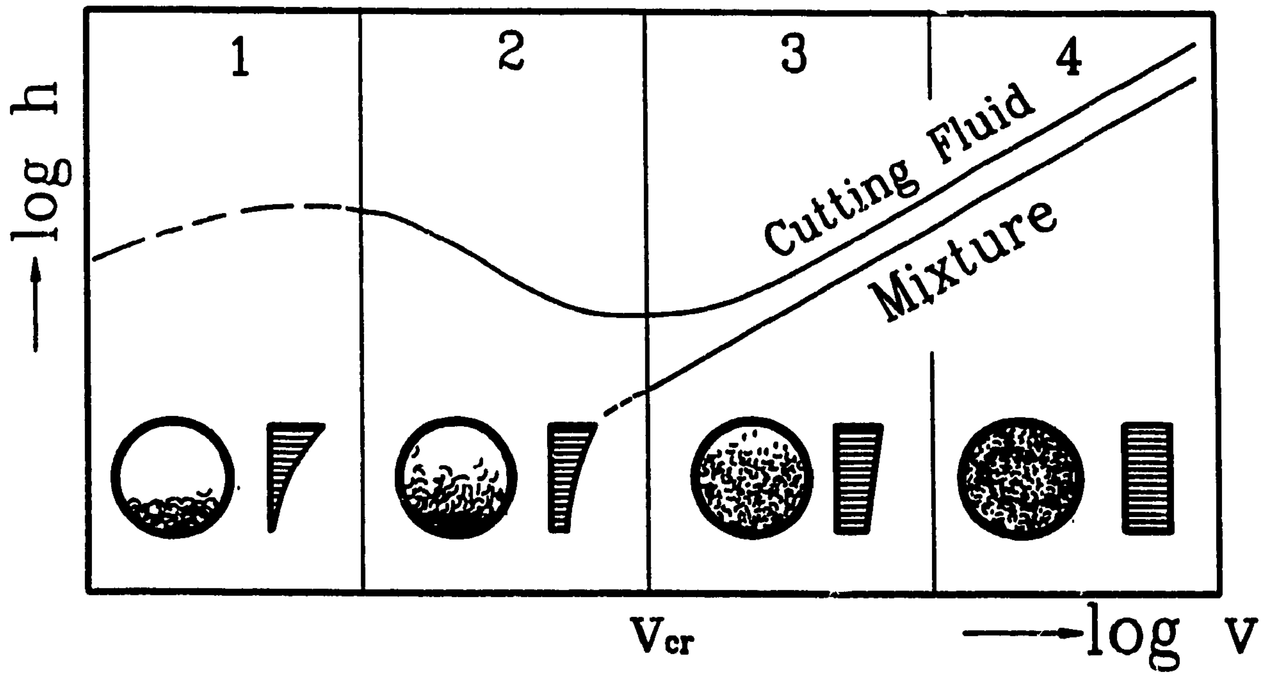


Figure 3.25: Friction loss along the chip removal channel for differentt swarf flow modes :

1. **Flow break with the stationary layer**
2. **Heterogeneous flow with the gliding layer**
3. **Heterogeneous**
4. **Pseudohomogeneous.**

velocity is less than v_{cr} then the formed chips accumulate in the boring bar thus blocking the chip removal process. As a result, the drilling process is forced to stop. Therefore reliable chip removal could be achieved when the swarf velocity in any cross section of the chip removal channels is more than the critical velocity.

The critical velocity of the swarf can be defined by the analysis of the three main components of the total swarf head loss, h_m,:

$$h_m = h_L + h_I + h_F \tag{3.1}$$

where h_L is the friction loss of the chip removal channels; h_I is the loss due to the cutting fluid-chip interaction under transportation; h_F is the loss connected with the energy necessary to maintain the chips in the flow conditions.

From the experiments carried out and the similarity theory applied to the pressure loss in the swarf flow, the following empirical formula for the critical velocity of the cutting fluid-chip mixture has been established [Gnatuk et al, 1989]:

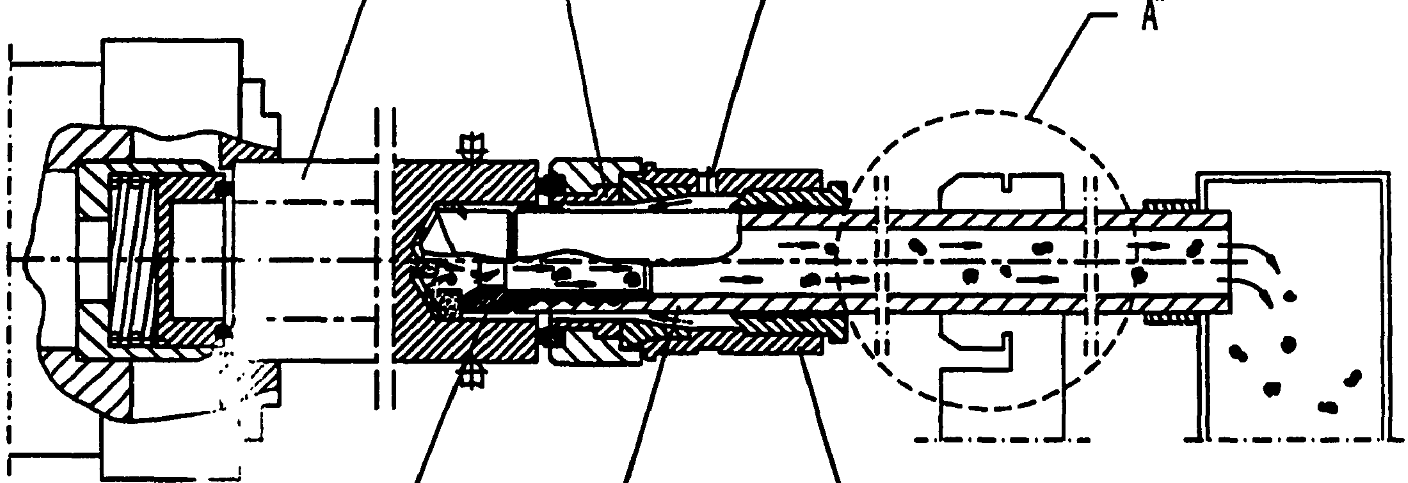
$$v_{cr} = 8.44 \left[c_w \left(\frac{\gamma_m}{\gamma_{cf}} - 1 \right) g \; v_c \; A_c^{1/2} \right]^{1/3} \tag{3.2}$$

Were, c_w is the weight concentration of the chip in the swarf; γ_m is the specific gravity of the chip's material; g is the gravitation constant; v_c is the free-fall velocity of the chip in the stable cutting fluid and A_c is the cross-sectional area of the chip removal channel.

From thhe experiments which are carried out to establish the conditions for reliable chip removal along the internal passage of the B.T.A and Ejector drills [Astakhov, 1982] it was observed that the swarf velocity of 15 to 21 percent higher than the critical velocity can be used to over come chip clogging problem.

$$v_{res} = (1.15 \ to \ 1.21)\, V_{cr} \tag{3.3}$$

In certain cases it is not possible to provide v_{res} for each passage in the whole chip removal system of the drilling machine. It is common for a small diameter of drills (6 - 30 mm), and in the case of necessity, to transport the mixture far away from the machine receiver. An additional flow of the cutting fluid should be introduced behind the tool-workpiece interface. In those cases, the total flow rate for the chip removal system can be determined in such a way that v_{res} (Q_0) is secured at any downstream section of this system (Figure 3.26). The additional flow of the cutting fluid Q_1 is fed through the barbed fitting 1 to the operating chamber 2. This flow passes through the ejector nozzle 3 to the mixing chamber 4. Due to this flow partial vacuum is created in chamber 5. The resulting vacuum sucks back the swarf along the boring bar. The combination of the suction and other additional flows takes place in the mixing chamber 4. At this point the resulting the swarf velocity exceeds the critical velocity.



Detail "A"

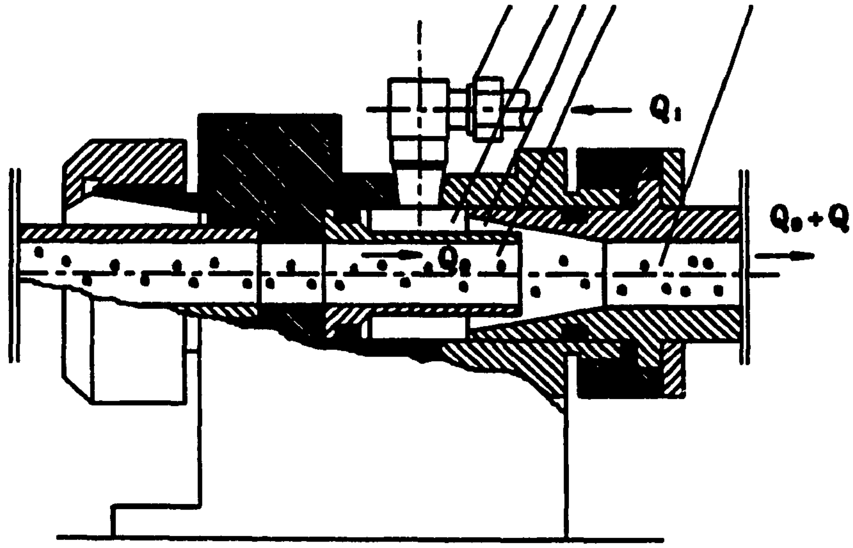


Figure 3.26: Introduction of additional flow in BTA using the ejector.

Due to this, the swarf can be transported as far as required by the drilling machine architecture and design. If Q is the swarf flow rate and Q_{add} is the additional flow induced downstreem, then the condition for reliable chip removal is:

$$\frac{Q + Q_{add}}{A_{max}} \geq v_{cr} \tag{3.4}$$

Where A_{max} is the cross-sectional area of the largest channel in the chip removal system.

3.5.7 PRESSURE DISTRIBUTION

In B.T.A and particularly in ejector drilling, the annular channels are narrow so that the rigidity of the boring bar is maintained high while allowing for maximum cross-sectional area of the internal chip removal channel. Therefore, the Navier-Stockes equations can be represented by only one equation for axial velocity of the coolant. The other velocity components are assumed insignificant or zero, i.e.,:

$$\frac{\partial^2 v_z}{\partial r^2} + \frac{1}{r}\frac{\partial v_z}{\partial r} = \frac{1}{\mu_c}\frac{dp}{dz} \tag{3.5}$$

with the following boundary conditions:

$$\begin{aligned} r &= R_1 \qquad v_z = 0\ ; \\ r &= R_2 \qquad v_z = 0\ . \end{aligned} \tag{3.6}$$

Solving equation(3.5) along with the continuity flow equation and considered boundary conditions (3.6), the following integral equation (in non-dimensional coordinates) for pressure distribution along the annular channel is obtained:

$$p = -12\mu_c L C_1 \int \frac{dz}{R_1 \delta_0^3} \pm 6\mu_c L \int \frac{dz}{\delta_0^2} + C_2 \; . \qquad (3.7)$$

Here, $\delta_0 = R_2 - R_1$; "±" Signifies the axial movement of one of the annular channel walls. "+" corresponds to coinciding flow and wall velocity, while "-" corresponds to velocities in opposite directions.

In reality, the inlet channels in deep hole drills, especially in the Ejector drills, are eccentric and the size varies along the length of the boring bar. If ε is the eccentricity of the annular channel (Figure.3.27), then the relative eccentricity is:

$$\bar{\varepsilon} = \frac{\varepsilon}{\delta_0} \; . \qquad (3.8)$$

If $\varepsilon = 0$ then $\varepsilon = 0$ and this case correspond to the nominal location of the annular channel's walls. If $\varepsilon = \delta_0$ then $\varepsilon = 1$ and this case correspond to the contact between the annular channel's walls. Thus, the variable clearance between channel's wall is:

$$\delta(z,\varepsilon,\theta) = (R_2 - R_1) - \varepsilon\cos\theta + (R_2 - R_1)\frac{z}{L} \qquad (3.9)$$

or in non-dimensional coordinates:

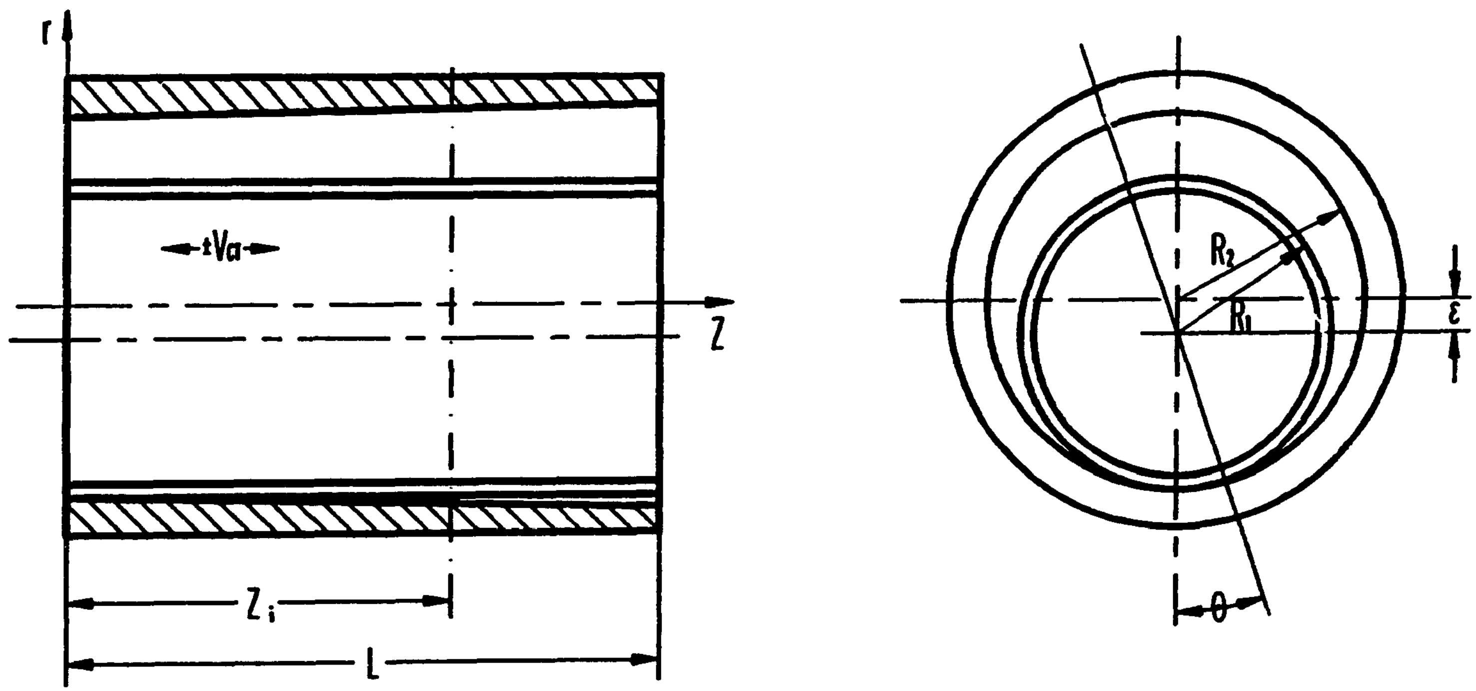


Figure 3.27: Cross section of the annular channel.

$$\delta(z,\bar{\varepsilon},\theta) = \delta_0(1 - \bar{\varepsilon}\cos\theta + k\bar{z}) \tag{3.10}$$

where k is:

$$k = \frac{\delta_{02} - \delta_{01}}{\delta_{01}} \ . \tag{3.11}$$

Where, δ_{01} and δ_{02} are the clearances in the inlet and outlet sections of the annular channel, respectively. Therefore, the equations of the channel's wall are:

$$R_1 = R_2 \ ; \tag{3.12}$$

$$R_2(z,\bar{\varepsilon},\theta) = \delta_0(z,\bar{\varepsilon},\theta) + R_1 \ . \tag{3.13}$$

Integrating (3.7) with respect to Eqs.(3.9), (3.12), and (3.13), we have:

$$p = \frac{6\mu_c L C_1}{R_1 \delta_{01}^3 k (C + k\bar{z})^2} \pm \frac{6\mu_c L V_z}{\delta_{01} k (C + k\bar{z})} + C_2 \ , \tag{3.14}$$

where C = 1 - $\varepsilon\cos\theta$; V_z is the axial velocity of the wall.

To define constants C_1 and C_2, the following boundary conditions can be used:

$$\begin{aligned} \bar{z} &= 0 \ , \qquad p = p_{in0} \ ; \\ \bar{z} &= 1 \ , \qquad p = p_1 \ . \end{aligned} \tag{3.15}$$

Then from equation(3.14) we get:

$$p = p_{in0} - \frac{(C - k)^2(2C\bar{z} + k\bar{z})^2}{(2C + k)(C + k\bar{z})^2} \pm A\frac{k\bar{z}(\bar{z} - 1)}{(2C + k)(C + k\bar{z})^2}, \tag{3.16}$$

where

$$A = \frac{6\mu_c V_z L}{\delta_{01}^2 \Delta p}. \tag{3.17}$$

Here, Δp is the pressure difference between inlet and outlet sections of the annular channel for stationary walls.

<u>Particular cases of the coolant flow:</u>

1. The annular channel's walls are cylinders with parallel axes, (k = 0), and stationary. Then from equation(3.16) we get the following expression for non-dimensional pressure:

$$p(\bar{z}) = p_{in0} - \bar{z}. \tag{3.18}$$

It is evident from equation(3.18) that the non-dimension coolant pressure varies linearly along the length of the channel irrespective of the eccentricity.

2. If the walls of the channel are relatively stationary, ($V_z = 0$) then ($k \neq 0$). In case of concentric walls (C = 1) from equation(3.16) we have:

$$p(\bar{z},k) = p_{in0} - \frac{(1-k)^2(2\bar{z}+k\bar{z}^2)}{(2+k)(1+k\bar{z})^2} \tag{3.19}$$

It can be seen from Figure 3.28 that the pressure drop increases significantly with the increasing k. In case of the eccentric walls ($\varepsilon \neq 0$), we have:

$$p = p_{in0} - \frac{(C - k)^2(2C\bar{z} + k\bar{z})^2}{(2C + k)(C + k\bar{z})^2} \tag{3.20}$$

The coolant flow rate per unit width of the flow can be defined as:

$$Q_u = \int_0^{\delta_0} V_z dz = \frac{1}{2\mu_c}\frac{dp}{dr}\delta_0^3 \pm \frac{V_z\delta_0}{2} = \frac{p_{in0}\delta_0^3}{12\mu_c L} \pm \frac{V_z\delta_0}{2} \; . \tag{3.21}$$

The total flow rate in case of the concentric and stationary walls ($\varepsilon = 0$) is:

$$Q_c = \int_0^{2\pi} Q_u R_1 d\theta = 2\pi\frac{p_{in0}\delta_0^3}{12\mu_c L}R_1 \; . \tag{3.22}$$

In eccentric walls ($\varepsilon = 1$), the equation for varying channel clearance is:

$$\delta = \delta_0(1 - \cos\theta) \; , \tag{3.23}$$

and the full coolant flow rate can be given by:

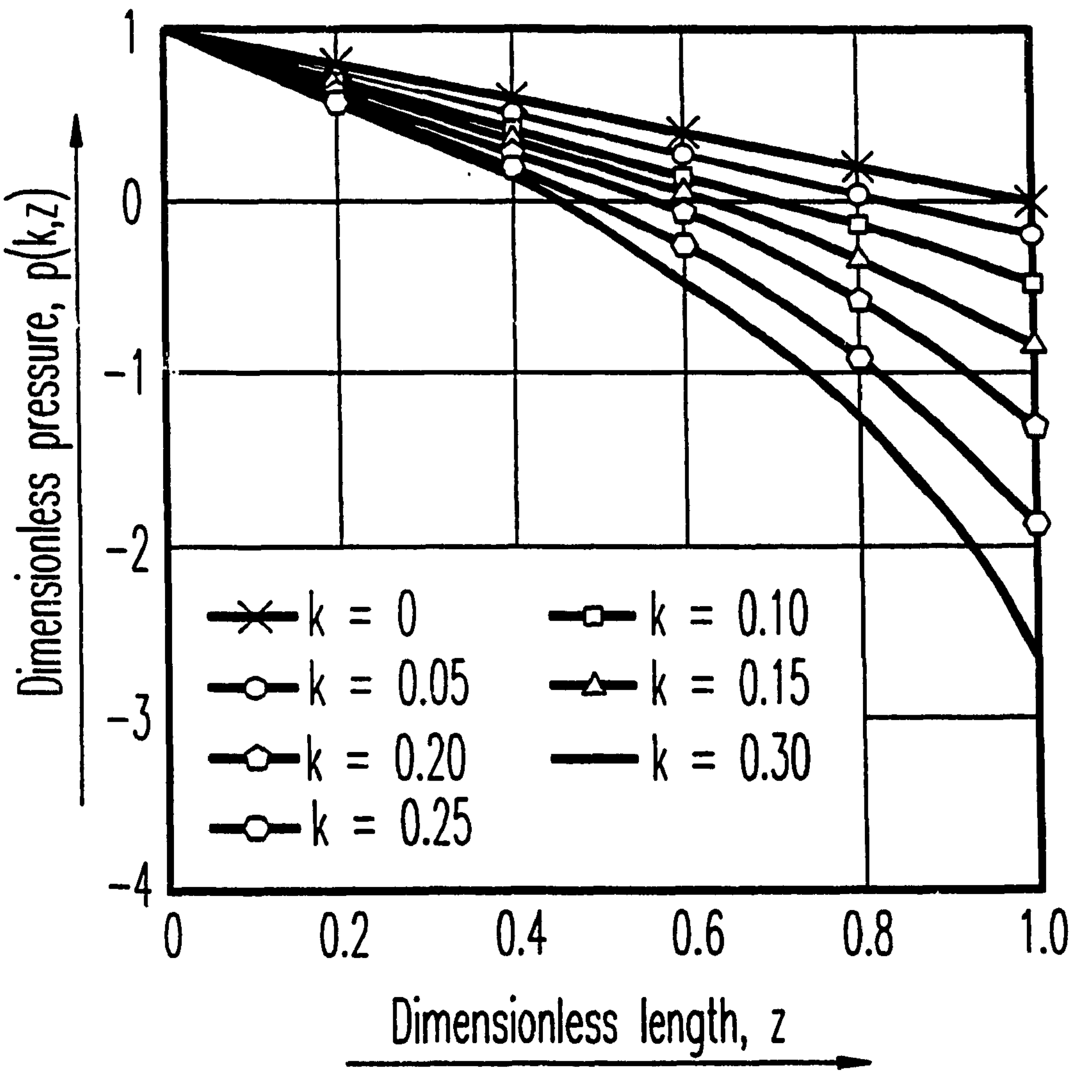


Figure 3.28: Pressure distribution along the annular channel.

$$Q_l = \int_0^{2\pi} Q_u R_1 d\theta =$$
$$\frac{p_{in0}\delta_0^3}{12\mu_c}\int_0^{2\pi}(1 - \cos\theta)R_1 d\theta = \frac{5\pi p_{in0}\delta_0^3}{12\mu_c L}R \; . \tag{3.24}$$

The comparison of equations (3.21) and (3.24) shows that for maximum eccentricity (ε = 1), the flow rate through the annular channel is 2.5 times flow rate in the concentric annular channel under the same inlet pressure p_{in0}. In practice, due to the additional hydraulic resistance at the inlet and outlet, the ratio of flow rates is [Astakhov, 1982]:

$$\frac{Q_c}{Q_l} = 1.7 \; to \; 1.9 \tag{3.25}$$

For example, the Ejector drill with eccentric inlet annular channel is shown in Figure 3.29. Inner tube 1 is eccentric relative to boring bar 2. The coolant is pumped to drill head 3 through eccentric annular channel 4 and holes 5. The eccentricity of the inner tube location corresponds to a larger chip mouth 6 of the drill head. Ejector nozzles, 7, are located in the inner tube.

3.5.8 STABILITY OF COOLANT FLOW

A detailed study of the coolant flow in B.T.A. drilling shows that, under certain

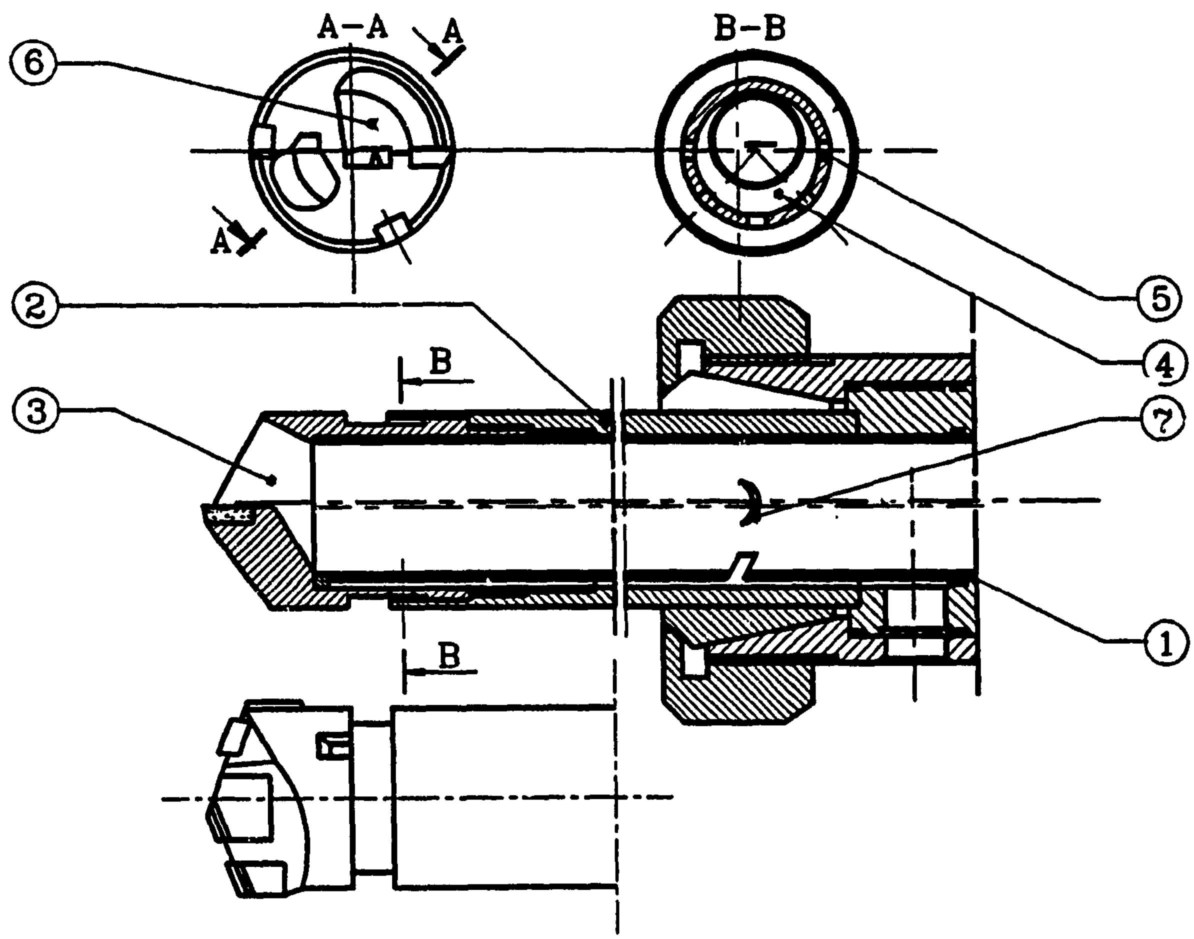


Figure 3.29: Ejector drill with eccentric annular channel.

combination of the rotation speeds (workpiece or boring bar), the clearance between the annular channel walls, and the flow rate, the coolant flow becomes unstable. The instability appears in the form of increasing or decreasing necessary pressure to achieve the nominal flow rate, the additional coolant heating, etc., [Astakhov and Scorupco, 1982; Astakhov, 1984]. The cause of the instability is the formation of macro-vortices (the vortices of Taylor) due to the rotation of one of the channel walls. The shape of macro-vortices depends upon various parameters namely the flow regime (Reynolds number), the temperature gradient, the channel's walls eccentricity, etc. Basic shape of the vortices is shown in Figure 3.30.

Based on the theoretical consideration, Taylor (1963) obtained the similarity number in the following form:

$$Ta = \frac{\omega_1}{\nu_c} R_1^{\frac{1}{2}} (R_2 - R_1)^{\frac{3}{2}} , \qquad (3.26)$$

where ω_1 is the angular velocity of the rotating wall

In case of rotating boring bar the velocity is:

$$V_b = \omega_1 R_1 , \qquad (3.27)$$

then:

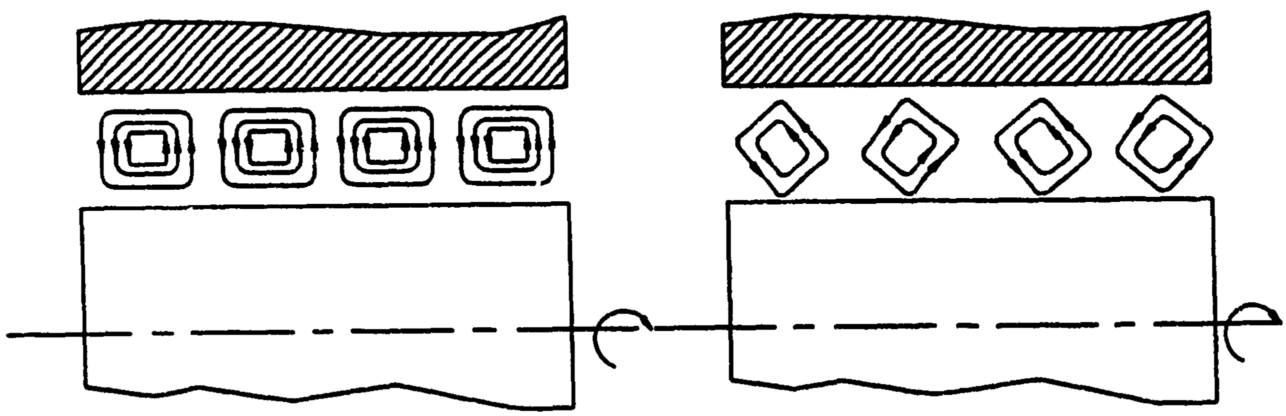


Figure 3.30: Basic shape of the Taylors macro-vortices.
(a) Without axial coolant velocity, (b) With axial coolant velocity.

$$Ta = \frac{V_b}{\nu_c}(R_2 - R_1)\left(\frac{R_2 - R_1}{R_1}\right)^{\frac{1}{2}} . \tag{3.28}$$

To estimate the influence of the boring bar rotation and axial coolant movement on the macro-vortices, the expression for the Taylor number can be represented in the following form (Coney, and Simmers, 1979):

$$Ta = Re_e\left(\frac{R_2 - R_1}{R_1}\right)^{1/2} . \tag{3.29}$$

Where, Re_e is the effective Reynolds number:

$$Re_e = \frac{2(R_2 - R_1)}{\nu_c}V_l , \tag{3.30}$$

and V_l is the effective coolant velocity:

$$V_l = (V_z^2 + (V_b \cdot k_1)^2)^{\frac{1}{2}} , \tag{3.31}$$

where k_1 is the ratio of the rotating speeds of the boring bar and the flow. For oil based coolants k_1 = 0.05 to 0.775.

The experimental results show that the axial flow velocity has a major influence on the flow conditions in the annular inlet channels. Under laminar flow mode, that is, under a small axial flow velocity, increase in the angular velocity of the boring bar causes the flow to change to the laminar mode with macro-vortices. Under constant axial velocity of the coolant flow, further increase in the boring bar angular velocity results in sudden change of flow to the turbulent mode with macro-vortices (without taking the pure turbulent mode). This leads to a significant increase in the hydraulic resistance of the inlet annular channel. The amount of the coolant reaching the cutting zone reduces (under constant inlet pressure). Under relatively high axial velocity of the coolant flow in the annular channel, the increase in angular boring bar velocity results in the coolant flow changing to turbulent mode and then to the turbulent with the macro-vortices (Figure 3.31). Under this mode, the heat exchange between inlet and outlet coolant and the hydraulic resistance of the channel depend upon the angular velocity of the boring bar alone. Coney and Simmers [1979] have shown that this mode takes place when $Ta \geq 41.3$.

The stability of the non-isothermal coolant flow through the annular channel with rotating wall(s) can be estimated by solving the equations of motion in Bussinesk's approach [Astakhov, 1990]. Let ω_1 and ω_2 are the angular velocities of the inner and outer channel's walls, respectively, and Θ_1 and Θ_2 are it temperatures. Assuming that other mass forces are absent. Then the equations of the coolant motion in non-dimensional cylindrical coordinates (r, φ, z) provide the exact solution in the following form [Gnatuk, Astakhov, and Chegodar, 1989]:

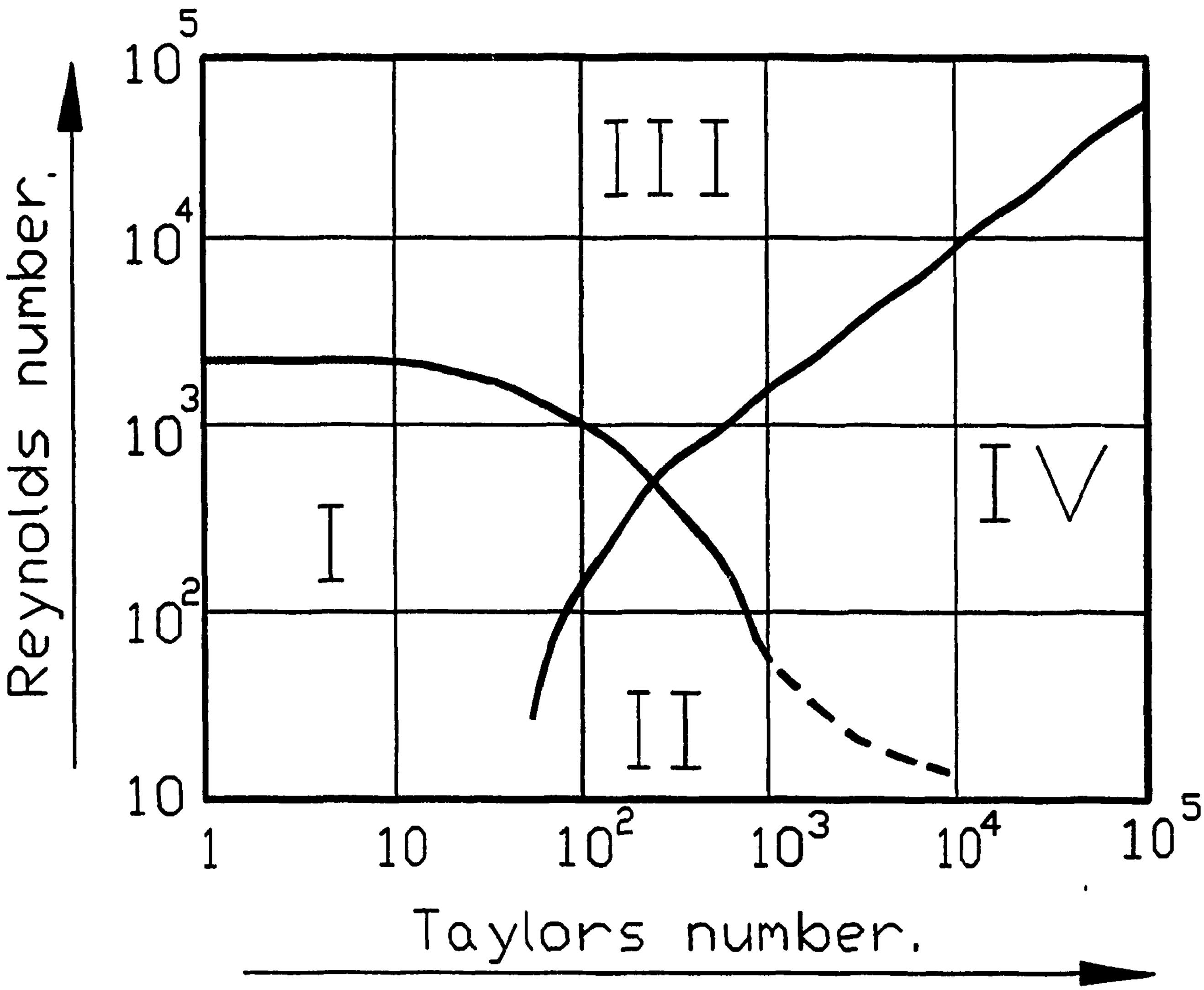


Figure 3.31: Coolant flow region boundary.

I - Laminar flow

II - Laminar flow with macro-vortices.

III - Turbulent flow.

IV - Turbulent flow with macro-vortices.

$$
\begin{aligned}
V_0 &= \{0, V_0(r), 0\} ; \\
\Theta_0 &= c \cdot ln \cdot r + 1 ; \\
p_0 &= \int_1^r \frac{V_0(s)}{s} \rho_0(s) ds + c ; \\
\rho_0 &= 1 + Cr \cdot lnr ; \\
V_0(r) &= k_0 r + \frac{m}{r} ; \; k_0 = \frac{\omega R^2 - 1}{R^2 - 1} , \; m = 1 - k ; \\
c &= \frac{\omega - 1}{lnR} ; \; R = \frac{R_2}{R_1} ; \; \omega = \frac{\omega_2}{\omega_1} ; \; \Theta_0 = \frac{\Theta_1}{\Theta_2} ,
\end{aligned}
\tag{3.32}
$$

where Cr is the Grashof number.

To define the instability regimes, the infinitesimal monotonic $2\pi/\alpha$ - periodic perturbations in the axial direction are applied to this non-isothermal flow:

$$
\begin{aligned}
V_r &= u(r)\cos\overline{\alpha z} ; \\
V_\varphi &= v(r)\cos\overline{\alpha z} ; \\
V_z &= w(r)\cos\overline{\alpha z} ; \\
\overline{\Theta} &= c_0 Pr\tau(r)\cos\overline{\alpha z} ; \\
\overline{p} &= q(r)\left(\frac{\overline{\alpha z}}{Re}\right) ; \\
\overline{\rho} &= GrPr\tau(r)\cos\overline{\alpha z} .
\end{aligned}
\tag{3.33}
$$

The problem is to find the critical Reynolds numbers and corresponding perturbation amplitudes. After the linearization and the separation of the variables:

$$\begin{aligned}
&\left(\frac{d}{dr} + \frac{1}{r}Rl\right)u + \overline{\alpha}^2 w = \frac{Gr}{r\rho_0}u ;\\
&\left(\frac{d^2}{dr^2} + \frac{1}{r}\frac{d}{dr} - \frac{1}{r^2} - \overline{\alpha^2}\right)u = \frac{1}{\rho_0}\frac{dq}{dr} + \\
&\qquad \frac{\lambda_c V_0}{r}\left(\frac{V_0}{\rho_0}GrPr\tau - 2v\right) ;\\
&\left(\frac{d^2}{dr^2} + \frac{1}{r}\frac{d}{dr} - \frac{1}{r^2} - \overline{\alpha^2}\right)v = 2k\lambda_c u ;\\
&\left(\frac{d^2}{dr^2} + \frac{1}{r}\frac{d}{dr} - \overline{\alpha^2}\right)w = -\frac{\overline{\alpha}}{\rho_0^2}q ;\\
&\left(\frac{d^2}{dr^2} + \frac{1}{r}\frac{d}{dr} - \overline{\alpha^2}\right)\tau = \frac{\lambda_c}{r}u ;\\
&u = v = w = \tau = 0 \quad when \quad r = 1, R .
\end{aligned} \tag{3.34}$$

The problem (Equations.(3.34)) was converted to the boundary problem for the system of eight conventional differential equations of the first order, which was solved by using the REOM software. The numerical minimization of Re_{cr} by the wave number α was used to find the critical regimes. The results of calculations of the most dangerous wave numbers α_* and corresponding to it Re_{cr} under different Grasthov numbers for the most common case of the B.T.A. drilling (R = 1.2) are shown in Table 3.2. To maintain the stable conditions of coolant supply, it is necessary to avoid the coolant flow rate corresponding to Re_{cr}.

From experimental results it was found out that the hydraulic resistance varies directly with boring bar velocity. The following experimental formula defines the Darsy

Gr	α	Re_{cr}
-1.0	4.95	574
-0.9	4.60	469
-0.8	4.31	378
-0.7	4.08	301
-0.6	3.79	234
-0.5	3.36	171
0.0	3.16	68
0.5	3.19	48
1.0	3.19	40
1.4	3.20	36

Table.3.2 The dependance of the most dangerous wave numbers α and its corresponding Re_{cr} from Grashof numbers Gr.

coefficient depending upon the boring bar velocity:

$$\lambda_R = \lambda_{ag}\left[1 + 0.5\left(\frac{V_b}{V_z}\right)^2\right]^{0.535} \tag{3.35}$$

where λ_{ag} is Darsy coefficient for the annular channels, which can be defined from Figure 3.32. Darsy coefficient depends to a large degree on the coolant viscosity. For oil-based coolant the viscosity is very sensitive to the temperature. This makes it necessary to investigate the heat exchange, between the swarf and the incoming coolant. The experimental results show the following:

1. The heat transfer from the swarf to the incoming coolant can be given as:

$$Nu = 0.092\left(Ta^2\ Pr\right)^{\frac{1}{3}} . \tag{3.36}$$

Considering one dimensional flow and using the expressions for dimensionless numbers [Grosse, 1986] we have:

$$\frac{\alpha L}{\lambda_c} = 0.015\left[1 + \frac{4.6(R_2 - R_1)}{L}\right] R^{0.45}\left(\frac{V_z L}{\mu_c}\right)^{0.80}\left(\frac{c_p \mu_c}{\lambda_c}\right)^{0.33} . \tag{3.37}$$

2. The presence of the Taylor vortices in the annular inlet channels results in increase in the convection coefficient 2 - 3 times in comparison with

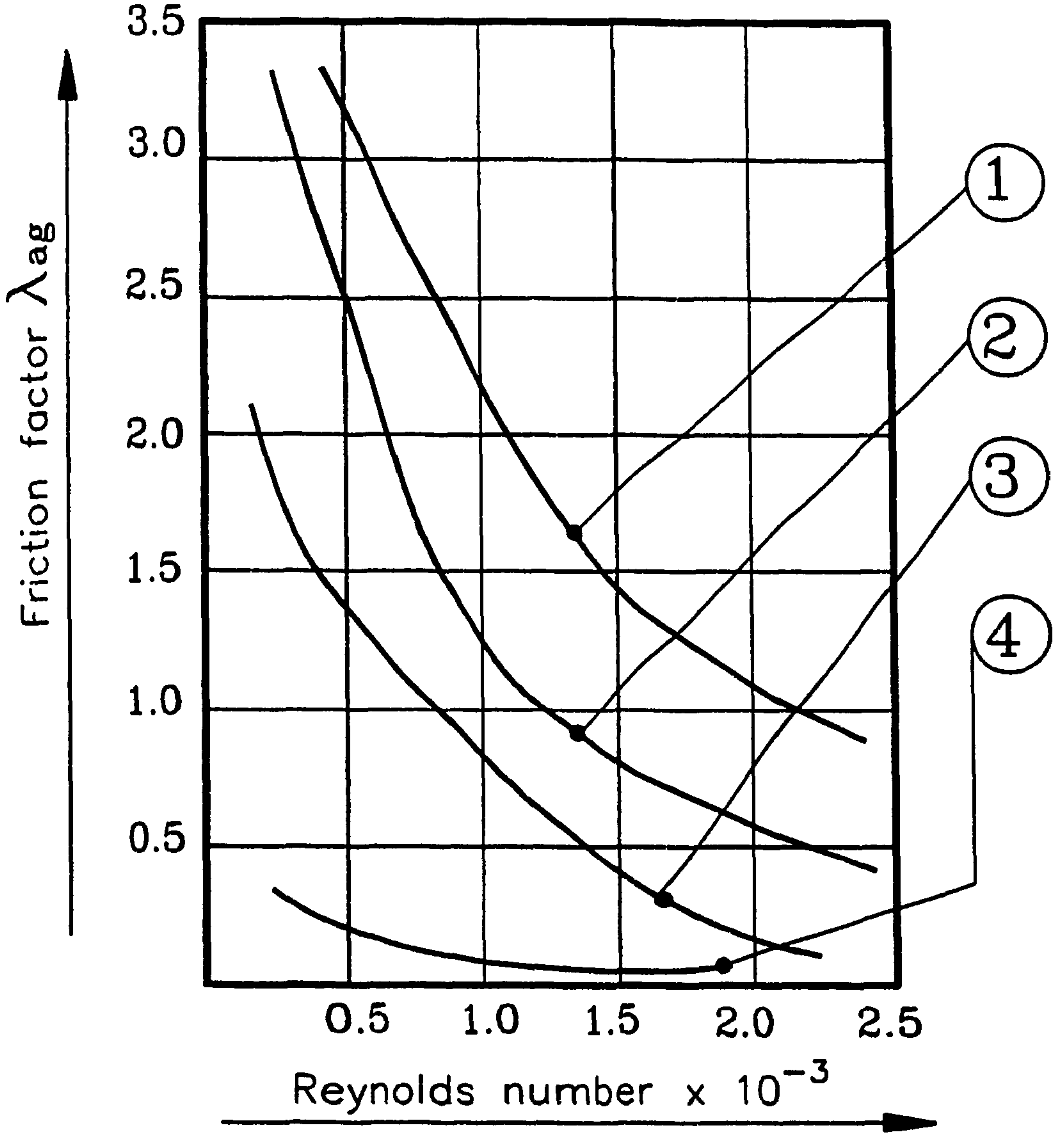


Figure 3.32: Darsy coefficients Vs Reynolds number for inlet annular channels.

1 - Oil, v_{50} = 0.17 * 10^{-4} m^2/s

II - Oil-based (MR3), v_{50} = 0.17 * 10^{-4} m^2/s

III - Oil-based (Shell Garia H)

IV - Water based coolant

that of the laminar flow.

3. The presence of the Taylor vortices under the turbulent flow mode results in increase in the convection coefficient by 24 - 25%.

4. The presence of the Taylor vortices under the developed turbulent flow does not affect the convection coefficient.

3.5.9 PRESSURE DISTRIBUTION AND FLOW IN THE EJECTOR DRILLS

The schematic diagram of the ejector drill is shown in Figure 3.33. Inlet coolant annular clearance can be classified into 3 regions: 1) AB is the part from inlet section (corresponds to point A) to the ejector nozzle (point B); 2) BD is the part corresponding to the nozzle, which connects the annular channel (D_b - d_{b2}) with the chip removal channel; 3) BC is the part from the ejector nozzle (point B) to the exit section of the outlet holes of the drill head. (point C).

The equations for the hydraulic heads for these parts can be given as:

$$H_{AB} = (z_B - z_A) + S_{AB}Q^2 \; ; \tag{3.38}$$

$$H_{BC} = (z_C - z_B) + S_{BC}Q^2 \; ; \tag{3.39}$$

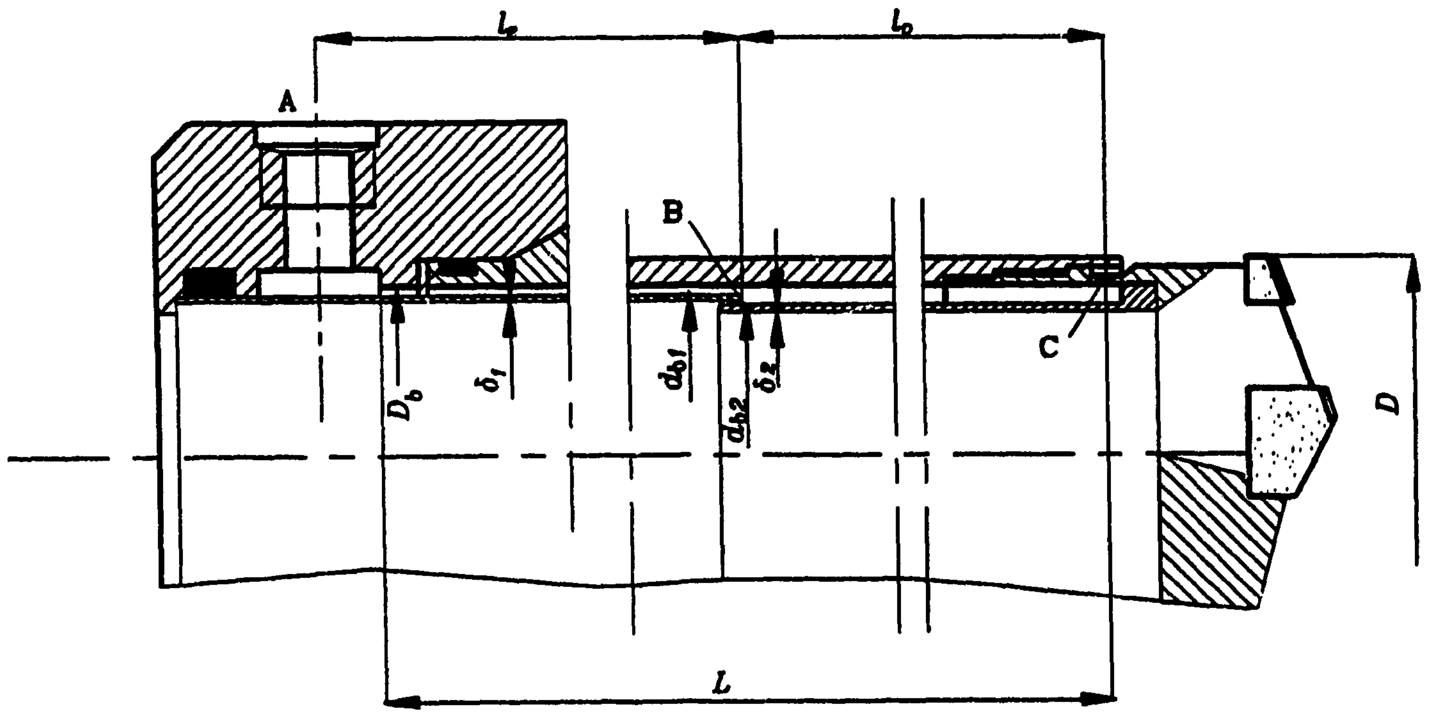


Figure 3.33: Schematic cross section of an ejector drill.

$$H_{BD} = S_{BD}Q^2 \,, \tag{3.40}$$

where z_A, z_B, and z_C are the elevations of the points A, B, and C, respectively, from the datum, m; S_{AB}, S_{BC}, and S_{BD} are the hydraulic resistance coefficients of above-mentioned parts, s/m^2.

The hydraulic resistance coefficient for part AB is:

$$S_{AB} = (\zeta_c + \lambda_{ag1}\frac{l_e}{d_{e1}})\frac{1}{128(D_b^2 - d_{b1}^2)^2} \,, \tag{3.41}$$

where ζ_c the pressure loss coefficient of the connector; λ_{ag1} is the Darsy coefficient for the annular channel restricted by diameters D_b and d_{b1}; l_e is the length of the annular channel from the inlet to point B, m; d_{e1} is the equivalent diameter of the annular channel, m. This diameter can be calculated as:

$$d_{e1} = \sqrt{D_b^2 - d_{b1}^2} \,. \tag{3.42}$$

The hydraulic resistance coefficient for the part BC is:

$$S_{BC} = \lambda_{ag2}\frac{l_o}{d_{e2}}\frac{1}{128(D_b^2 - d_{b2}^2)^2} + \frac{\zeta_{oh}}{2gA_h^2} \,, \tag{3.43}$$

where λ_{ag2} is the Darsy coefficient for the annular clearance restricted by diameters D_h and d_{b2}; l_o and d_{e2} are the length and the equivalent diameter of this clearance, m; ζ_{oh} is the pressure loss coefficient for the outlet holes of the drill head; A_h is the total cross-section area of these holes. The hydraulic resistance coefficient for part BD is:

$$S_{BD} = \frac{1}{128[(d_{b1} - \delta_1)^2 - d_{b2}^2]^2 \mu_n}, \qquad (3.44)$$

where δ_1 is the thickness of the inner tube (Figure 3.29); μ_n is the discharge coefficient of the ejector nozzle.

Parts BC and BD are connected in parallel. The Eqs.(3.39) and (3.40) can be summed by the coolant flow rate:

$$Q_{\Sigma_{BCD}} = \sqrt{\frac{H_{BC} - (z_C - z_B)}{S_{BC}}} + \sqrt{\frac{H_{BD}}{S_{BD}}}, \qquad (3.45)$$

and the hydraulic heads on these parts are $H_{BC} = H_{BD} = H_{\Sigma BCD}$. For horizontal drills $z_C = z_B$ and flow rate is:

$$Q_{\Sigma_{BCD}}^{\ 2} = \frac{H_{\Sigma_{BCD}}}{S_{BC}} + 2H_{\Sigma_{BCD}}\sqrt{\frac{1}{S_{BC}S_{BD}}} + \frac{H_{\Sigma_{BCD}}}{S_{BD}}. \qquad (3.46)$$

It is obvious that the flow rate $Q_{\Sigma BCD}$ is equal to the inlet flow rate Q_Σ. Then:

$$H_{\Sigma_{BCD}} = \left(\frac{1}{\frac{1}{S_{BC}} + 2\sqrt{\frac{1}{S_{BC}S_{BD}}} + \frac{1}{S_{BD}}} \right) Q_{\Sigma}^2 . \tag{3.47}$$

The parts AB and Σ_{BCD} connected in series. Letting $z_A = z_B$ (horizontal drill location), sum Equations.(3.38) and (3.47) by the hydraulic heads:

$$H_{\Sigma} = \left(\frac{1}{\frac{1}{S_{BC}} + 2\sqrt{\frac{1}{S_{BC}S_{BD}}} + \frac{1}{S_{BD}}} + S_{AB} \right) Q_{\Sigma}^2 . \tag{3.48}$$

Equation (3.48) shows the relationship between the inlet hydraulic head (H_Σ) and flow rate (Q_Σ). Equation (3.47) defines the hydraulic head at the front end of the ejector nozzle.

The pressure loss coefficients ζ_o and ζ_{oh} are determined experimentally by the using a special method [Astakhov et al., 1983] and a special experimental setup [Astakhov and Ajrikjan, 1983] (Figures.3.34 and 3.35).

The equations for the coefficient of hydraulic resistance for parts AB and BC (Equations.(3.41) and (3.43)) are written for the general case of ejector nozzle location

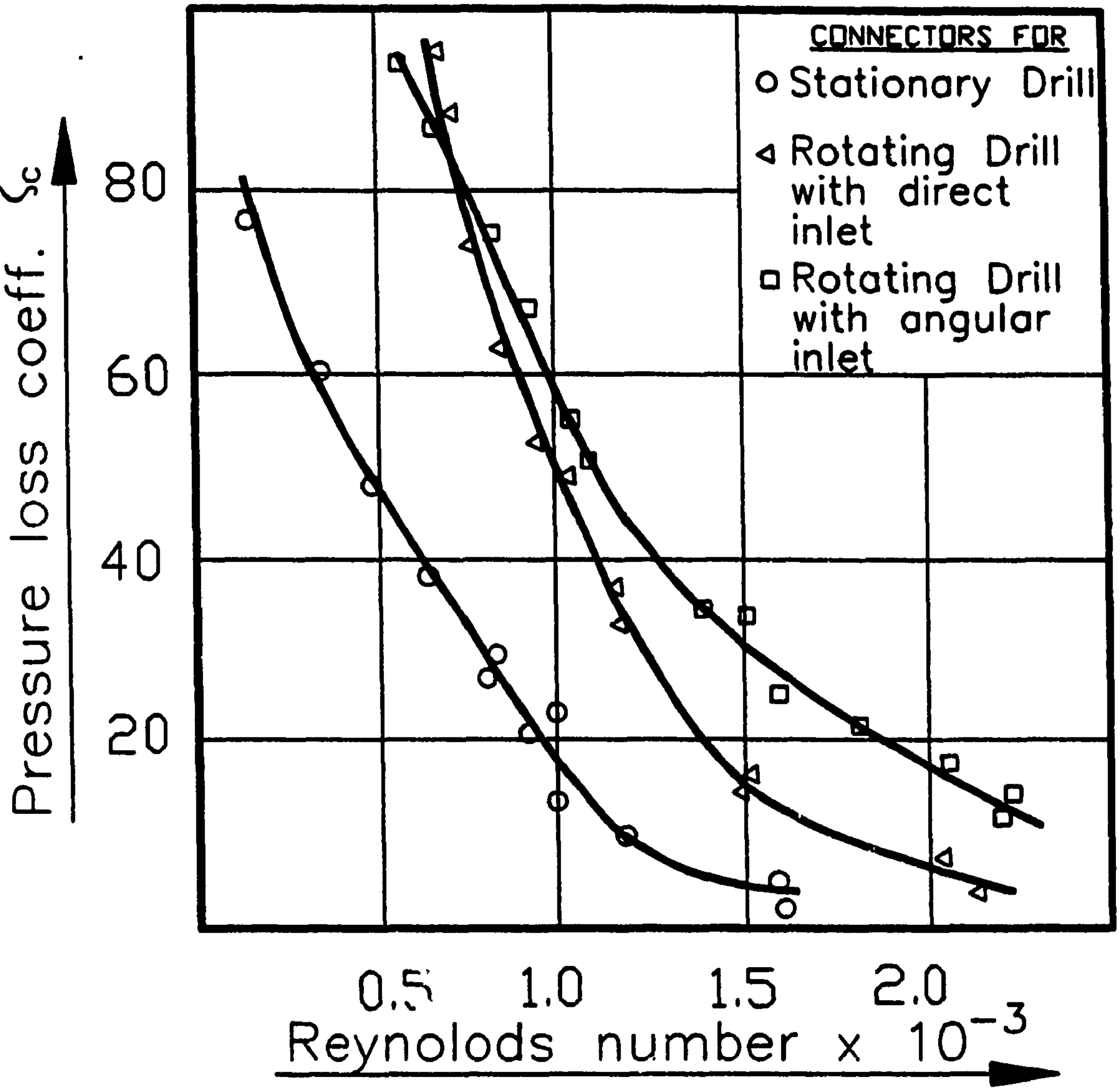


Figure 3.34: Pressure loss coefficient ζ_c Vs Re.

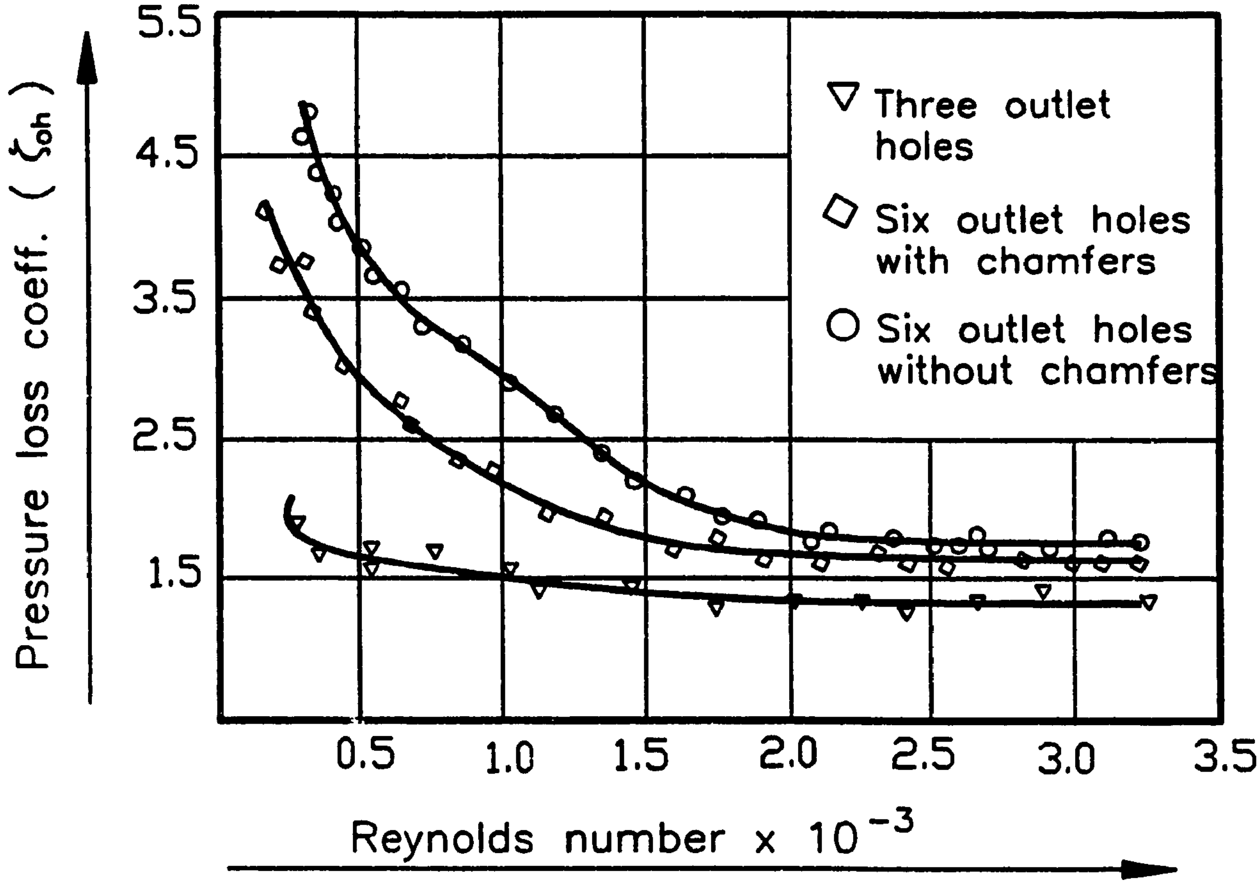


Figure 3.35: Pressure loss coefficient ζ_{oh} Vs Re.

between the connector and the drill head. Currently there are two basic particular cases of the nozzle location available in practice:

1. The nozzle located in the connector. Then:

$$S_{AB} = \zeta_c \frac{1}{128(D_b^2 - d_b^2)^2} , \tag{3.49}$$

$$S_{BC} = \lambda_{ag} \frac{l_1}{d_e} \frac{1}{128(D_b^2 - d_b^2)^2} + \frac{\zeta_{oh}}{2gA_h^2} , \tag{3.50}$$

where d_b is the external diameter of the inner tube; λ_{ag} is the Darsy coefficient for the annular clearance restricted by diameters D_b and d_b; l_1 is the distance between the connector and the outlet holes of the drill head.

2. The nozzle located in the drill head.

$$S_{AB} = \left(\lambda_{ag} \frac{l_1}{d_e} + \zeta_c\right) \frac{1}{128(D_b^2 - d_b^2)^2} ; \tag{3.51}$$

$$S_{BC} = \frac{\zeta_{oh}}{2gA_h^2} \, . \tag{3.52}$$

The calculations of the coolant flow and pressure distributions in the annular clearance for the ejector nozzle can be conveniently done by using the graph-analytical method. Taking the certain values of the flow rate and using equations (3.38) - (3.40), draw curves 1, 2, and 3 corresponding to these equations (Figure.3.36). By the analogy with equations, graphical summation of curve 2 and 3 gives curve Σ_{2+3}, corresponding to equation(3.47). Curve 1 and $\Sigma 2+3$ gives curve Σ_{1+2+3}, corresponding to equation(3.48). Then lay off a length along the Q-axis corresponding to the optimal flow rate, Q_{opt}, (for reliable chip removing) (Astakhov, 1984) through the machining zone. Using curve 2, locate the hydraulic head $H_{\Sigma BCD}$ at point B. This value has significant importance as it defines the working regime of the ejector (ejector working head H_w) (Astakhov and Scorupco, 1982). From $H_{\Sigma BCD}$ and curve 3, the coolant flow rate through the ejector nozzle can be defined (Q_w). Further, using curve Σ_{2+3}, the total coolant flow rate (Q_Σ) can be defined. Using the flow rate Q_Σ and curve Σ_{1+2+3}, the inlet hydraulic head (H_Σ) can be obtained. Thus, the necessary inlet pressure can be calculated as:

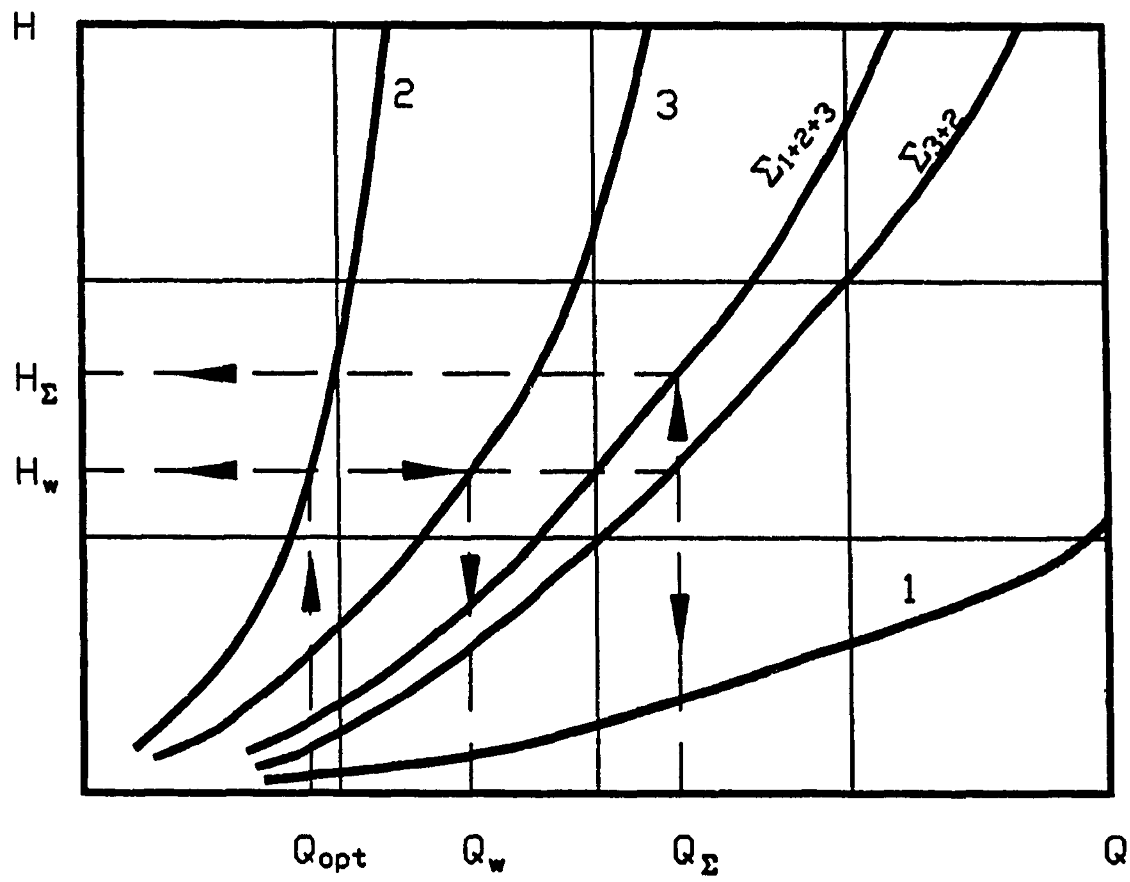


Figure 3.36: Graph-analytical method diagram.

$$P_\Sigma = \frac{H_\Sigma}{\gamma_c}, \tag{3.53}$$

where γ_c is specific gravity of the coolant. Parameters Q_Σ and P_Σ are used for the design or selection of the coolant supply system.

The use of the graph-analytical method for calculating the pressure and flow rate distributions in the ejector drill (20 mm dia) is shown in Fig.3.37. The drill has the annular channel $D_b \times d_b$ = 14×12 mm and length l_1 = 0.6 m. As it can be seen, for calculated optimal flow rate the inlet hydraulic head H_Σ = 260 m ($P_\Sigma \approx$ 2.8 MPa) (curves 1, 2, and 3). After increasing in clearance of the annular channel to $D_b \times d_b'$ = 14×11.6 mm (curves 1, 2, and 3), the inlet hydraulic head becomes smaller and equal to 100 m ($P_\Sigma \approx$ 1MPa) and the relative flow rate for the ejector nozzle becomes close to the optimal ($q = Q_{opt}/Q_w$ = 11/40) [Astakhov and Scorupco, 1982]. A detailed theoritical study of Ejector Design for efficient coolant supply is included in the appendix 3.

3.6 PARAMETER SELECTION BY HEURISTIC OPTIMIZATION:

Initial parameters are the machining parameters which are set before the machining process is started. This is done by heuristic optimization of available knowledge

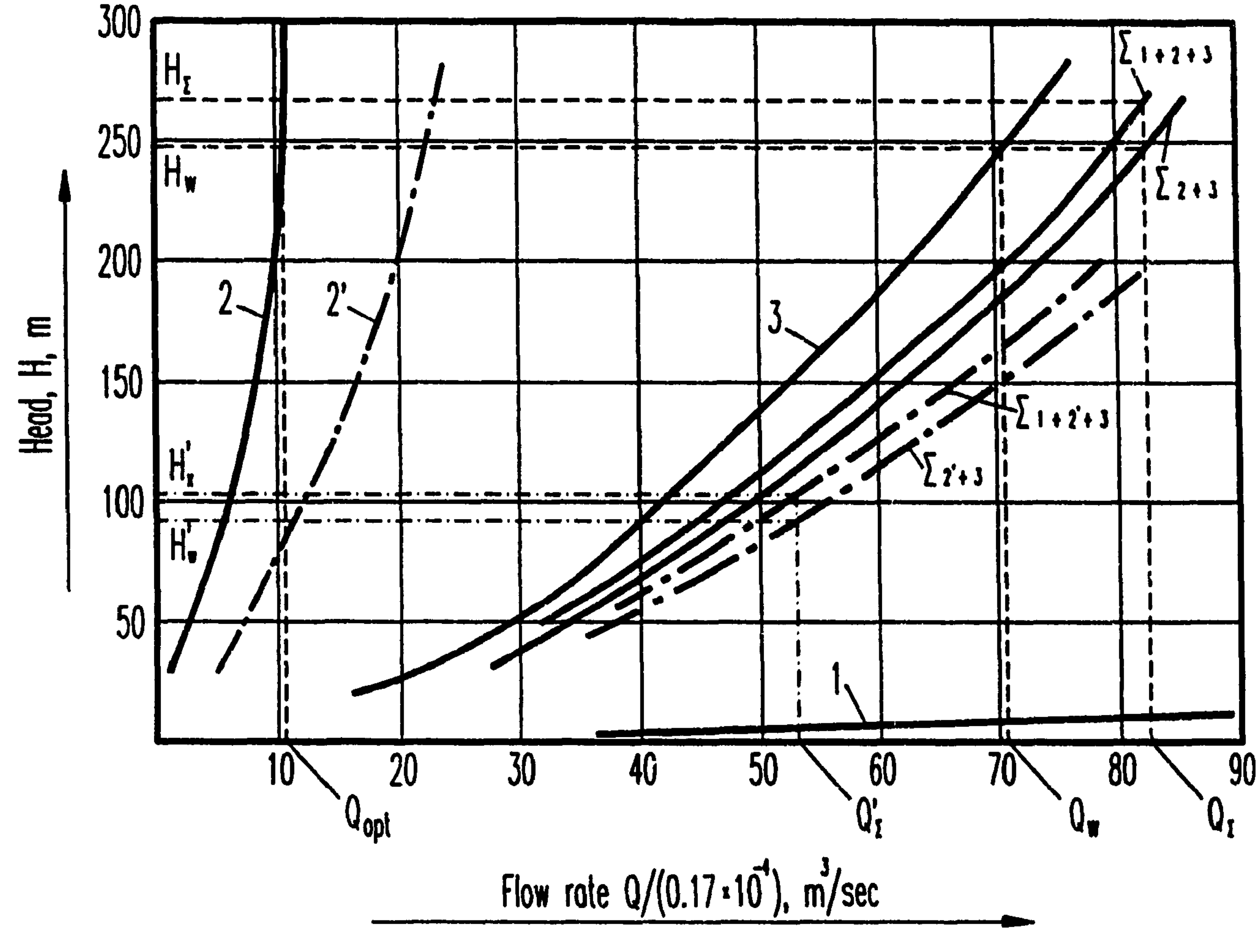


Figure 3.37: Example of the graph-analytical method.

regarding various parametes and their consequences. Heuristic optimization is a method of optimization wherein all the relations between various parameters and their consequences are not available in any comprehencible mathematical form. The goal of heuristic optimization is to obtain a set of values of any parameter which can satisfy or nearly satisfy all or most of the known relations of that parameter to the machining requirements. For example it is known that an increase of feed rate reduces the tool life of the deep hole drilling tool but no mathematical or statistical relation has been established to prove the same because of the extreme expenses involved in carrying out such a study. Thus the goal of Heuristic optimization is to keep the feed rate as low as possible but at the same time maintaining a higher cutting speed to compensate the slow down of the process because of the lower selected feed rate. Eventhough most of the times, the same can be done by using available optimization methods [Subramanya, 1989], heuristic optimization utilizes the knowledge represented in the form of production rules. This is a disadvantage in the conventional optimization method. Figure 3.38 illustrates the algorithm.

3.7. CONCLUSIONS

Torque and thrust in DHMS are an important indicators of the machining and tool conditions. An abrupt increase in the amplitude of torque and thrust values followed by a sudden dorp in their mean value is an indication of tool failure. This knowledge

```
    Get Hole_Oversize and Hole_Surface_finish_Ra
    Search data_base for FRange_Size(min, max)
    Search data_base for FRange_finish(min,max)
    Find_Common_Range(Size, Finish)
    Find_max_Range(Size, Finish)
    Return ( Feed_rate(f))
10  Search data_base for Feed_Dia(Torque)
    Return (Torque)
    Search data_base for Torque(Cut_Speed)
    Return (Tq(Cut_Speed))
    Search data_base for Thrust(Cut_Speed)
    Return (Th(Cut_Speed))
100 Search data_base for Torque(Feed_rate)
    Search data_base for Thrust (Feed_rate)
    Return (Tq(Feed_rate))
    Return (Th(Feed_rate))
    If Tq(Feed_rate)<>Th(Feed_rate)) then
    Feed_rate(f) = Feed_rate(f)-0.05
    Goto 100
    If Tq(Cut_Speed)<Th(Cut_Speed)) then
    Cut_Speed(v) = Th(Cut_Speed))
    Else Cut_Speed(v) = Tq(Cut_Speed))
    Calculate Coolant Flow(Cut_Speed)
    Search data_base for Thrust_Range(min,max)
    Search date_base for Torque_range (min, max)
    Return (Cut_speed(v), Feed_rate(f), Flow_rate)
    Return Thrust(min, mean, max)
    Return Torque (min, mean, max)
```

Figure 3.38: Algorithm for Parameter Selection using Heuristic Optimization

is very useful for controling the DHMS. Levels of torque and thrust can be varied by varying cutting speed and feed rate. For a given cutting speed and feed rate a stable machining region can be defined by upper and lower limts of torque and thrust values. When the torque and thrust values are within this limit, cutting parameters can be appropriately varied by the trend control block to achieve the required machining specification. Heuristic optimization provides an important tool for selecting appropriate cutting parameters to achieve the desired machining specifications of quality and economy.

The coolant is a very important component of deep-hole machining. Various conditions that exist during the operation as well as the design of the drill hydraulic system have a great bearing on the efficiency of the drilling operation. The investigations thus far conducted have shown that:

1 . The coolant pressure in the inlet annular channel varies along the length of the channel linearly for both concentric and eccentric walls. In case of eccentric walls, maximum flow rate is achieved, for a given inlet pressure, when the eccentricity is maximum.

2. In the annular inlet coolant channels with rotating wall(s) the loss of stability in coolant supply conditions was found under certain combinations of the

rotating wall velocity and the axial flow velocity. It is explained by the presence of the Taylor vortices that cause instability in the coolant flow. The method for analytical prediction of the instability flow regimes is established.

3. The presence of the Taylor vortices in the inlet annular channels causes increased heat exchange between incoming coolant and out going swarf. The rate of the increase depends on the flow mode of the incoming coolant.

4. The hydraulic resistance of the inlet annular channels with rotation of inner wall (boring bar) increases with the increase in wall angular velocity and decrease with the increase in the axial flow velocity.

5. The pressure and flow rate distributions in the ejector drills depend upon the ejector nozzle location along the inner tube, design parameters of the nozzle, and the clearance of the annular channel between the boring bar and inner tube. The optimization of the flow rate distribution and the determination of the necessary inlet pressure can be done using the proposed graph-analytical method.

6. The efficiency of cooling, lubrication and chip removal depends on the rate of flow of the coolant as well as that of swarf. The performance of coolant is also

determined by the boring bar/workpiece velocity or the cutting speed. This relation is important as it has to be incorporated in the knowledge base for expert control of DHMS.

7. There exists an optimum coolant flow rate for maximum tool life. Higher than this flow results in decrease of tool life. Thus, for economical and efficient machining the flow has to be maintained within a given range of Q_{min} to Q_{max}.

CHAPTER 4

IMPLEMENTATION OF EXPERT CONTROL

4.1 INTRODUCTION

Implementation of expert controller includes, sensing, decision making and initiating corrective actions. The philosophy and methodology of decision making by expert controller and identification of various parameters required to come to a decision have been dealt in detail in the previous chapters. With an established decision criterion, methodology and decision variables, the knowledge engineer has to implement the expert controller to a deep hole machining system. The implementation requires selection of various sensors namely, force sensors, proximity sensors, pressure sensors etc. Once the sensors are selected, these sensors have to be calibrated to reflect the reality of the process and the signals from the sensors have to be amplified and conditioned before presenting to the data acquisition processor. Selection and formatting of data acquisition process is an important function of the knowledge engineer. Finally the knowledge engineer has to carry out suitable retrofitting of the deep hole drilling machine with variable speed and feed motors which can be controlled by the expert controller via, control consoles. Rewiring the control console to take voltage inputs and calibration of the controller to follow the output of data acquisition processor should also be emphasized. In this chapter a detailed account of various strategies and methodologies adopted for implementing the expert controller is presented.

4.2 REQUIREMENTS FOR IMPLEMENTATION

Core of the expert control system for DHMS is the real time expert system which initiates action based on the input to the system. In order to make a decision, the required input parameters has to be supplied to the expert system by means of data acquisition system and sensors. The decision thus made by the expert system has to be transmitted to the actuators in an appropriate form thereby enabling the implementation of expert control of DHMS. Thus the two main components of the expert control system, other than the expert system are:

a) Sensing System, and

b) Actuating System.

Sensing system has to incorporate the sensing of sequence of operation and process sensing. Sequence of operation is sensed by proper location of proximity switches in the DHMS. Five necessary and sufficient parameters were identified as process descriptors. These five parameters can be broadly classified into two groups:

i. Force parameters, and

ii. Chip factor.

Force parameters, namely, axial force, torque and their respective amplitudes can be measured by means of standard force transducers For measuring the chip factor a special sensor has to be designed and implemented.

4.3 MEASUREMENT OF FORCES

The magnitude of the cutting forces may be estimated (i) by indirect power measurements, (ii) by calorimetric methods and (iii) by suitable development of tool force dynamometers. But, for research purposes, direct measurements by dynamometers have won a general acceptance and since then, many researchers in the field of metal cutting have tackled the problem of constructing dynamometers to measure accurately the force signals using different types of transducers. The transducers normally employed are, piezo-electric transducers, potentiometric transducers, low voltage differential transformers, capacitive transducers, strain gauge transducers, photocells, and permanent magnets. The measurement of forces in machining corresponds to the analytical problem of determining all the components of a general force and moment vector in a chosen coordinate system. As it is impossible to measure the cutting forces at their actual point of generation, the reactions have to be measured in an appropriately defined plane distance from the cutting edge. Therefore, a satisfactory measuring system is essential through which one can obtain the characteristic values, independent of the point of application of the force and/or the moments.

In order to obtain such precise measurements, the measuring system should satisfy the following requirements [Gautschi, 1971]:

a) the original rigidities of the machining system should be maintained as far as possible;
b) the dynamic characteristics should be similar to the original set up;
c) the mounting of the tool or workpiece should remain as close as possible to those of the original set up;
d) the frequency response should be as large as possible;
e) any cross talk or interference between the components of the force and moments should be a minimum;
f) sensitivities of the transducers should be unaffected by time; temperature and the location of mounting of tool and workpiece;
g) the capacity to measure small variations of force and/or moments accurately;
h) the adjustments of zero points, sensitivities, and measuring ranges should be non-critical.

4.3.1 FORCE TRANSDUCER

Piezo-electric transducers are well suited to measure the dynamic process as they fulfil all the requirements specified for a force measuring system in DHMS. Moreover, the main component of a piezo-electric transducers, namely, Quarts crystals, are sensitive to shear. This advantage can be exploited for multi component measurements.

A 2-component load waster, Kistler Type 9065, for measuring axial force (FO, FF) and torque (MT, MF) is selected. This load washer is used in the 2-component dynamometer, Kistler Type 9271A. Based on the standard mounting as specified by the supplier (Kistler), the load washers were pre-loaded to 120KN. At this pre-load, the range for force is -20 to +20KN and the range for torque is -200 to +200Nm. Technical details of this transducer is illustrated in Table 4.1.

These measuring systems consist of three basic parts :

i) A detector transducer stage for detecting the physical variable. Here, in the present case, it is the force due to machining. At this stage the transducer converts the magnitudes of the cutting force into an electric signal.

ii) An intermediate stage for conditioning the signal by magnification, filtration and stabilization.

iii) A final data acquisition stage for processing these signals.

PROPERTY	GENERALIZED FORCE	UNITS	NUMERICAL VALUE
Measuring Range	Axial Force	KN	±20
	Torque	N-M	±200
Resolution	Axial Force	N	0.02
	Torque	N-cm	0.02
Sensitivity	Axial Force	pc/N	-1.8
	Torque	pc/N-cm	-1.6
Overload	Axial Force	KN	144
	Shear Force	KN	±12
	Bending Moment	N-M	±200
Cross Talk	Axial force ↔Torque	N-cm/N	≤±0.02
	Torque↔Axial Force	N/N-cm	≤±0.01
Stiffness	Axial Direction	KN-m	≈6.5
	Torsional Direction	N-m/m-rad	≈500
Resonant Frequency		KHz	≈40
Weight		gm	150
Temperature Error	Axial Force	N/°C	+30
	Torque	N-cm/°C	-5

Table 4.1: Technical details of the load washer.

A schematic arrangement of such a measuring device is shown in Figure. 4.1.

4.3.2 THE FORCE MEASUREMENT SET UP

The set up for measuring both the static and dynamic torque and axial force in deep-hole machining consists of the following sub-systems:

a) The deep-hole drilling machine.

b) The dynamometer capable of holding the two component load washer.

c) Two charge amplifiers.

d) An XYZ compensator.

e) Data acquisition & Storage system

f) Software for signal to symbol transformation.

g) User interface for viewing the data.

The deep hole drilling machine is equipped for rotation of the tool. The stationary workpiece was mounted to the dynamometer-chuck assembly on the spindle. The charge amplifier was used for converting the electrical charge from the piezo-electric transducer of the dynamometer into a proportional voltage. The function of the compensator is to cancel out the cross coupling of torque and thrust signals. The two operational amplifiers were fed with the two measured signals from the charge amplifiers. Any desired portion of the output voltage of any one channel may be

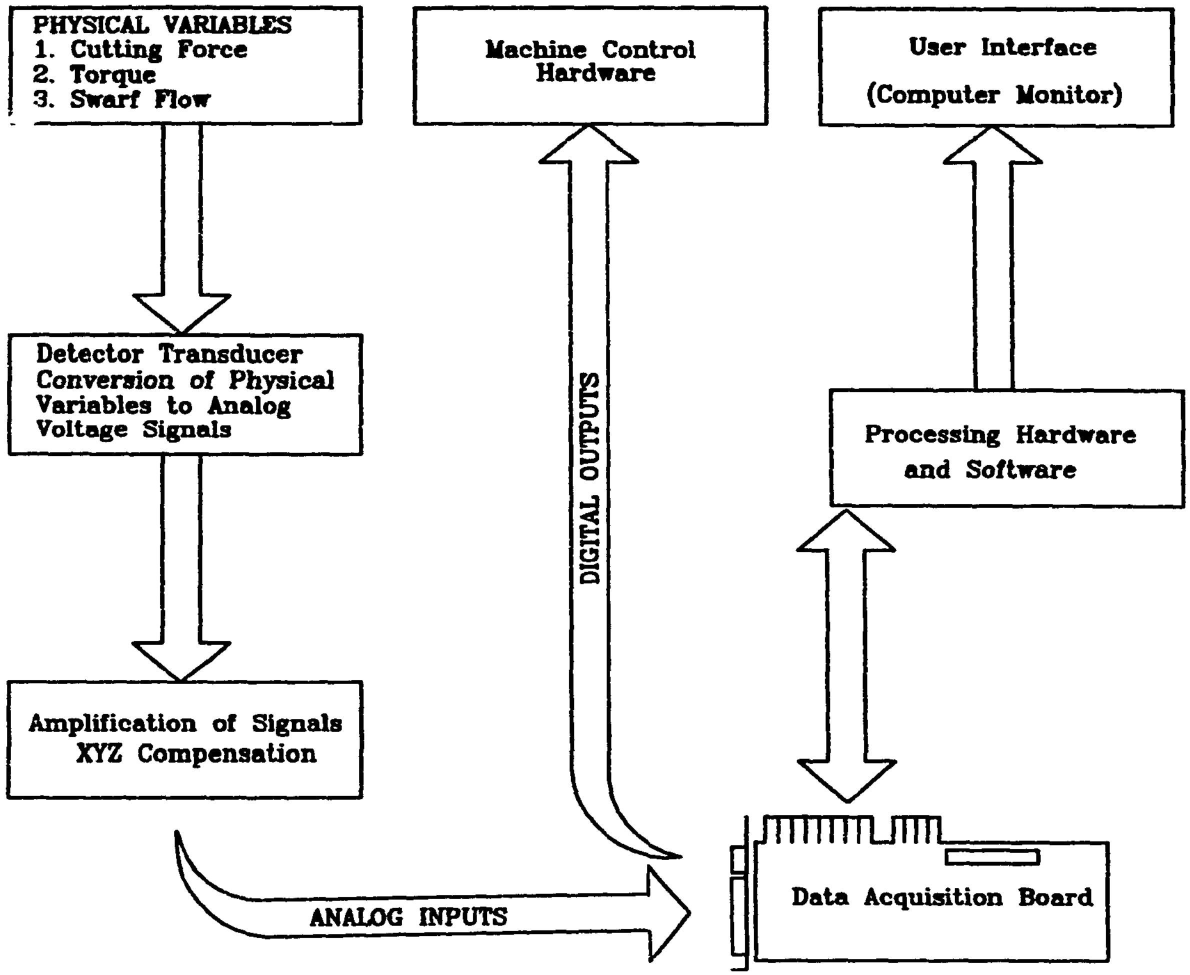


Figure 4.1: Schematic arrangement for process data acquisition.

tapped through a voltage divider and fed as a compensation signal to the input of the other channels. When this is done for both the components, the outputs from the compensator were the actual measured values of the forces. Figure 4.2 shows the functional schematic diagram of the compensator.

There is a real possibility of measuring the vibration of the tool instead of the force fluctuations, to avoid this, A dynamic test of the dynamometer-workpiece-machine tool system is carried out to determine the useful frequency band, within which the dynamometer can be used for reliable measurements.

4.3.3 STATIC CALIBRATION OF THE DYNAMOMETER

The objective of static calibration is to establish a relation between the measured value of the dynamometer and the actual value of the forces to be measured. In spite of the fact that the piezoelectric transducers are generally used to sense dynamic quantities, the static calibration is possible because of the high insulation resistance of the load washer, and by the high input impedance provided by the charge amplifier. Thus, for any value of the load, less than the maximum value, in both the axial and torque direction, the time duration of the signal is sufficiently large enough to permit the reading on a digital voltmeters. So, in the present case a static calibration is performed for both the channels of observation

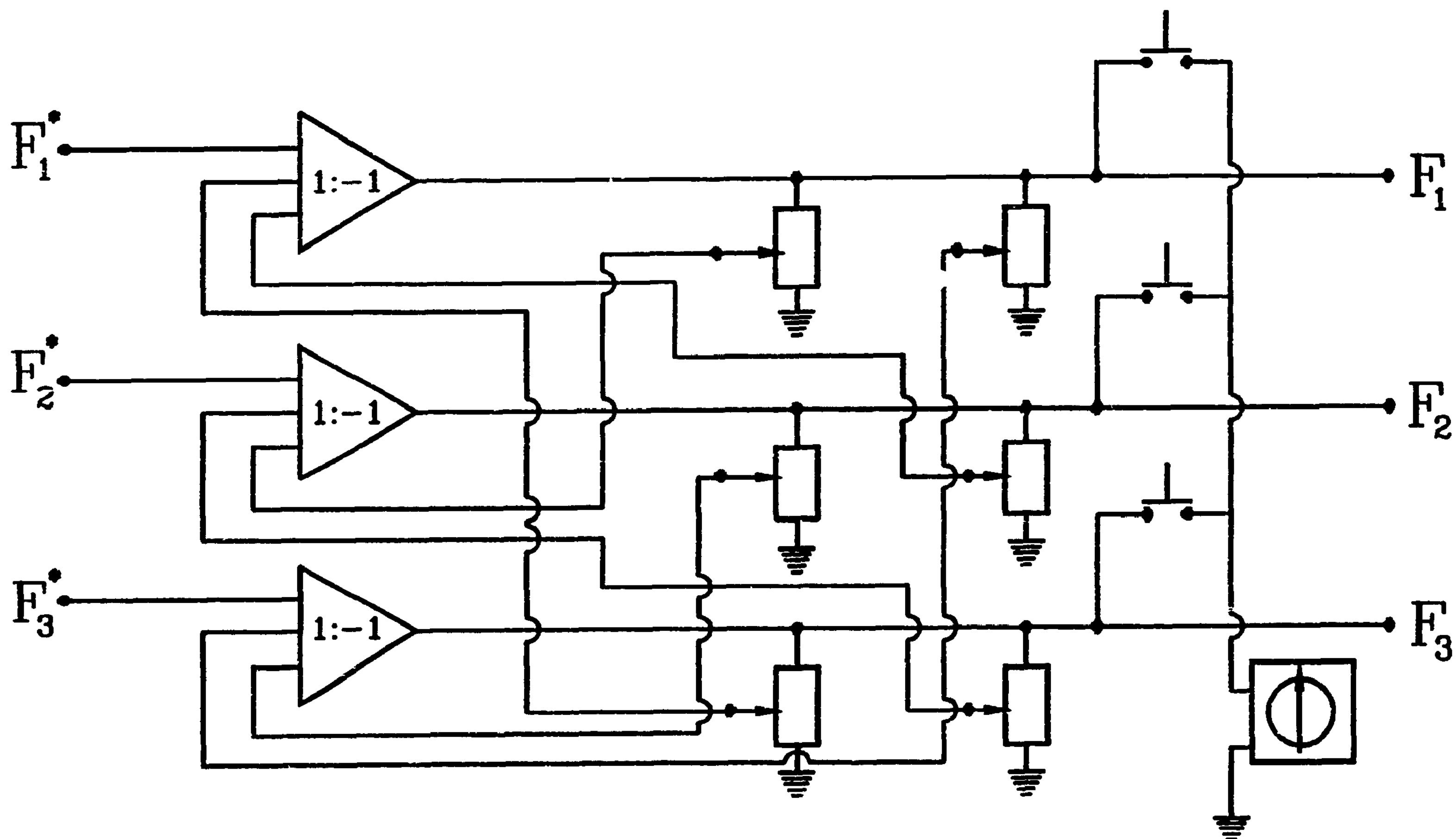


Figure 4.2: Schematic construction of the compensator.

4.3.4 STATIC CALIBRATION ALONG THE AXIAL DIRECTION

The static calibration of the dynamometer is carried out by applying various loads of known magnitudes, measuring the output of the dynamometer, and establishing a graphical relation between the measured quantities and the applied forces.

In the case of the static calibration of the dynamometer along the axial direction the known load was applied using a vertical loading machine. A schematic diagram of the dynamometer and instrumentation used along with it is presented in Figure 4.3.

During the measurement, the charge amplifiers were set to "long", so that the time constant of the system becomes large, and the rate of decay of charge during the static measurement is reduced. This minimizes considerably the error introduced in the static calibration. The load was increased from zero to a value of 13.4 kN and readings were recorded. The output from the torque channel was also recorded to estimate the cross coupling between the channels. Similar readings were recorded during the unloading cycle. The latter is performed to estimate the hysteresis present in the system. It was observed that the difference in the readings obtained during loading and unloading were small, implying minimal hysteresis.

The charge developed in the load washer was calculated using the equation

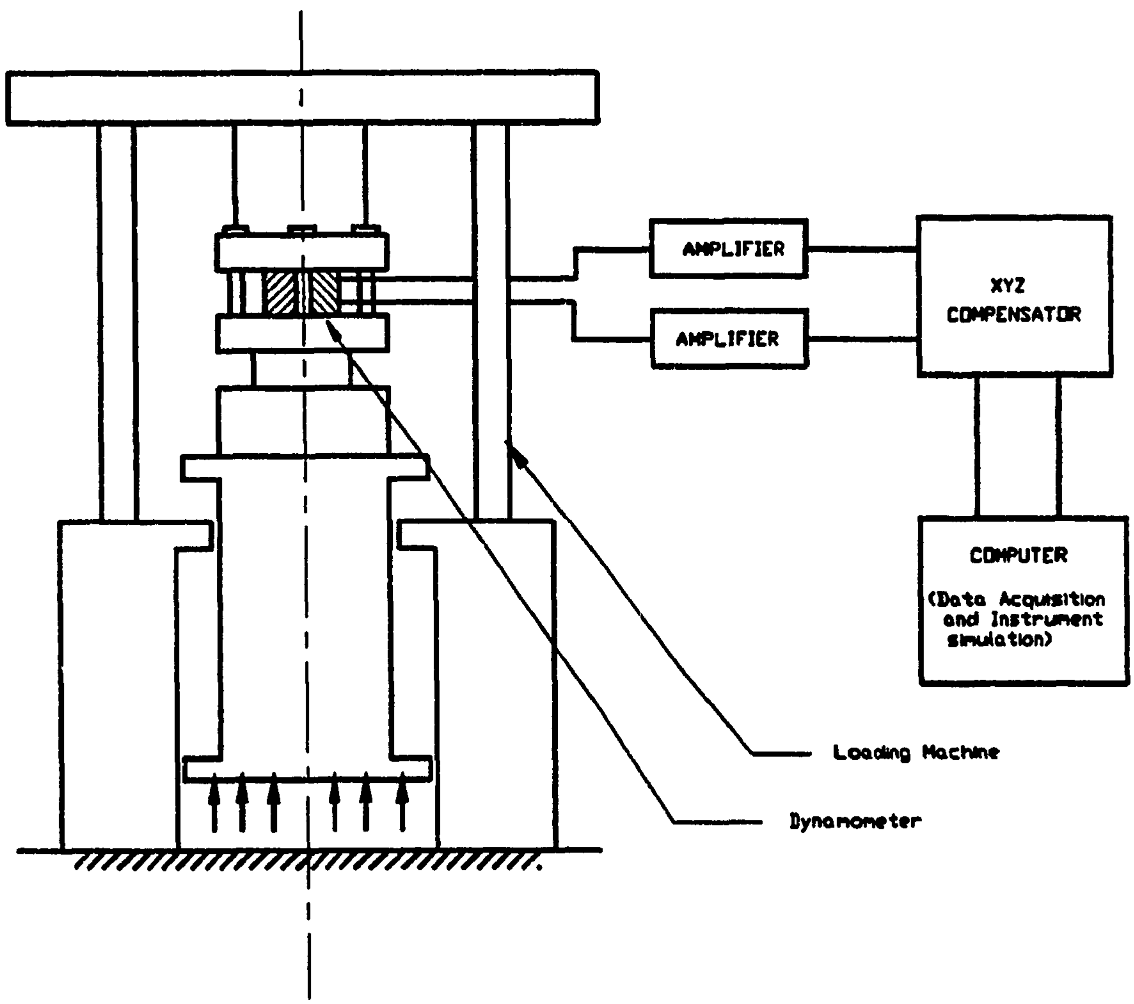


Figure 4.3: Schematic arrangement for static thrust calibration .

$$Q=CV \tag{4.1}$$

where

Q represents the charge developed by the piezo-electric load washer

V denotes the reading of the digital voltmeters

C represents the gain set on the charge amplifier.

4.3.5 STATIC CALIBRATION IN THE TORQUE DIRECTION

The static calibration of the dynamometer along the torque direction is carried out by applying a pure torque of known value. A bar was attached symmetrically on the chuck. Dead weights were introduced at the free ends of this bar. So, the resultant torque introduced equals the moment arm times the load. A schematic diagram of the dynamometer and the instrumentation used is described in Figure 4.4. Similar to the calibration procedure described in the previous sub-section, the charge amplifiers were set to "long" so that the error introduced in static calibration is a minimum. The compensator was adjusted with 80% of the maximum torque, using the nomogram provided [Kistler Ltd.,].

The load was increased gradually from zero to about 470 N and the voltage readings

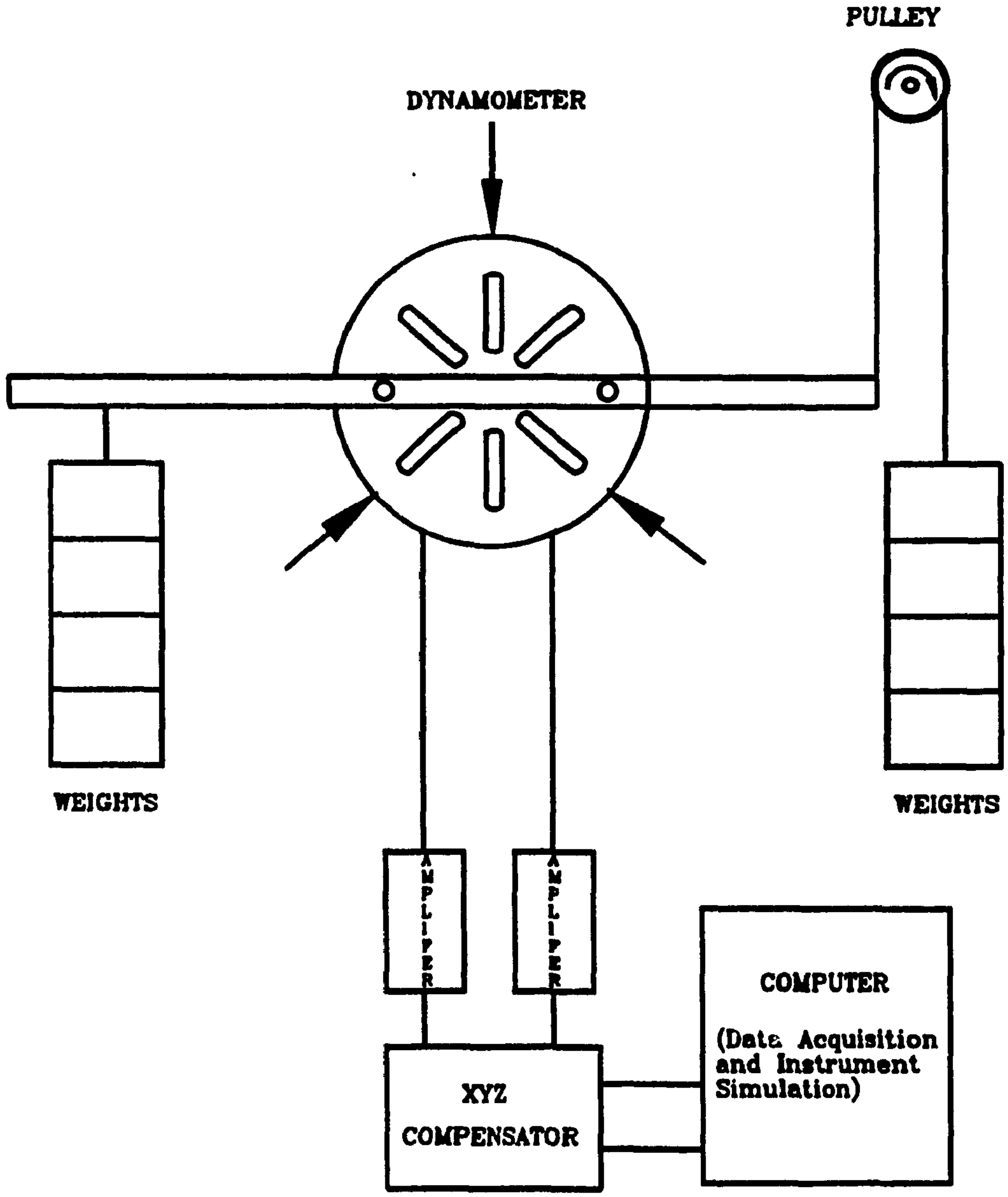


Figure 4.4: Schematic arrangement for static torque calibration .

recorded. The output from the axial force channel is also recorded to estimate the influence of the torque on the axial force. Similar voltage readings were recorded during an unloading sequence. It is observed that the readings taken during loading and unloading were very close indicating minimal hysteresis in the system.

Based on the obtained voltage readings of digital voltmeters a calibration curve for the charge developed by the load washer and the applied torque is established and is given in Figure 4.5. The charge developed is calculated using equation (4.1).

4.3.6 CORRELATION BETWEEN THE MEASURED AND TRUE VALUES

The static calibration of the dynamometer enables the determination of a relation between the charge developed and the magnitude of the axial force and torque. Since, in this case a two component dynamometer is used, the relation between the measured values of the axial force and torque obtained from the calibration chart (Figure 4.5), and the true values of the axial force and torque can be presented as:

$$\begin{bmatrix} F^{*}_{am} \\ T^{*}_{m} \end{bmatrix} = \begin{bmatrix} 1 & K_1 \\ K_2 & 1 \end{bmatrix} \begin{bmatrix} F_{am} \\ T_{m} \end{bmatrix} \tag{4.2}$$

k ,k2 cross correlation coefficients

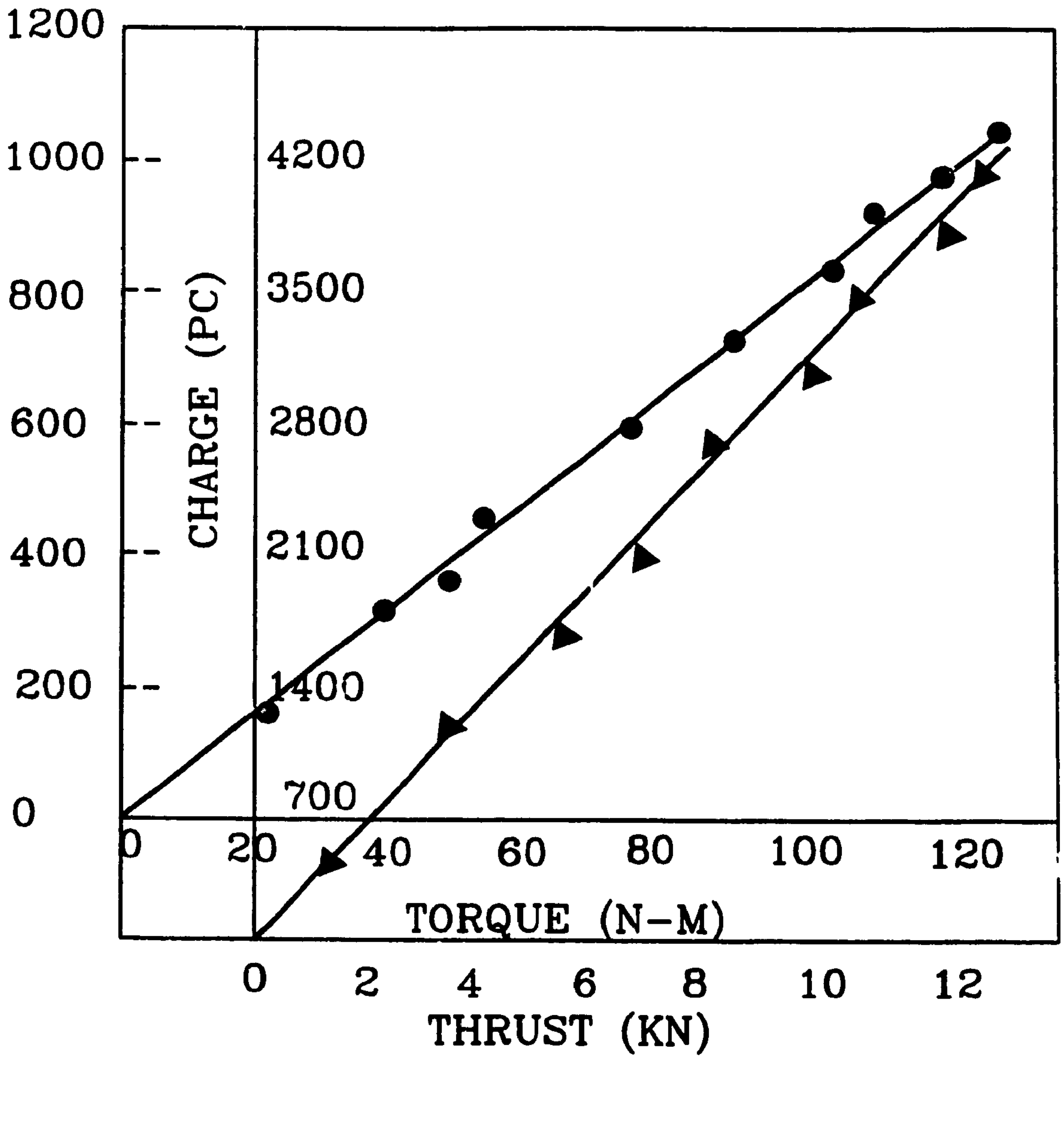


Figure 4.5: Calibration curves for torque and thrust (static).

In this case, $k_1 = k_2 = 0.02$(provided by Kistler). Using equation (4.2) a relation between the true values of the axial force and torque and their respective measured values is obtained as

$$\begin{bmatrix} F_{am} \\ T_m \end{bmatrix} = \begin{bmatrix} 1.004 & -0.02 \\ -0.02 & 1.004 \end{bmatrix} \begin{bmatrix} F^{*}_{am} \\ T^{*}_{m} \end{bmatrix} \qquad (4\ 3)$$

and is employed in further calculations.

4.3.7 DYNAMIC CALIBRATION OF THE DYNAMOMETER

The dynamic calibration of the dynamometer is essential for two reasons:

(i) To determine the resonant frequencies of the dynamometer so as to make sure that the dominant frequencies of the resultant force system are not in the neighbourhood of the dynamometer resonant frequencies.

(ii) To determine the frequency response of the dynamometer-workpiece-machine tool system, so as to determine the range of frequency of the resultant force system which can be measured without signal distortion.

The resonant frequencies of the dynamometer along the axial force and torque directions, were determined using the impact method. A steel ball of about 0.01m diameter is dropped onto the dynamometer along both the axial force and torque directions. The output signals were recorded on an oscilloscope screen. The natural frequencies along the axial and torsional directions were determined to be 4 kHz and 2.8 kHz respectively. The dominant frequencies of the resultant force system were expected to be well within these values of the natural frequencies and hence, test measurements would be valid.

The frequency response of the machine tool-workpiece system is determined in order to find out the frequency band in which the ratio of output to input remains unity. This frequency band determines the range of frequencies of the resultant force system, which can be measured without the introduction of any signal distortion due to the system dynamics or near resonance response.

The experiment to determine the frequency response is first performed with the workpiece fixed to the dynamometer. Next, the workpiece is removed and the load is directly applied to the dynamometer. This was carried out to reflect an actual drilling operation, wherein the cutting tool approaches the dynamometer as the depth of the hole increases. This continuous decrease in distance between the tool and the dynamometer leads to continuous changes in the frequency response of the system. So, to determine the upper and lower bounds of the frequency response characteristics, the experiment was repeated with and without the workpiece. Here, the dynamometer is held in the chuck of the deep-hole drilling machine and the workpiece is clamped onto it. Further, a steady rest supports the workpiece, in order to simulate the effect of the pressure head. The free end of the workpiece is attached to an electro-dynamic shaker. The force with which the shaker excites the system is measured by a piezo-electric crystal introduced between the shaker and the workpiece. A frequency sweep was carried out at a fixed load, and the magnitude of the ratio of the output of the dynamometer to the force applied by the shaker, in frequency range 0 to 2 kHz, was determined. The schematic diagram of the instrumentation used for this experiment is indicated in Figure 4.6. Since forces that were being measured were dynamic, the charge amplifier is set to "short". The results showed that the ratio of the output to input is very close to unity in the frequency range of 0-800 Hz, without the workpiece, and 0-700 Hz in the presence of the workpiece. The cross correlation plots show that the ratio of the output to input has a constant value of 0.02 in the frequency range 0-650 Hz without the workpiece and 0.500 Hz with the workpiece. The decrease in frequency can be attributed to the reduction in mass.

The frequency response of the machine tool-dynamometer-workpiece system when the input force is along the torque direction was carried out using the experimental set up proposed by Chandrashaker[1984]. The experiment was initially performed with the workpiece clamped to the dynamometer and later the experiment was repeated without

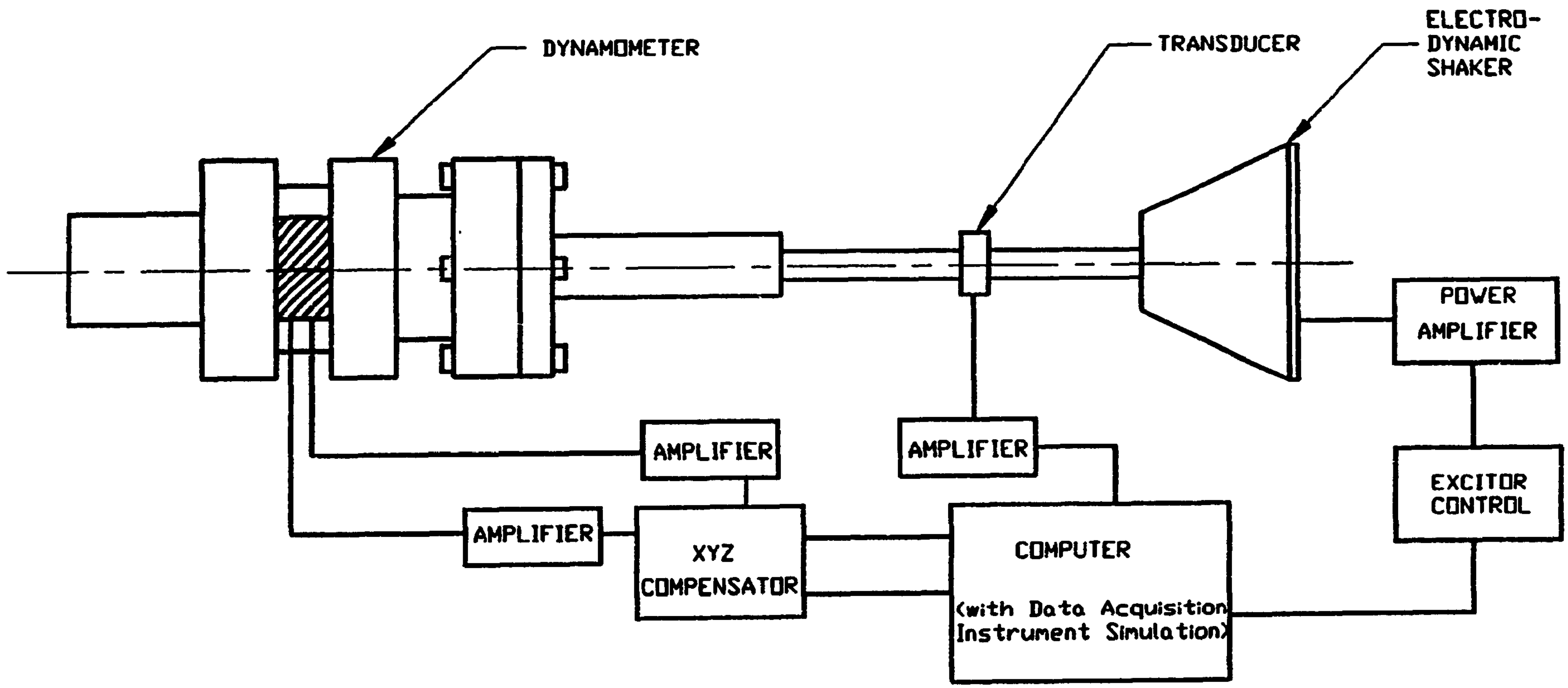


Figure 4.6: Schematic arrangement for dynamic calibration.

the workpiece. The schematic diagram of the instrumentation used for this experiment is similar to Figure 4.6. A similar experiment, to the one described previously, was carried out. The results showed that the ratio of the output to input is unity in the frequency range of 0-350 Hz for both the cases of with and without the workpiece. Similarly the cross-correlation show that the ratio of the output to input has a constant value of 0.02 in the frequency range C-350Hz[Chandrashaker, 1984]. So, it can be concluded that, in the present case, the reduction in mass does not have a significant influence on the useable frequency range.

4.3.8 CORRELATION BETWEEN THE TRUE AND MEASURED SIGNALS

The response of a cutting force dynamometer is often complicated due to fluctuating errors or distortions, which arise when the dynamometer moves during dynamic cutting or due to the dynamic behaviour of the machine tool. Such dynamic errors can be corrected using a suitable data modification programme [Knight et al, 1971], by using a time series technique [Garcia-Gardea et al, 1979], or by obtaining the frequency response by transient loading [Field, 1982]. In this investigation, the dynamic errors were corrected by evaluating the transfer function of the machine tool workpiece system assembly by exciting the system at various frequencies.

As seen from the various transfer function plots, the transfer function value remains a constant until about 350 Hz after which nonlinearities are introduced into the system due to influence of nearness of the system to resonance. Therefore, the reliable measurable range of frequency of the resultant force system is only 0-350 Hz. In this frequency range, the cross-correlation between the axial force and torque is 0.02. Therefore, the following relation can be written between the measured values and the true values of the resultant force system

$$\begin{bmatrix} F^*_a \\ T^* \end{bmatrix} = \begin{bmatrix} 1 & 0.02 \\ 0.02 & 1 \end{bmatrix} \begin{bmatrix} F_a \\ T \end{bmatrix} \tag{4.4}$$

from which the following relation can be established between the true values and the measured values of the resultant force system.

$$\begin{bmatrix} F_a \\ T \end{bmatrix} = \begin{bmatrix} 1.004 & -0.02 \\ -0.02 & 1.004 \end{bmatrix} \begin{bmatrix} F^*_a \\ T^* \end{bmatrix} \tag{4.5}$$

4.4 CHIP SENSOR

To measure the chip factor a new chip sensor is designed. Figure 2.13 shows the schematic diagram of the basic design of the chip sensor. Swarf from the boring bar is directed to diaphragm made of polymide film. A full bridge diaphragm strain gauge(Omega, SG-20/240-DG11) is glued on to the polymide diaphragm. The diaphragm is held under tension using an adaptor (Brass) mounted at the front end of the housing. The entire assembly is positioned at 45° to the boring bar axis to avoid reflecting swarf hindering the impact of incoming swarf. The strain gauge is bridge circuit is energized with 15VAC. The output of the transducer is connected to a single channel strain gage monitoring system (Omega DMD-21). The dynamic strain readings were monitored via analog output of the strain gage monitoring system.

4.5 DATA ACQUISITION

The signals from the sensors has to be communicated to the expert system in the computer for decision making. This is accomplished by data acquisition system. An "AT" class data acquisition system manufactured by Microstar Laboratories Ltd., Washington, DAP 2400, was used for data acquisition. DAP 2400 provides 16 channel Analog and 16 Channel Digital input facilities at a sampling rate of 10MHZ. Six analog output and 16 digital output pins are also provided in the hardware. The inputs to the Data Acquisition Processor are voltages. The Data Acquisition Processor converts voltages to numbers, and then performs computations on the resulting numbers.

4.5.1 ANALOG INPUT VOLTAGES

The input voltage for each analog channel is buffered, amplified by the charge amplifier and the channels gain factor, and fed to the analog-to-digital converter. The gain factor is between 1 and 1000 and is specified independently for each channel. The integers generated by the analog-to-digital converter always are between -32,768 and +32,767. The range of valid input voltages depends on the channel's gain and on the analog-to-digital converter range jumper. Possible ranges with -10 volts to +10 volts.

4.5.2 BINARY REPRESENTATION

Within the Data Acquisition Processor, integers are represented by 16-bit signed binary numbers. These are numbers of the form

xxxx xxxx xxxx xxxx,

Where each "x" represents a "0" or a "1." The highest order (left most) bit always represents the sign. A number is positive if its highest order bit is 0 and negative if its highest order bit is 1. Conversion results should be considered as signed binary fractions. The voltage at the input pin which results in a given conversion result is the fractional reading multiplied by the full scale voltage.

The data acquisition hardware used (DAP 2400) has 12 bit analog-to-digital converters. In order to represent a 12 bit conversion result as a 16 bit number, some of the bits are set to 0. for bipolar ranges, the four right most bits are set to 0 so the conversion results are in the form

xxxx xxxx xxxx 0000

Possible numbers in this form range from -32,768 to 32,752; all are divisible by 16. The DAP 2400 is calibrated so that '0000 0000 0000 0000' represents zero volts in all ranges. There are 4096 possible readings for each scale, so the highest reading is just 4095 steps above the lowest. As a consequence, the highest reading for each scale is slightly below the nominal full scale reading. These binary representation can be formatted to values with decimal points.

The range of the data acquisition is set to bipolar mode, i.e., the range spans on either side of zero (- to +). In bipolar mode, a zero reading from the analog-to-digital converter corresponds to a zero voltage; all other readings depend on the voltage range and the gain factor selected. When the range is -10 volts to +10 volts and the gain is 1, the nominal voltage range spans 20 volts. Each step in the conversion results corresponds to a change of 20/4096 volts, or 4.883 millivolts. A reading of -32,752 corresponds to +9.995 volts. The following table summarizes the integer values generated by the analog-to digital converter with a bipolar voltage range of -10 to +10 volts:

Input Voltage	Digitized value
-10.000	-32768
-9.995	-32752
-9.990	-32736
--------	---------
--------	---------
-0.005	-16
0.000	0
+0.005	+16
-------	--------
-------	--------
+9.985	+32720
+9.990	+32736
+9.995	+32752

4.5.3 DIGITAL INPUT VOLTAGES

Digital inputs are voltage signals with only two significant levels, low (0) and high (1). Data Acquisition Processor inputs follow the standard TTL[1] specification that any voltage between ground and 0.8 volts is low, and any voltage between 2.0 volts and the supply voltage, approximately 5.0 volts, is high. Voltages between 0.8 volts and 2.0 volts are regarded as transition voltages, and are sensed as low (may be sensed as either high or low).

All the digital signals which forms inputs to the data acquisition processor should

[1]Transistor Transistor Logic

originate from TTL compatible components such as '0' and '1' switches. The 16 pins of the digital input port may be interpreted either as forming one 16 bit digital input, or as 16 binary inputs. There are 10K ohm pullup resistors on all digital inputs. Because of these pullup resistors, unused inputs appear as '1's.

4.6 HARDWARE AND SOFTWARE ORGANIZATION

Figure.4.7 illustrates how data pass through the sampling hardware of the Data Acquisition Processor. The Data Acquisition Processor reads analog voltages from 16 single- ended input pins or 8 differential pin pairs or any combination of single-ended and differential inputs. The Data Acquisition Processor reads digital voltages through a 16 bit digital input port. The maximum sampling rate of the DAP 2400 is 10MHz. The analog voltage is channelled from input pins through a multiplexer and an instrumentation amplifier to a programmable gain amplifier. The gain is selected independently for each channel, with possible values of 1, 10, 100, and 1000. A level converter shifts the bipolar voltages to the correct unipolar voltage range. A sample and hold circuit holds the voltage stable for the analog-to-digital converter. The number of inputs can be increased up to 512 analog inputs and 128 digital inputs using external multiplexer.

The input control and timing circuit controls the input circuitry, so that the

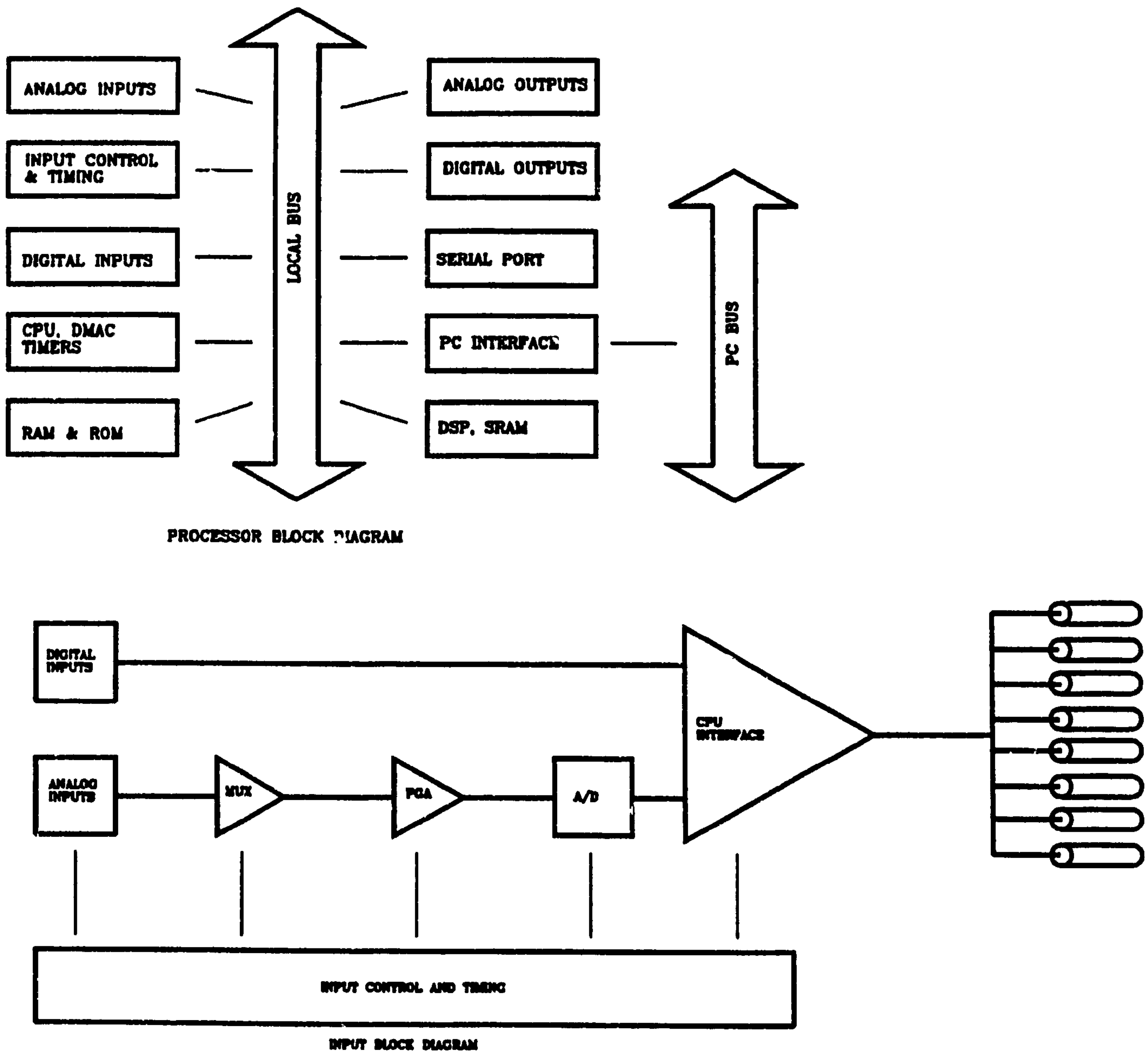


Figure 4.7: Hardware arrangement for data acquisition.

microprocessor is free to process the digital results. This circuit selects input pins and channel gains. The direct memory access controller transfers the digitized analog-to-digital converter output to the on-board memory.

4.6.1 REAL-TIME OPERATING SYSTEM (DAPL)

Data acquisition requires sophisticated software which schedules and performs many tasks concurrently. The software must run in real-time without losing data, and it must be able to adapt to an unpredictable data stream at the analog input pins. The Data Acquisition processor provides this software as the multitasking operating system called DAPL. DAPL is supplied by Microstar Laboratories Ltd., Washington, in ROM to be incorporated on the Data Acquisition Processor.

When an application is running in a data acquisition processor, DAPL controls the analog sampling hardware and schedules DAPL tasks. DAPL tasks process data in the Data Acquisition Processor and transfer data to the PC for further consideration by the expert system. Figure 4.8 illustrates software data flow.

DAPL tasks communicate through buffers called pipes. Tasks place data in pipes, and tasks remove data from pipes. A pipe holds data from one task until another task is ready for the data. DAPL keeps data in the correct order by removing data from pipes

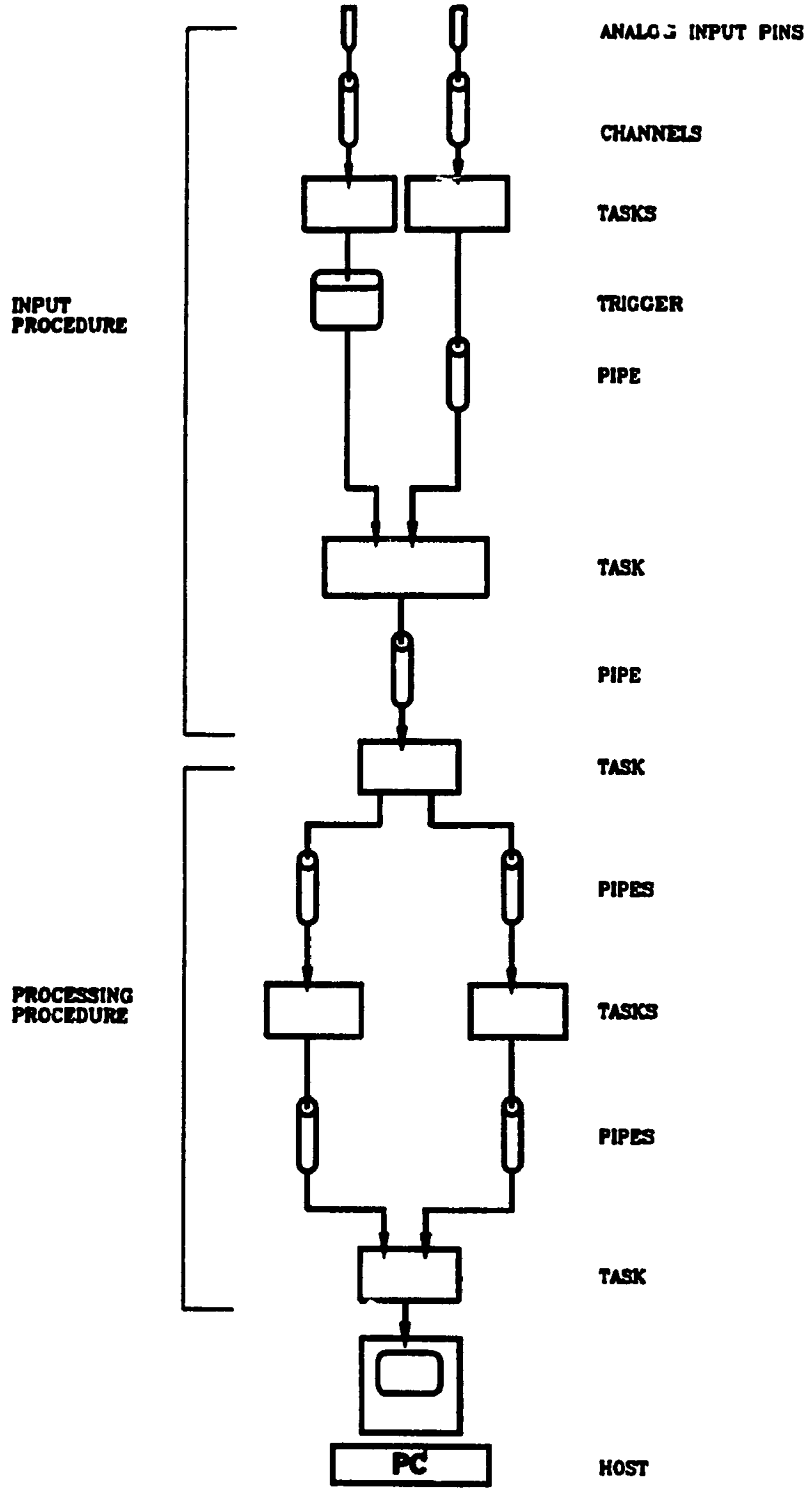


Figure 4.8: Software data flow.

on a first-in-first out basis. DAPL synchronizes tasks and allocates buffer memory. A task must wait if it attempts to read from an empty pipe or to write to a full pipe. If a task exceeds its time allocation or can not proceed for any reason, DAPL goes on the next task. DAPL guarantees that pipes expand and contract automatically. The limits for pipe expansion can be set for each application.

DAPL supports special pipes called triggers which are used for synchronizing tasks. Trigger events are not limited to sensing levels or rates of change. One task asserts a trigger each time a trigger event occurs. Other tasks wait for the trigger, synchronizing to the trigger events. Triggers are used in implementing the flag changing as explained in chapter 2 for maintaining the sequence of operation and also for detecting the tool failure wherein a trigger sets the spindle and feed motor to 'STOP' once the peak amplitude trigger is asserted by a peak in the amplitudes of torque or thrust.

A DAPL task is defined by specifying:

A command,

One or more parameters

One or more triggers on which the task waits

One or more triggers which the task asserts

One or more pipes from which the task reads data

One or more pipes into which the task writes results

4.6.2 THE DAPL COMMAND INTERPRETER

The DAPL command Interpreter is a DAPL task whose function is to parse and execute DAPL commands received by the data acquisition processor from the expert system residing on the host PC. The DAPL command interpreter is always kept active.

DAPL commands fall into following categories: System commands, input configuration commands, input task definition commands, output configuration commands, output task definition commands, and processing task definition commands. System commands define DAPL structures such as variables, pipes and triggers, start and stop sampling, set system options and request status information,. Input configuration commands define the sampling configuration of the data acquisition processor. input task definition commands define tasks which process input channel data. Output configuration commands define the output configuration of the data acquisition processor. Output task definition commands define tasks which prepare output channel data. Processing task definition commands define tasks which process data other than channel data. The DAPL command interpreter processes all commands immediately. Commands, however, may have long lasting effects. For example, the system command

PIPE P

creates a pipe P. After the command is executed, the pipe and its name remain. The pipe name may be used in task definitions, and the pipe may be used to buffer data sent from one task to another.

In order to understand how the acquisition processor works, it is important to distinguish between a task definition command and the task which is defined by that command. When the data acquisition processor receives a task definition command from the expert system on the host PC, the DAPL command interpreter first parses the command to check for errors. If there are no errors, the DAPL command Interpreter sets up a data structure which defines a task. If there are errors, the DAPL command interpreter sends error messages back to the PC. It is important to note that the defining a task does not activate the task. Each task is defined within an input procedure, and output procedure or a processing procedure. A task is inactive until the definition of the procedure is completed and the procedure is started. Only then does the task become active and start processing data.

4.6.3 COMMAND INTERPRETER MODES

Input, output, processing, and system commands each must be entered when the DAPL command interpreter is in the appropriate input mode. System commands are entered when the DAPL command Interpreter is in "System" mode. Input commands are entered when the DAPL command interpreter is in "Input procedure definition" mode. Output commands are entered when the DAPL command interpreter is in "Output procedure definition"mode. Processing commands are entered when the DAPL command interpreter is in "processing procedure definition" mode.

initially, the DAPL command interpreter is in the system mode. Any system command may be entered at this time. Once the system command is entered the command interpreter changes over to procedure definition mode. Termination command exits input procedure definition mode. At this stage a process definition command places the DAPl command interpreter in processing procedure mode. In this mode the signals obtained by various channels of the acquisition system are processed through the expert system. At the end of processing a termination command places the command interpreter in the output procedure definition mode. The output procedure definition includes the TIME and list of channels for various outputs.

4.6.4 DIGITAL FILTERING

The data acquisition processor provides digital filtering algorithms in its firmware. To perform digital filtering, a VECTOR of filter coefficients must be defined. The acquired signals are copied into two pipes. The pipes utilized for monitoring the average forces are filtered. A low pass filter (0 -350Hz) is adopted to filter out the unwanted high frequency noise. Since no frequency analysis is incorporated in the expert controller, the number of taps on the filter is limited to 20 as a compromise on speed.

4.6.5 ANTI-ALIASING

A sampled data system cannot resolve frequencies above half of the sampling frequency. If high frequency components are present, it is necessary to condition the inputs with anti-aliasing filters. The low pas filter adopted in data acquisition procedure removes all components at frequencies above half of the sampling frequency. Thus accomplishing anti-aliasing.

4.6.6 PROGRAM ON THE PC

The expert system for controlling the DHMS is resident on the host PC. This calls for a communication method between the host PC and the data acquisition processor via DAPL command interpreter. This is relatively simple considering the fact that the data acquisition processor is responsible for all buffering and real time control. The expert system uses the data acquisition processor as a DOS device with the name ACCEL. Since DOS treats devices as files, an PC program like expert system can read from and write to ACCEL as if it were a disk file. The expert system opens one binary input file and one binary output file, both with the name ACCEL. The expert system writes commands to the data acquisition processor by writing to the output file and reads data from the data acquisition processor by reading from the input file. The binary communication between the PC and the data acquisition processor eliminates the time consuming format conversion from binary to ASCII. Binary format also requires transmission of fewer bytes from the data acquisition processor to the PC thus reducing the communication overhead.

4.6.7 DATA TRANSFER

If each and every data were to be processed by the expert system, then the shear volume of the data available makes it impossible to process all the data within the expert system and propose appropriate outputs. This leads to an unstable system wherein the output changes with every input. In order to eliminate this only two types of data, the group average(r.m.s) and the peak value are transferred to the expert system from the data acquisition processor. The peak value determines the output trigger which may stop the process in case of a tool failure. The average data are further processed in the trend control block for specifying appropriate output values to the data acquisition processor. The processed data are then discarded to minimize the load on the storage medium and only the rule which was activated is stored to generate the history of the process for further consideration by the expert system should it be required by the next set of data.

4.6.8 ANALOG PERFORMANCE

The programmable gain amplifier is the only analog section of the data acquisition processor which limits high speed data acquisition. The minimum sampling times of the data acquisition system, in microseconds, are 4.25 at gain 1, 8 at gain 10 and 40 at gain 100, and 500 at gain 1000. Sampling faster than these minimums results in data with less than 12 bits of accuracy. For fast sampling at high gains, external signal conditioning is required. In the developed expert controller for DHMS a unit gain is adopted for sampling and hence the sampling time is minimized to 4.25 microseconds.

4.6.9 OVERFLOW AND UNDERFLOW

When sampling occurs faster then data can be processed and transmitted to the host, buffer overflow eventually must occur. Therefore, Averaging and peak detection are the only processes requested by the acquisition system prior to transmission to the host PC. If the sampling is done faster than the time required for Averaging and Peak detection the data acquisition halts gracefully. Sampling steps without loss of data, and the data acquisition processor continues processing the valid acquired data. Even at the highest speed, the Data acquisition processor correctly buffers the acquired data in its on board memory. An overflow of buffer memory never results in any loss or corruption of buffered data. To overcome the overflow the sampling time should be higher than the processing time. Averaging and Peak detection processes are done by the DAPL within 2 microseconds and hence the minimum sampling time of 4.25 microseconds at unit gain does not lead to overflow.

4.6.9.1 PREVENTING THE OVERFLOW

The first step in preventing overflow is to determine where data values are backing up. This can be accomplished by sampling until an overflow occurs and then the displaying the contents of all the pipes defined. A pipe which contains more than 1000 data at the overflow instance may by the cause of overflow. If no pipe contains excessive data, overflow is occurring in the input channels. This means that the input tasks cannot keep up with the sampling rate. Solutions include reducing the sampling rate, using ignore options, using triggers to extract only the relevant data.

4.6.9.2 UNDERFLOW

When output procedure updates occur faster than the data acquisition system can place data into output channels, channel underflow eventually must occur. When underflow occurs, output updating stops. Even though the underflow problem does not exist in the expert controller for DHMS, as the outputs are updated infrequently, it desirable to provide ways in the system design to overcome this problem. This is achieved by either cyclical output procedures when the channel data are repetitive or by increasing the output buffering time.

4.7 RETROFITTING THE DEEP HOLE BORING MACHINE

Final stage of implementing the expert controller is the transformation of expert decision into change of machining parameters. To enable this the actuators on the machine should be able to follow the command issued by the computer.

The spindle motor is replaced with a 10hp variable speed A.C. motor and the feed motor is replaced with a 3hp variable speed A.C. motor. The motors are individually controlled by AC inverters. The AC inverters are designed to provide required volts/hertz ratio, allowing the AC motors to run at their optimum efficiency and providing rated torque capability through the motor's rated base speed. The basic principle relating the frequency and motor speed is given by:

$$N_s = \frac{120*F}{P} \tag{4.6}$$

where Ns = Synchronous (unloaded) motor speed,

F =Frequency (HertZ), P = Number of motor poles

Motor (Loaded) Speed = Synchronous speed - Motor Slip (RPM).

The control section of the AC inverter consists of a control board with a 16-bit microprocessor and keypad interface board with an 8-bit microprocessor. Drive programming can be accomplished via the keypad or the optional serial communications port. The controller is provided with many protection and feed back circuits. The AC inverter has provisions for controlling the motor speed by a process signal between 0 and 10 volts DC. A schematic diagram of the motor control scheme is illustrated in figure 4.9.

4.7.1 MOTOR SPEED AND REFERENCE VOLTAGES

Both the spindle motor and feed motor are 4pole AC motors whose speed can be varied by supplying voltages from 0 to 10 volts to the respective controllers. The input voltages to the controllers are the output voltages of the D/A converters of the data acquisition processor. The D/A converter accepts digital information from the

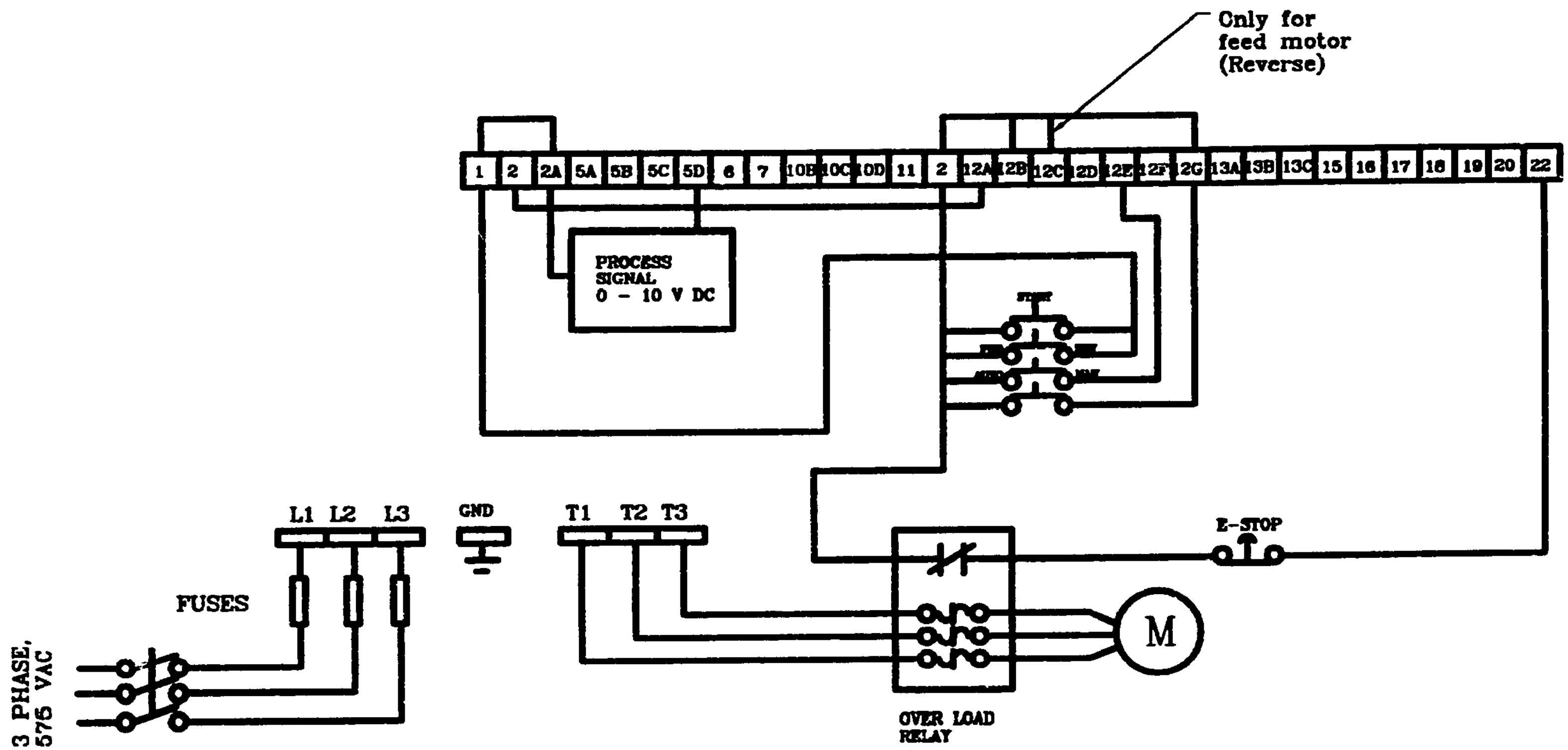


Figure 4.9: Motor control scheme.

expert controller and transforms it into an analog voltage. The digital information is in the form of a binary word. The D/A converter converts a digital word into an analog voltage by scaling the analog output to be zero when all bits are zero and maximum value when all the bits are at one. In this context, the output of D/A converter can be viewed as a scaling of some reference voltage.

$$V_x = V_R[b_1 2^{-1} + b_2 2^{-2} + \dots\dots\dots + b_n 2^{-n}] \quad (4.7)$$

where

V_x = analog voltage output

V_R = reference voltage

$b_1\ b_2 \dots b_n$ = n-bit binary word.

This equation(4.7) shows that the minimum V_x is zero and the maximum depends on the number of bits in the binary word. From the above equation it is also imperative that a 16 bit word will convert a maximum reference of 10 volts into a maximum output voltage of 10 volts without any loss due to binary word representation:

$$V_x = 10(2^{-1} + 2^{-2} + \dots\dots + 2^{-16}] \;=\; 10.00\,Volts \quad (4.8)$$

4.7.2 RESOLUTION OF CONVERSION

It has been shown, from equation 4.8, that a 16 bit word will result in exact conversion of the reference voltage into output voltage. The conversion resolution is also a function of bits in the binary word. The smallest possible change in the output voltage can be given by:

$$\Delta V_x = V_R 2^{-n} \tag{4.9}$$

where ΔV_x represents the smallest increment or decrement in the output voltage of the data acquisition processor. Form the above equation it is evident that the 16bit binary word can produce a resolution of $1.526 * 10^{-4}$ volts in 65540 steps. This implies a change of 0.0275 RPM of the motor. Since this resolution is less than 1RPM of the motor, The motor speed can be varied in steps of one rotation per minute between the speeds of 0 to 1800rpm providing 1800 cutting speeds and feed rates for a given tool diameter. This fine resolution is not required and cannot be sustained for expert control of DHMS because of the limitations imposed by the frequency resolution and the frequency stability of the AC inverter. A complete summary of the AC inverter specifications, limitations etc., are given in Table 4.2.

The expert controller requires the change in motor speeds in three ways as suggested by the expert during knowledge acquisition, they are, GRADUAL, SLOW, and SUDDEN. These three ways are implemented as follows:

SLOW: Output voltage is incremented to required voltage over a time span of 5seconds. Increment step size is maintained constant and voltage step

Table 4.2 : Technical details of speed and feed motor controllers

DESCRIPTION OF CONTROL CIRCUIT TERMINALS		
ANALOG INPUT TERMINALS	2A	Analog circuit common, used as a reference for analog speed inputs and analog speed and load indicating outputs.
	5A	Speed potentiometer wiper input, used in conjunction with terminals 6 and 2A for potentiometer speed control - Manual operation.
	5B	Process current input (4-20mA), 100Ω input impedance, referenced to terminal 2A.
	5C	Process voltage input (0-5VDC), 100Ω input impedance, referenced to terminal 2A.
	5D	Process voltage input (0-10VDC), 200Ω input impedance, referenced to terminal 2A
	6	Speed potentiometer reference supply (5VDC) where 100% (high) side of a speed control pot.(1-25KΩ) is connected. Referenced to terminal 2A.
DIGITAL INPUT TERMINALS	1	Digital stop input terminal, must remain connected to terminal 2 during operation
	2	Digital circuit common, used as a reference for digital command inputs and the digital frequency output. Isolated from earth ground.
	7	Digital keypad select terminal, active during manual control
	12A	Digital start input terminal
	12B	Digital forward direction selection terminal (Always active for spindle motor)
	12C	Digital reverse direction selection terminal(Active for tool withdrawal, feed motor only)
	12D	Digital Jog input, for manual speed variation.
	12E	Speed potentiometer reference selector.
	12F	Process current reference selector
	12G	Process voltage reference selector, 0-5VDC connected to 5C and 2A or for 0-10VDC signal connected to 5D and 2A.
	22	Digital emergency stop (E-STOP) input, Always active.

Table 4.2 : Technical details of speed and feed motor controllers

#	PARAMETER NAME	DATA RANGE	SETTING	UNITS
1	Current limit	5-180	180	%
2	Thermal overload	50-150	150	%
3	Slip compensation	0-0.1	0	%
4	Speed @ 4mA / 0 VDC	0-360.0	0	Hz
5	Speed @ 20mA / 10 VDC	0-360 0	120	Hz
6	Jog speed	0.5-62	10	Hz
7	Normal Acceleration Ramp	0.1-999.9	01	Seconds
8	Jog Acceleration Ramp	0.1-999.9	30	Seconds
9	Normal Deceleration Ramp	0.1-999.9	30	Seconds
10	Jog Deceleration Ramp	0 1-999 9	30	Seconds
11	Frequency Out @ Maximum	1-360	120	Hz
12	Load Out @ Maximum	10-200	175	%
13	Minimum Frequency	0.5-120.0	0.5	Hz
14	Maximum Frequency	0.5-120.0	120.0	Hz
15	IR Compensation	0-20	0	%
16	DC Brake	Off, Timed, Cont.	Timed	----
17	DC Brake Load	1-180	50	%
18	DC Brake Time	0.1-300	0 1	Seconds
19	Speed unit	Hz,RPM, %RPM	Hz	-----
20	Load units	AMPS, %LOAD	%LOAD	-------

size is calculated by the expert system.

GRADUAL: Output voltage is incremented to required voltage over a time span of 2 seconds. Four increment steps are adopted and the voltage step is determined by the expert system using the current voltage output and the desired voltage outputs.

SUDDEN: Output voltage is incremented to required voltage in one step over 0.1 seconds which is the normal acceleration time required by the motors.

Figure 4.10 illustrates the complete setup of the expert control system for DHMS. Figure 4.11 is the schematic representation of the entire setup for expert control scheme.

4.8 CONCLUSIONS

Implementation of the expert control requires accurate measurement of process signals and full transfer of reference voltages from the computer to the controller. The process signals are transformed to symbols and passed on to the expert system as the rule premises, thus emphasizing the importance of accurate measurement of process signals. To ensure the accurate measurement of process signals, proper calibration of the sensors is necessary. The force sensor used for measuring the torque and thrust is calibrated for both static and dynamic measurements. A low pas filter is adopted

Figure 4.10 : Expert control set up for DHMS.

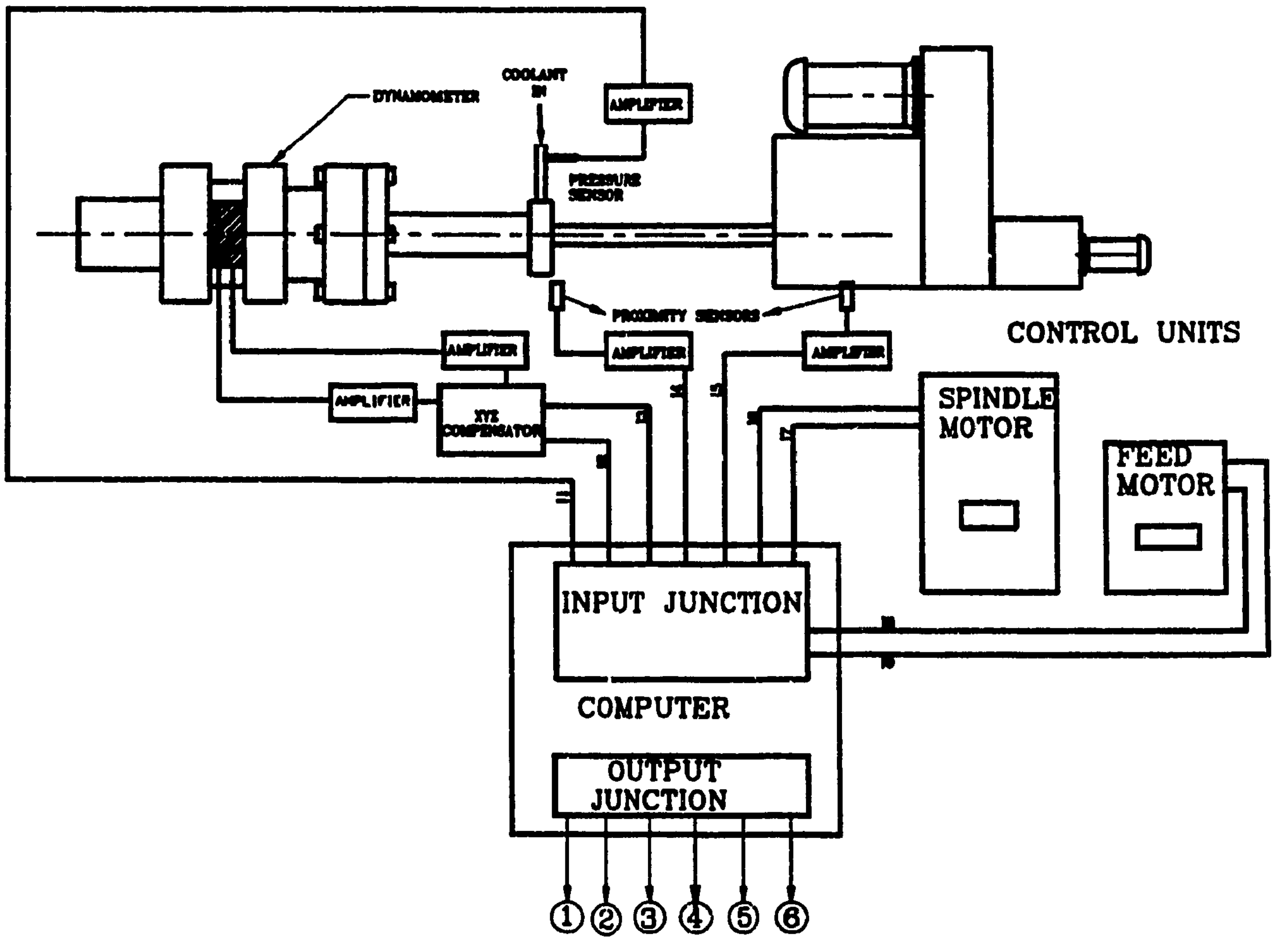


Figure 4.11: Schematic representation of the expert control set up for DHMS.

in the expert control software to ensure the proper resolution. High speed data processing is achieved by binary representation of the acquired data. Voltage losses due to binary conversion is overcome by representing the reference voltage as a 16 bit binary word. the requirements for the implementation of exert control was fulfilled by retrofitting the deep hole drilling machine with variable speed motors. The motor speeds are varied by 0 to 10 VDC supply from the expert controller to the microprocessor terminals of the motor controller. The feed motor reversal for tool withdrawal is enabled by channelling the digital input from the forward terminal to reverse terminal of the microprocessor of the feed motor controller.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 OVERVIEW

Although control system theory provides useful solutions to many of the practical control problems in manufacturing in general and deep hole machining in particular, nevertheless, there still exists a large class of realistic problems which cannot be solved by the existing control system theory alone. These problems are often encountered where the object of control is poorly defined, highly uncertain and can hardly be represented mathematically in its entirety.

To increase the system reliability, and to ensure operational security, human interventions play an important role in the operation of such systems, particulary when the system is subject to contingencies, such as component failures, severe operating condition variations, etc. Therefore expert control was designed and implemented to mimic the human being in dealing with contingency system conditions.

The expert control for DHMS consists of a knowledge base of expertise, a real time information processor, an inference engine, signal acquisition, conditioning and classification system and finally a control mechanism. Thus the expert control of DHMS is an intelligent decision making system which incorporates heuristic as well as mathematical knowledge of all aspects of deep hole machining in suggesting an appropriate corrective action while reasoning the user about these actions.

5.2 EXPERT CONTROLLER IN OPERATION

Expert controller was tested with a test case of deep hole machining operation under the supervision of the expert controller. The details of the test case are shown in Table 5.1. The operation of the expert controller wsas classified into (i) supervised learning cycle and (ii) unsupervised operational period. In supervised learning period, the expert controller operates with available general knowledge about the deep hole machining operation and gathers specific knowledge to update the knowledge base. After several cycles of learning periods the expert controller is ready to carry out the unsupervised operation. The up-dating of the knowledge is done by the knowledge engineer based on the reports generated by the expert controller at the end of each learning cycle. The report generated by the expert controller consists of the mean and peak values of for' es, time, cutting speed ant feed at that instance and the rule fired at that instance. An example of the report of the expert controller for the test case is

Table 5.1 : Details of a test case.

#	ITEMS	DESCRIPTION, UNITS etc.,
1	Workpiece material	AISI 1045, Steel
2	Tool	BTA Solid Boring tool, 22.225 mm diameter
3	Coolant	Shell Garia H
4	Principle	Stationary workpiece, rotating tool
5	Maximum Cutting speed	200 m/min
6	Minimum Cutting speed	30 m/min
7	Maximum Feed	0.2 mm/rev
8	Minimum Feed	0.009 mm/rev
9	Maximum Coolant Flow rate	45 lit./min
10	Minimum Coolant Flow rate	35 lit./min
11	Initial Cutting speed	40 m/min
12	Initial Feed	0.05 mm/rev
13	Initial Coolant Flow rate	35 lit./min
14	Normal Torque range	20-30 Nm
15	Normal Thrust range	1900-2300 N
16	Above Normal Torque range	30-50 Nm
17	Above Normal Thrust range	2300-3150 N
18	Below Normal Torque range	10-20 Nm
19	Below Normal Thrust range	1700-1900 N
20	Missing Tool Torque Limit	1-10 Nm
21	Missing Tool Thrust Limit	20% feed motor load (>350% Normal Thrust)
22	Maximum Peak Torque	230% initial and 200 % after initial phases
23	Maximum peak Thrust	300% initial and 160% after initial phases

illustrated in Table 5.2.

Analysis of the learning report showed that the torque and thrust levels specified were accurate and the knowledge base did not require any modifications at this particular set up. Even though there are no fine tuning required at this juncture, need for careful observation of the reports for every cycle and fine tuning of the knowlege base should be carried out over regular interval of time to keep the system up to date. he most important fall out of these observations' is that the designed chip sensor was not capable of meeting its requirements for the following reasons:

1. Velocity of the swarf is too small and hence the swarf jet had no impact on the diaphragm of the sensor. In order to achieve higher velocity of the swarf an additional supply of coolant into the swarf channel is required. This was not done considering the economic and time constraints on the system.

2. Chips were getting clogged at the nozzle at frequent intervals thus rising false alarm. This could be overcome by increasing the diameter of the nozzle throat but this corrective action will further decrease the velocity of the swarf jet falling on the sensor. Because of the above difficulties the chip sensor was removed and the the chip factor was set to 'NORMAL'.

Table 5.2 : Expert controller report.

TIME	V	S	MT	MF	FO	FF	CF
0	60	0.05	---	---	---	---	---
RULE	----						
15	65.312	0.0603	32.811	7.423	2200.13	252.62	-N-
RULE	FO=2∪FF=2∪MT=3∪MF=3∪CF=2						
20	65.455	0.061	30.401	6.223	2019.22	175.31	-N-
RULE	FO=2∪FF=2∪MT=3∪MF=2∪CF=2						
30	68.192	0.064	28.121	5.131	1982.12	143.17	-N-
RULE	FO=2∪FF=1∪MT=2∪MF=2∪CF=2						
40	70.008	0.088	28.014	4.589	2001.36	132.87	-N-
RULE	FO=2∪FF=1∪MT=2∪MF=1∪CF=2						
50	71.269	0.091	26.323	3.787	1991.15	132.29	-N-
RULE	FO=2∪FF=1∪MT=2∪MF=1∪CF=2						
60	70.112	0.090	27.863	4.901	2200.82	166.65	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=1∪CF=2						
70	77.886	0.092	30.101	5.111	1862.42	121.28	-N-
RULE	FO=1∪FF=1∪MT=2∪MF=2∪CF=2						
80	79.021	0.092	30.152	4.882	1800.86	119.04	-N-
RULE	FO=1∪FF=1∪MT=2∪MF=1∪CF=2						

TIME	V	S	MT	MF	FO	FF	CF
90	81.124	0.101	29.389	4.331	1601.77	186.62	-N-
RULE	FO=1∪FF=1∪MT=2∪MF=1∪CF=2						
100	84.887	0.101	28.163	5.109	1991.81	199.83	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=2∪CF=2						
110	85.100	0.101	29.218	4.320	1904.72	201.12	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=1∪CF=2						
120	87.021	0.105	24.5	4.482	1908.11	190.91	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=1∪CF=2						
130	85.006	0.106	25.106	3.315	1919.29	196.61	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=1∪CF=2						
140	85.006	0.110	24.521	3.412	1963.81	181.61	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=1∪CF=2						
150	85.006	0.110	25.106	4.131	1929.29	200.10	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=1∪CF=2						
160	80.91	0.128	20.111	5.112	1911.6	133.16	-N-
RULE	FO=2∪FF=1∪MT=2∪MF=2∪CF=2						
170	82.86	0.126	22.129	4.209	1908.7	171.26	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=1∪CF=2						

TIME	V	S	MT	MF	FO	FF	CF
180	84.67	0.122	26.701	7.001	1978.0	170.61	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=3∪CF=2						
190	84.91	0.112	24.123	6.012	1981.9	185.18	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=2∪CF=2						
200	85.013	0.112	26.362	5.813	1988.2	161.86	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=2∪CF=2						
210	85.003	0.115	28.321	4.809	2011.6	170.07	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=1∪CF=2						
220	87.166	0.125	26.163	4.506	2001.8	182.67	
RULE	FO=2∪FF=2∪MT=2∪MF=1∪CF=2						
230	87.168	0.125	24.199	4.513	2112.5	190.01	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=1∪CF=2						
240	85.011	0.125	26.5	4.386	1988.3	176.10	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=1∪CF=2						
250	85.011	0.125	24.323	3.890	1974.5	166.63	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=1∪CF=2						
260	85.011	0.125	20.119	3.667	1909.4	132.09	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=1∪CF=2						

TIME	V	S	MT	MF	FO	FF	CF
270	90.126	0.112	26.332	4.131	2001.54	188.76	-N-
	FO=2∪FF=2∪MT=2∪MF=1∪CF=2						
280	90.126	0.112	28.112	3.998	2200.12	191.01	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=1∪CF=2						
290	90.126	0.112	26.665	4.088	2219.91	200.16	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=1∪CF=2						
300	90.148	0.116	28.881	5 343	2299.88	233.58	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=2∪CF=2						
310	90.148	0.116	29.002	5.666	1990.82	187.98	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=2∪CF=2						
320	90.148	0.116	28.882	5.001	1919.09	166.77	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=2∪CF=2						
330	90.148	0.116	26.666	5.883	2001.11	173.57	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=2∪CF=2						
340	90.148	0.116	29.998	5.969	2209.65	189.73	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=2∪CF=2						
350	90.148	0.116	27.377	5.066	2499.07	202.36	-N-
RULE	FO=2∪FF=2∪MT=2∪MF=2∪CF=2						

The reports generated by the expert controller in both supervised learning cycle and unsupervised opertion period were examined to determine the efficiency and performance characteristics of the expert controller and the effect of expert control on the quality of work piece and economy of deep hole machining with respect to tool wear.

5.2.1 RULE FIRING

The expert system which forms the sole of the expert controller is designed to make decisions based on the time available for making a decision and implementing the decision. The high speed decisions were made by meta rules while the remaining rules in the knowledge base were concerned with decision making at relative leisure. This high speed decision making was found very useful especially during emergency situations namely, tool failure and chip blockage. It is quite interesting to note that the meta rules for high speed decisions are used less than the other rules in the knowledge base. This is due to the fact that the expert controller tends to keep the cutting force and torque within its normal range instead of a fixed normal value. The signals were continuously monitored for their peek values crossing the abnormal range.

5.2.2 SYSTEM STABILITY

There is a tendency in the expert controller to constantly vary the cutting parameters when the signals indicate a normal machining operation. This is done to maximize the productivity. Such a change in the cutting parameters are continued till the signals indicate an abnormal machining condition. At this stage the expert controller varies the cutting parameters again to bring the signals back to normal machining operation. This results in an unstable machining condition. In order to overcome this problem a range of normal parameters is used instead of a fixed value. Corrective actions are taken only when the signal grows above this limit and no actions are taken while the parameters are within this range.

Continuous analysis of all the signals will make the system unstable and the time requirement for processing such signal highly critical. The signal classification into various clusters takes care of this problem. More over the problem of 'Overflow' [Chapter 4] is eliminated by signal classification by mean and peak values.

Faster the data transfer, higher the system stability. Conversion of signals into engineering numbers such as integers and decimal values consume a considerable amount of time in data conversion whereas a binary system is highly time efficient. It was observed that the data transfer can be done at a speed of 10MHz when using binary values where as the speed reduces to less than 6MHz for decimal values.

5.2.3 SYSTEM PERFORMANCE

The system was tested for its performance by:

1. cutting of one of the sensors inputs.

   The expert system is designed in such a way it operates only when all the signals comprising a premise are present. When one or more of these signals are not present the values of the signals are set to zero and the expert system sends an alarm to the screen. This checking is accomplished by **'scanf(&fo, &ff, &mt, &mf, &cf);'** code. This code fails when one or more of the variables cannot be scanned. The motors will not be actuated if they were not yet started. If the process is already in progress, then failure of one or more of the sensors will switch the operation to manual mode. This is indicated by a sound alarm by the computer.

2. Machining with a failed tool.

   A broken tool was mounted on to the boring bar and the machine was started to evaluate the effect of such a tool on the expert controller. The expert controller stopped the machining operation before the tool penetrated the workpiece by 0.1mm. This tool penetration of 0.1mm was due to time required for sensing and stopping the operation.

5.3 REALIZATION OF REAL TIME CONTROL

Consider a tool failure during a machining operation. When the flank surfaces wears out by 0.3mm or brakes (at tool failure the flank loss is assumed to be at least 0.3mm). The broken tool will cause damage to the workpiece with the flank surface hits the premachined surface which is approximately at a distance off 0.3mm from the flank face. In order to minimize the damage to the workpiece from the broken tool the expert controller should be able to stop the feed before the tool travels by 0.3mm. Figure 5.1 illustrates the time required at various speeds and feed rates, for the tool to travel by 0.3 mm. This gives the maximum time available for the control system to stop the tool at a given speed and feed rate. Table 5.3 illustrates the various time requirements for expert controller and conventional controller. It is interesting to note that, from the Table 5.3 and Figure 5.1, a conventional control system with high speed optimization routine can be used for real time control of deep hole machining only when the cutting speed is less than 40meters per minute and a corresponding feed rate of less then 0.005 mm per revolution. This range of cutting speed and feed rates are highly un economical and cannot be used for producing workpiece of high quality as the surface roughness at these parameters will be very high. In addition, it carries all the draw backs of slow speed machining such as built up edge etc. It can be seen

Table 5.3 : Comparison of time requirement of emergency feed stop.

#	Expert controller		Conventional controller	
	Time specification	**Time**	**Time specification**	**Time**
	Data Acquisition Processor &sensor.			
1	Conversion time (A/D)	5μsecs	Conversion time	5μsecs
2	Settling time	4μsecs	Settling time	4μsecs
3	Aperture time	5nsecs	Aperture time	5nsecs
4	Acquisition time	1μsecs	Acquisition time	1μsecs
5	Hold time/ Droop rate	1μsecs	Hold time/ Droop rate	1μsecs
6	Conversion time (D/A)	1μsecs	Conversion time (D/A)	1μsecs
7	Sensor Response	5μsecs	Sensor Response	5μsecs
	Expert control & Supporting systems			
7	High speed Decision	40msec	Optimization time (min)	7secs
8	Max Slow Decision	480msec	Optimization (max) >	300sec
	Actuator & Supporting system			
9	Control unit response	5μsecs	Control unit response	5μsecs
10	Brake time	0.1sec	Brake time	0.1sec
	Minimum Time Required	**0.14021 seconds**	**Minimum Time Required**	**7.04021 seconds**

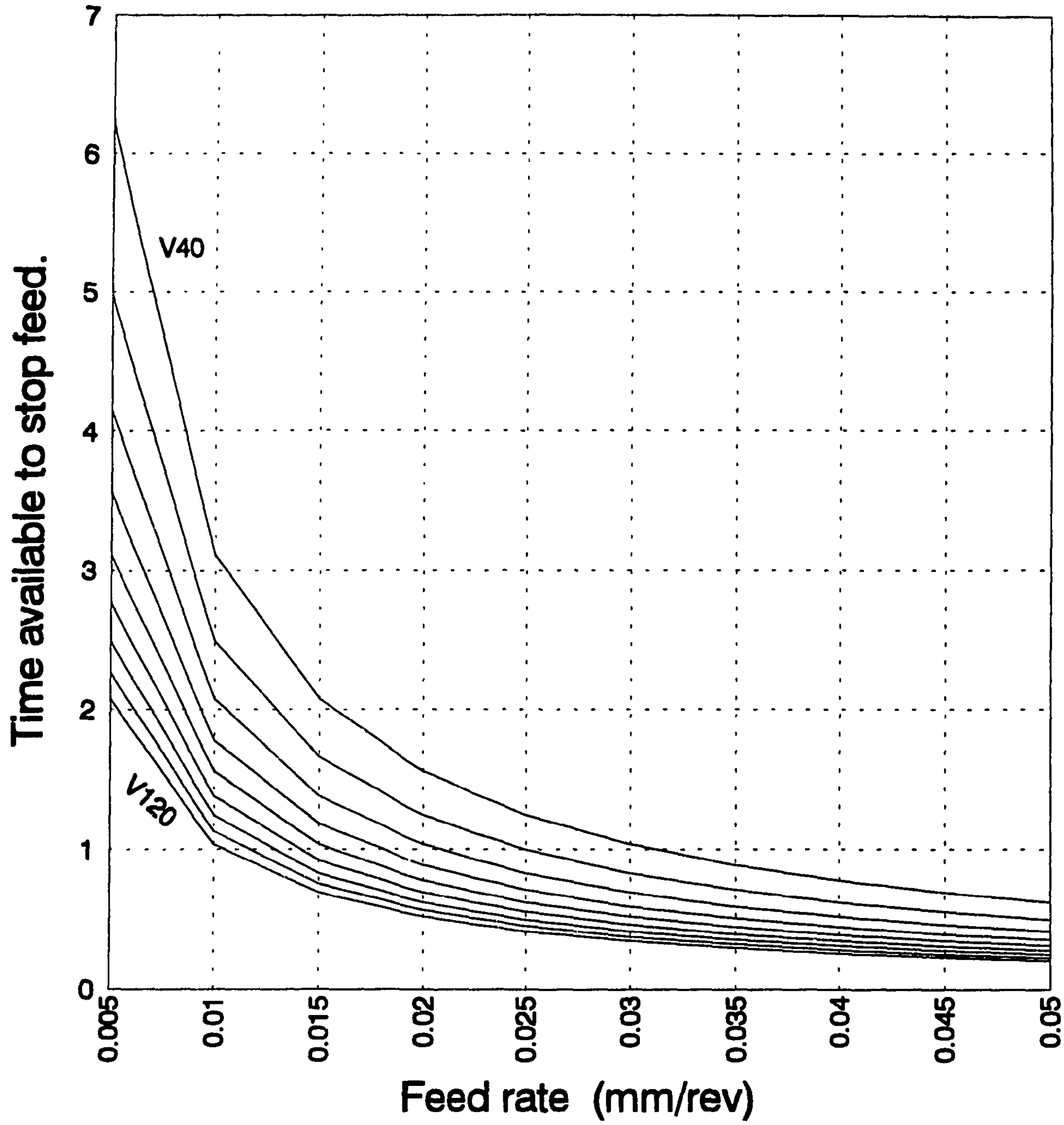


Figure 5.1: Time required for DC braking at various cutting parameters.

that the expert controller can be operated at the entire range of speed and feed rates. The time constraints at the higher speed and feed ratios are higher than the minimum time requirements of the expert controller. Moreover, the expert controller is capable of predicting the tool failure before the actual failure takes place and initiates lowering of speed and feed rates. Thus when the actual failure occurs the cutting speed and feed rates are at the lowest permitted values. This results in increase in the available time to its maximum as illustrated in Figure 5.1. This shows that with an expert controller DC breaking time of the motors can be set to a higher time value. This DC braking of the motors over a longer time period reduced the amount of heat produced in the motors and hence the demand for adequate cooling facilities for the motor before restarting, is reduced.

5.4 EFFECT ON FORCES, AND CUTTING PARAMETERS

The variation of cutting speed and feed rate over the length of a work piece is illustrated in Figure 5.2 and 5.3. It is interesting to note that the forces, both torque and thrust, despite the changes in cutting parameters, were held relatively stable by the expert controller. This is illustrated by Figure 5.4 and 5.5. It was observed that the feed rate reduced when the cutting tool entered the region where the workpiece was held by the chuck jaws indicating excessive tightening of the workpiece. To study the effect of over tightening of the workpiece by the chuck haws, the jaws were severly

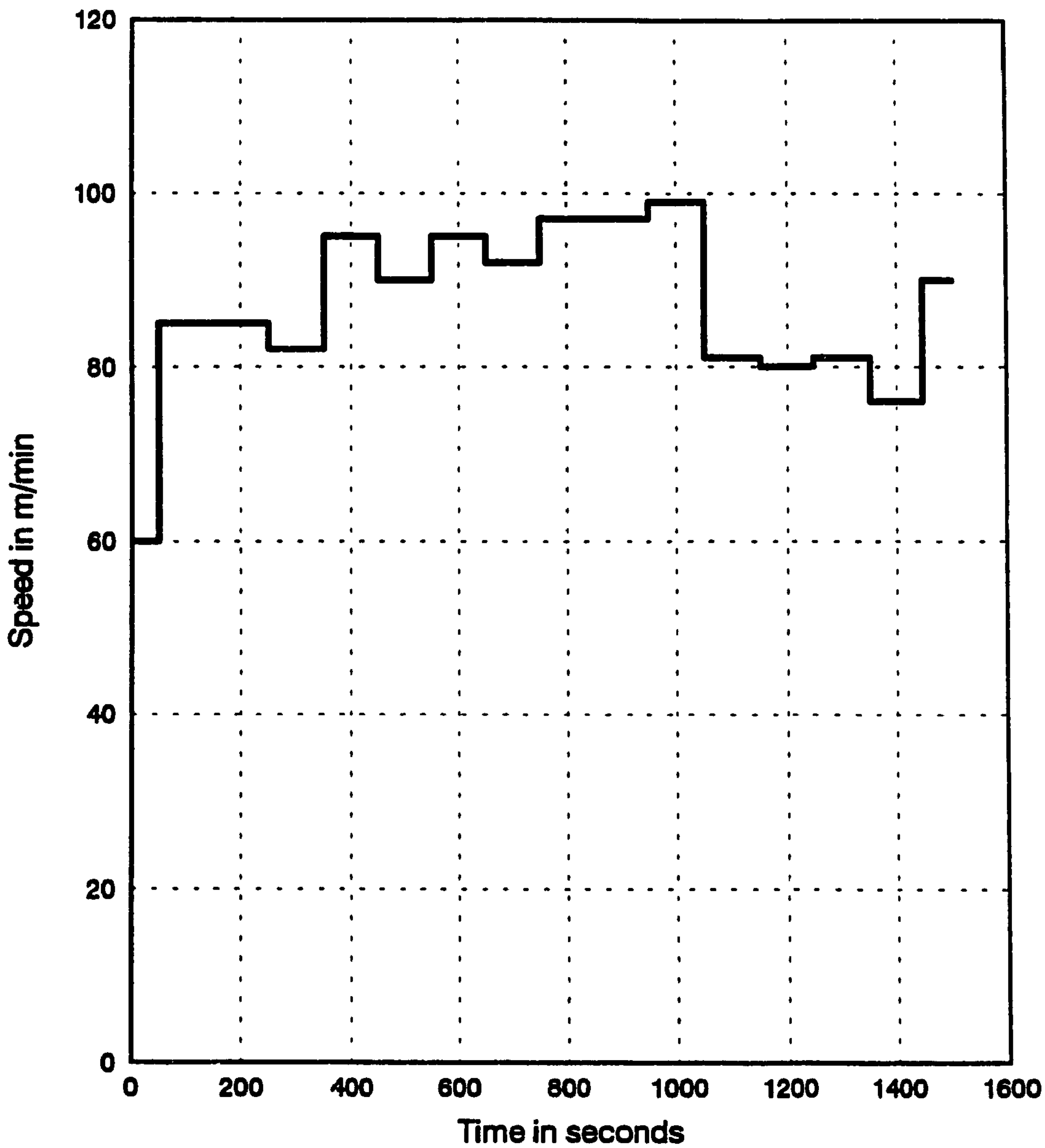


Figure 5.2: Variation of cutting speed by the expert controller.

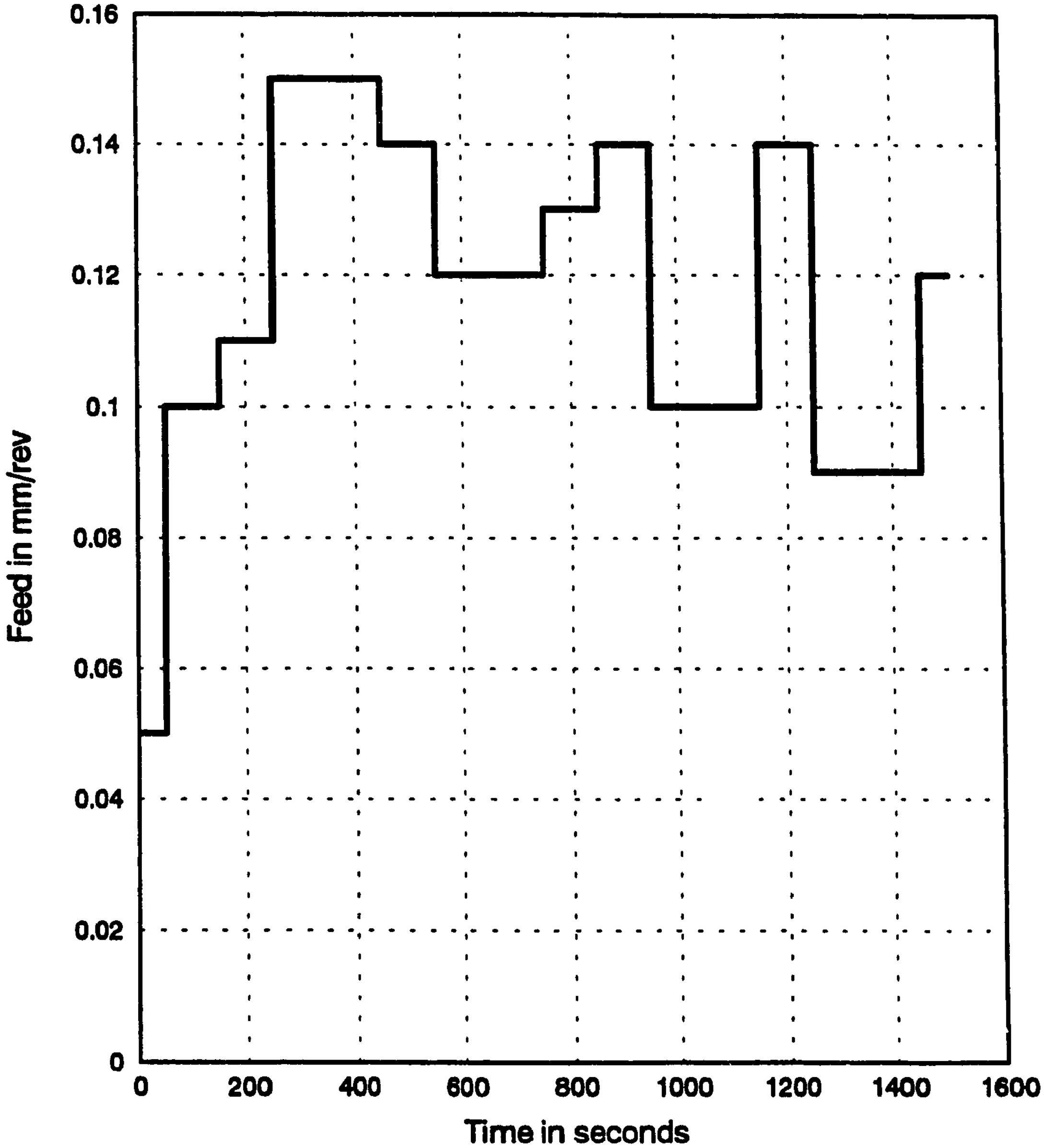


Figure 5.3: Variation of feed by the expert controller..

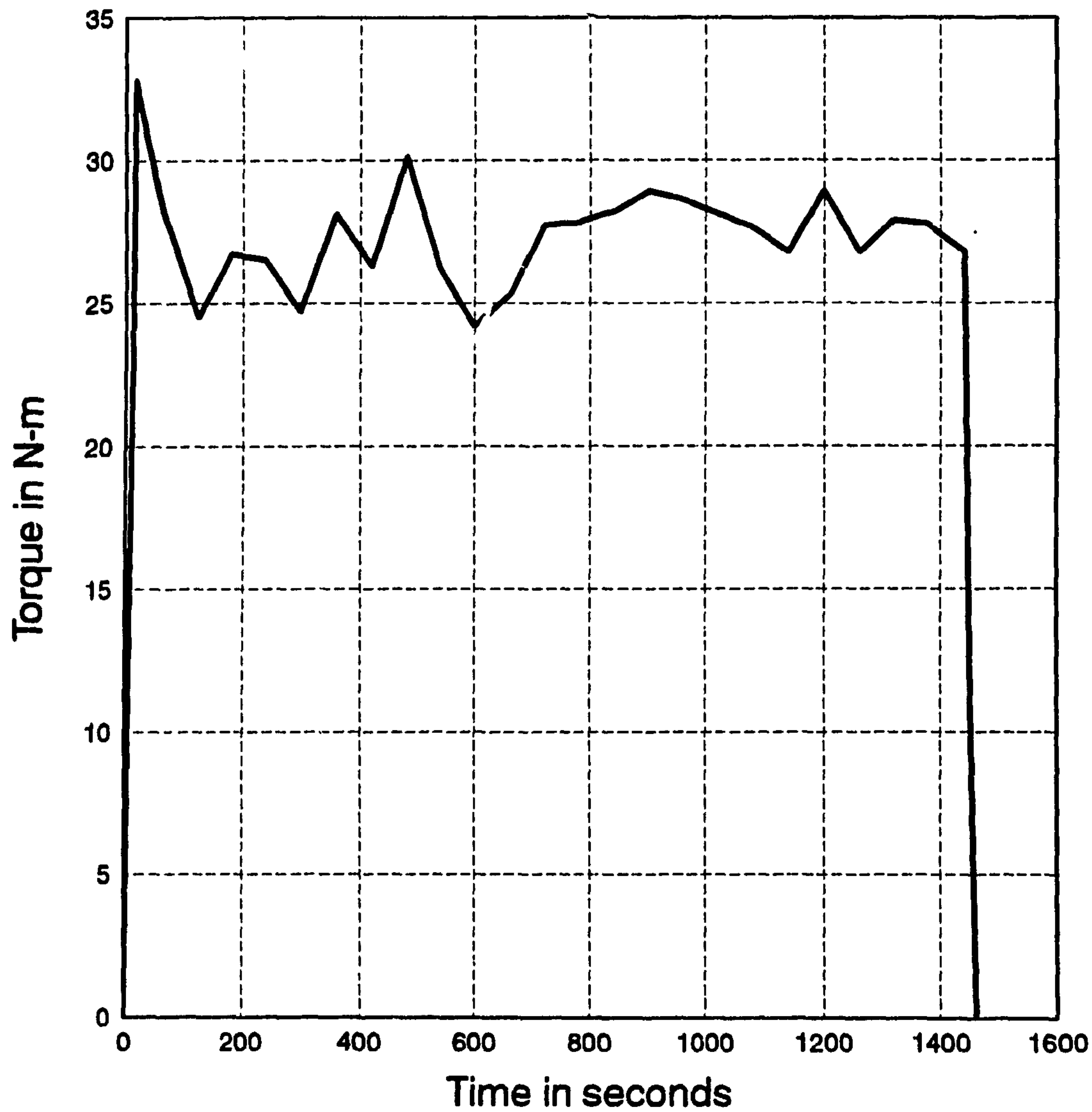


Figure 5.4: Torque readings during machining.

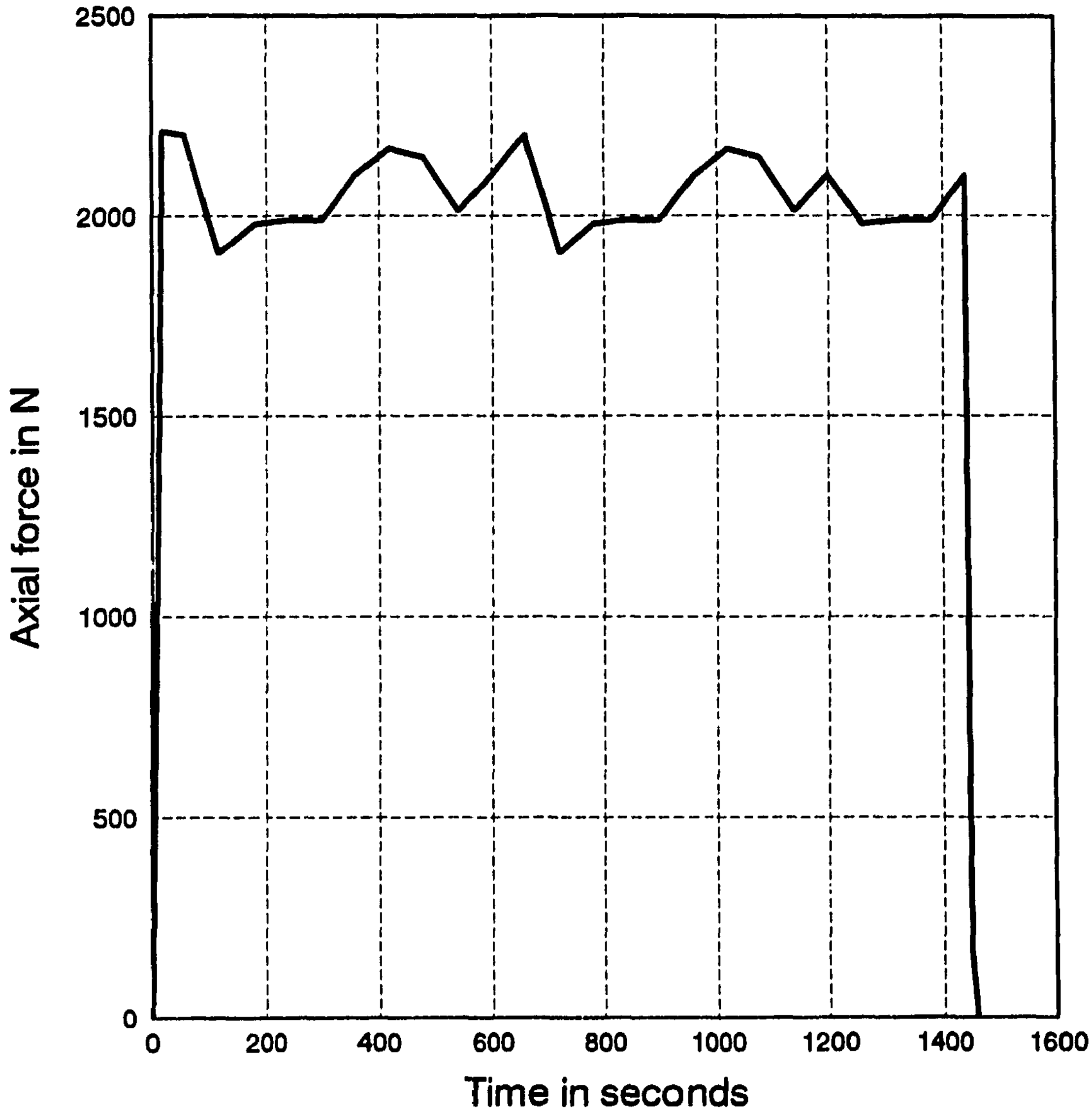


Figure 5.4: Thrust readings during machining.

tightened using a hammer. Surprisingly, the machining operation ceased at this cross section and the expert controller halted the tool advance. Repetition of the experiment with normal load on the chuck jaws did not stop the machining at this corss section.

5.5 EFFECT ON THE WORKPIECE

Figure 5.6 illustrates the surface finish of the work piece machined under the supervision of the expert controller. The statistics of the surface finish measurement over the entire length of the workpiece is illustrated in Table 5.4(a). It can be noted that the expert control resulted in fairly constant surface finish over the length of the workpiece. Figure 5.7 illustrates the roundness of the hole produced. The roundness of the holes at different cross sections is measured by slicing the tube at different lengths and the statistics of roundness measurement is shown in the Table 5.4(b).

5.6 EFFECT ON THE TOOL

Figure 5.8 illustrates the wear rate of the cutting tool for both expert controlled machining and conventional machining with out control. It can be noted that the rate of wear of the cutting tool was fairly constant when used under the supervision of the expert controller while the wear rate of the tool under conventional un controlled machining the wear rate was higher and constantly changing.

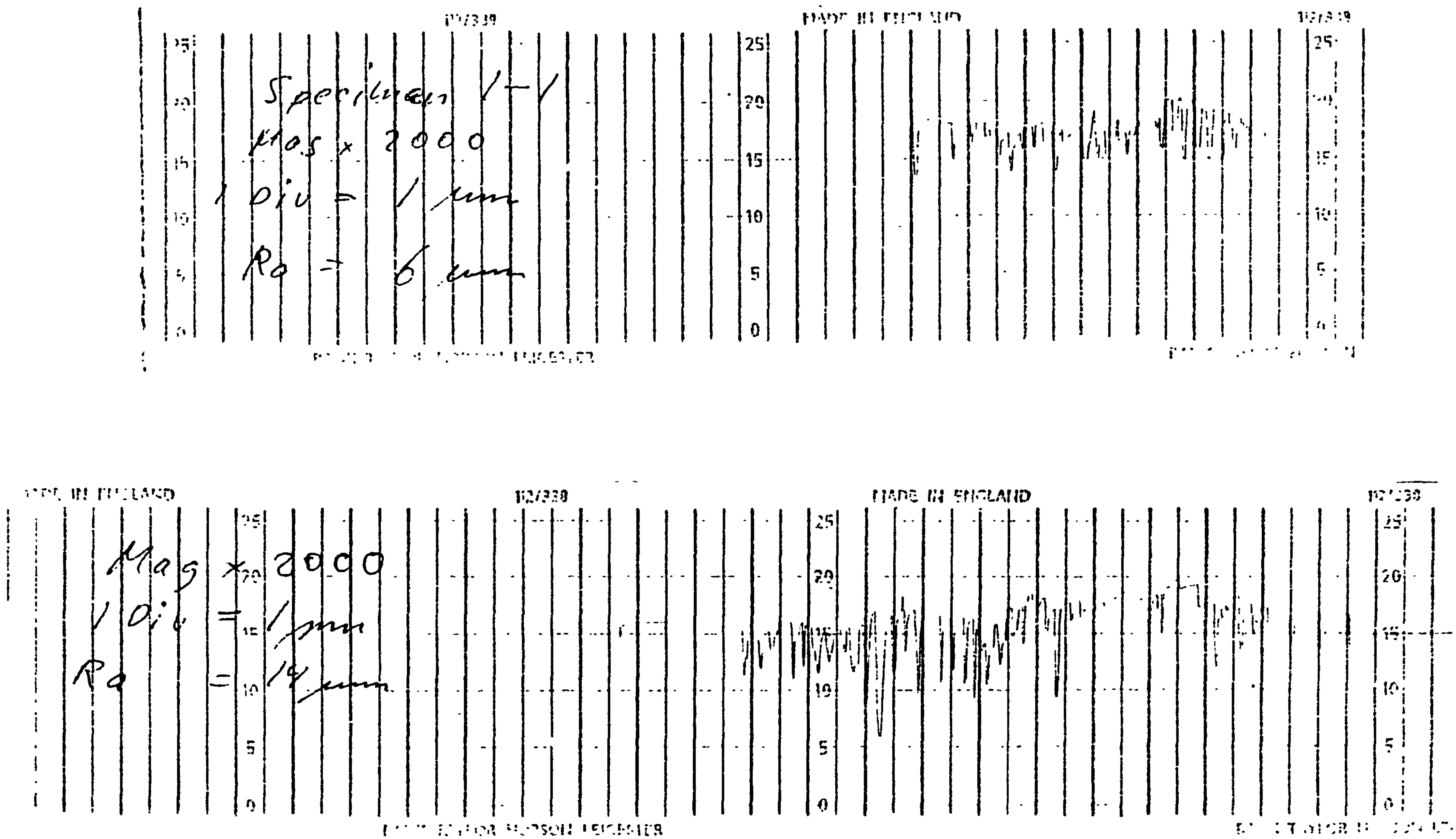


Figure 5.6: Surface finish examples.

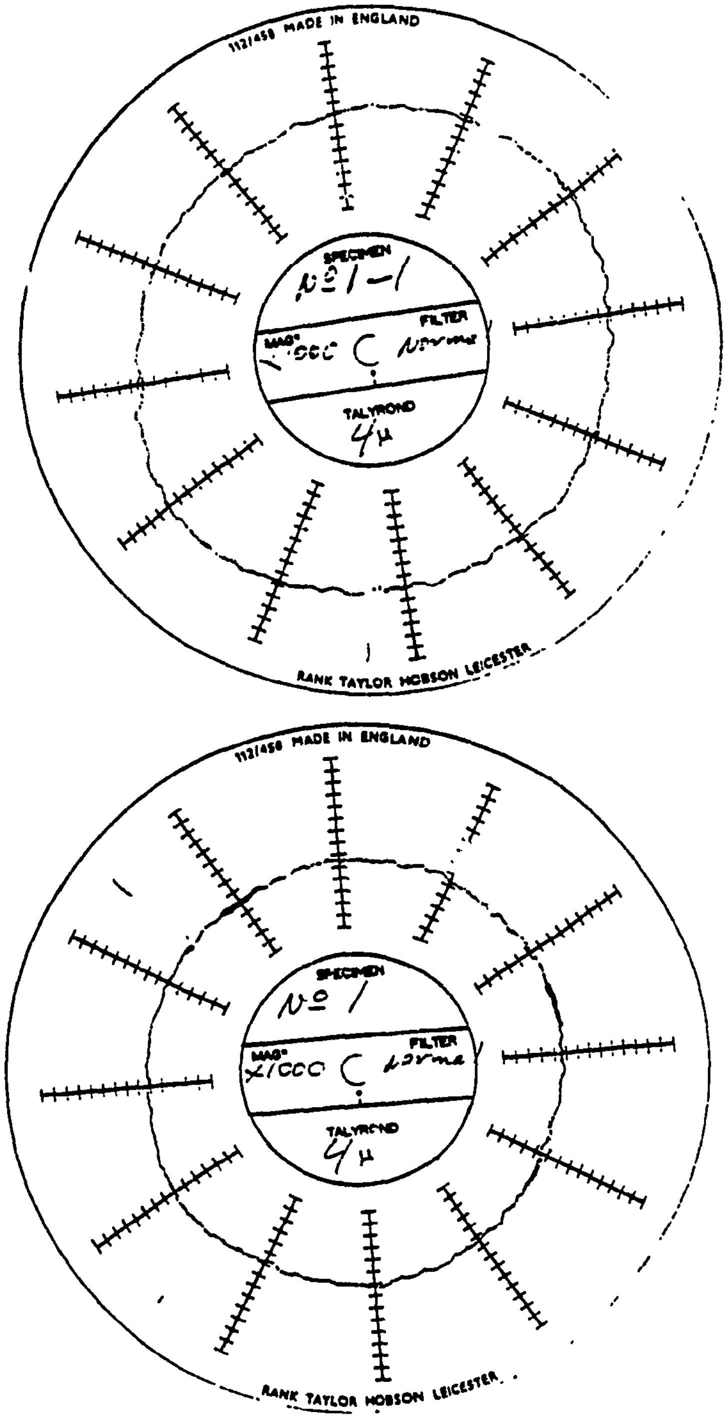


Figure 5.7: Roundness measurement examples.

Statistics of the surface finish measurements (Ra) µm	
Minimum : 3	Maximum : 20
Range : 17	Median : 9.5
Mean 10.214	Std. Deviation : 4.95

Table 5.4 (a): Statistics of surface finish measuremtns for the test case

Statistics of the roundness measurements (µm)	
Minimum : 3	Maximum : 11
Range : 8	Median : 6.5
Mean : 6.7	Std. Deviation : 2.75

Table 5.4 (b) : Statistics of the roundness measurements for the test case

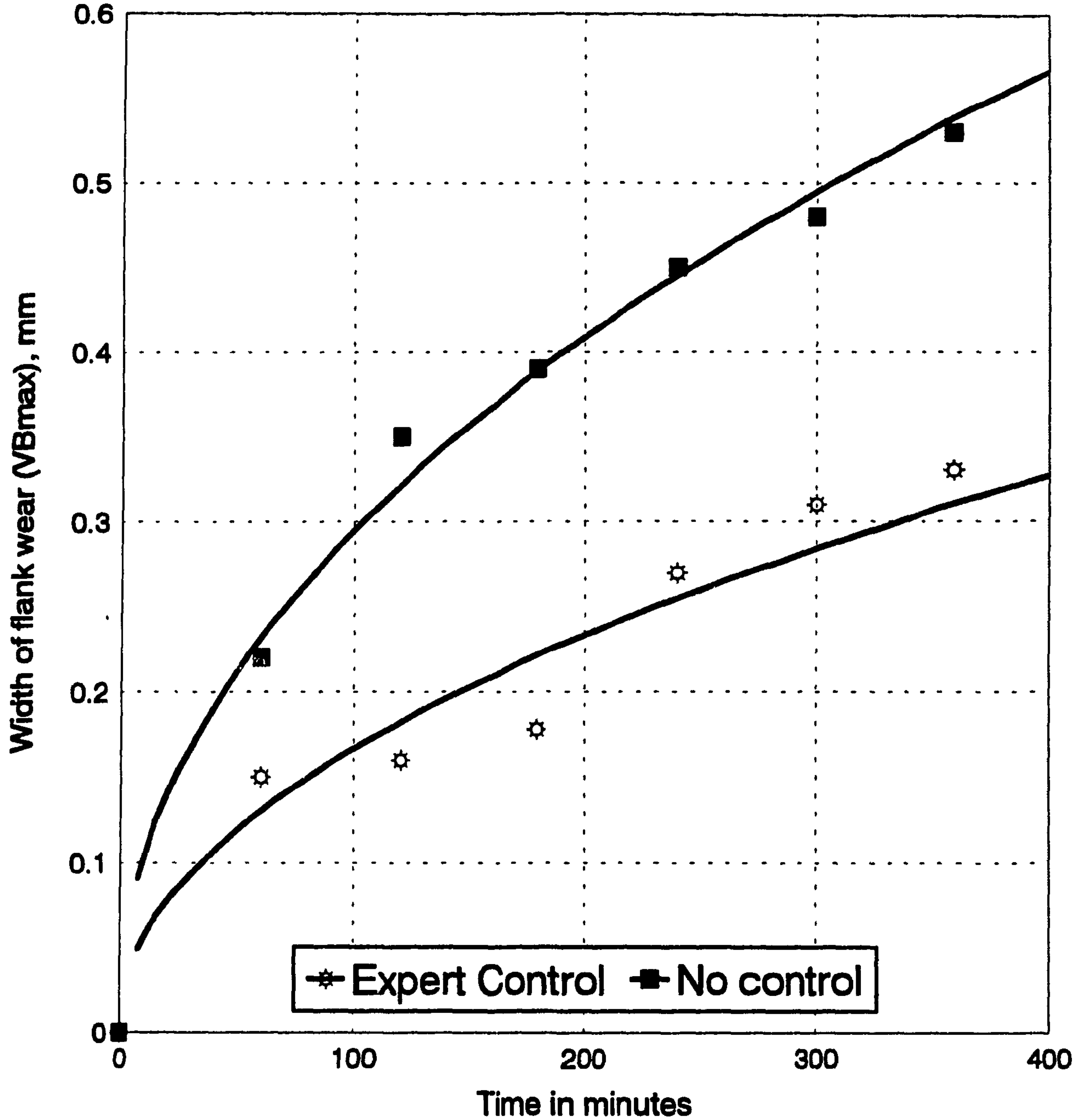


Figure 5.8: Comparison of flank wear rates.

5.7 CONCLUSIONS:

Real-time control of machining process in general and deep hole machining process in particular is becoming increasingly important. The challenge to the realization of real-time control lies in the complexity and uncertainty inherent in these processes. Controls strategies based on mathematical models, thus enable to overcome these difficulties, have failed to reach beyond the thick bounds of scientific journals. A control scheme based on artificial intelligence technique namely, expert systems was developed to realize the real time control of complex processes like deep hole machining systems. This research resulted in the development of expert control, its implementation and validation.

To utilize the capabilities of the expert systems, which forms the soul of the expert control, the problem domain is represented by symbolic models taking into account the various events, their sequence, instance and intervals. The machining condition was described using five process descriptors. These process descriptors are represented as the premise of the production rules of the knowledge base The conclusion part of these production rules are supplied by the experts in the domain of deep hole machining. It is well known that two experts need not suggest the same reasoning. To accommodate this difference of opinion of various experts a conflict resolution mechanism was developed and implemented. Two modes of expert control development was suggested for overcoming the lack of expertise available in the entire domain. Available expertise was put to test under supervised learning operation and then the updated knowledge base was adopted for unsupervised operation of the deep hole machining system under expert control.

Knowledge base is the brain of the expert control system and the completeness of the domain knowledge represents the degree of expertise incorporated into the expert controller. A large amount of available research results are used to develop the knowledge base. However, lack of research publications regarding the flow and pressure distribution of coolants and their effect on the performance of deep hole machining lead to the investigation of the effect of coolant 'quantity' on the deep hole machining performance. This investigation revealed that the coolant 'quantity' is an important contributor to the performance and economy of deep hole machining operation and hence be incorporated in to the control strategy. The pressure distribution in the machining zone was improved by increasing the coolant supply to the machining zone. This resulted in improvement of tool life and chip removal. However, it was also observed that there exists a maximum coolant flow rate and any increase in the flow rate beyond that range will result in accelerated tool wear and hence lower tool life. Thus, the significance of the limiting values of coolant flow were emphasized. A relation between cutting speed (boring bar velocity) and the flow loss was also established for increasing the flow with increase of cutting speed.

Implementation of expert control requires sensors to sense the process responses, signal conditioning to filter out the noise, compensate for the cross coupling of various channels and amplification to 0-10 VDC voltage values to be used by the expert controller. These voltage readings are transformed to appropriate symbols for determining the machining condition. The inference engine determines the actions required based on the machining condition described by the incoming signals and the matching rules in the knowledge base. The actions, thus determined are implemented under the supervision of temporal rules and procedural rules. The actions are communicated to the actuators in the form of analog voltages ranging from 0-10 VDC. The microprocessor in each controller interprets these voltages at different channels and vary the motor speeds accordingly. Accurate conversion of output voltages from the reference voltages was accomplished by using 16 bit binary conversions. Antialiasing, software digital filtering etc., are incorporated to ensure the system performance.

Finally, the validation of expert controller was done by means of a case study. The results demonstrated that:

1. Expert control is realizable.
2. Expert control is flexible as it can allow continuous improvement to its knowledge base mimicking the human learning process over a time period.
3. Expert control is reliable and can handle emergency situations better than the

conventional control methods.

4. Expert control increases the economy of the deep hole machining process.
5. Expert control gives better or comparable hole quality in deep hole machining.
6. Constant improvement of the knowledge base ensures that even when the experts in deep hole machining leave the industry or laboratory, their expertise can be continuously utilized and hance less demand on training a new employee.

Thus, it can be concluded that the expert control can perform real-time control of deep hole machining along with sequence control and safety monitoring. This is more than a combination of CNC, Adaptive control and other safety management systems at much lower cost. Hence, the future of expert control is boundless and this research is just the beginning. I am sure that, expert control is for real and will reach beyond the annals of scientific journals.

CHAPTER 6

SUGGESTIONS FOR FURTHER RESEARCH

The expert controller developed in this research has shown that the expert systems can be successfully used for real time control of complicated systems like deep hole machining system. Even though the operation of the expert controller is satisfactory from the point of view of research and development, further modifications are necessary to make it more efficient. In this regard the further modifications are suggested in the following areas:

1. Computer architecture.
2. Deep hole machining system
3. The chip sensor.

6.1 COMPUTER ARCHITECTURE

The computer architecture used in the development of expert controller is based on Von Neumann Architectures. This architecture is basically developed for numeric processing. Processing in expert systems is primarily non numeric and does not run efficiently on the conventional Von Neumann machine. Symbolic knowledge representations used in expert control systems are fundamentally different from those of numeric processing. Symbolic operations are memory intensive. A Von Neumann architecture presents a processor to memory bottleneck when intensive irregular memory access are made [Hwang, 1987]. Additionally, the von neumann architecture is fundamentally sequential in nature and does not utilize concurrency to increase performance [Hodson, 1991].

The dataflow computer [Dennis, 1988; Herat, 1988] is another general purpose system which is non-von neumann in design and addresses some of the shortcomings of the von Neumann architecture. The data flow computer moves away from the idea of centralized sequential control. The execution of a data flow program proceeds as data becomes available. The data flow approach has the potential to exploit a high degree of concurrency efficiently [Tiber, 1984]. However, data flow designs are relatively recent and are still controversial. some of the potential problems in the general purpose data flow computer are [Hodson, 1991];

- # a data driven approach at the instruction level can cause excessive pipelining overhead.
- # data flow programs tend to waste memory and increase program length.

when there is a large number of instructions and processing elements, the system can be cost-prohibitive.

There have been many multiprocessor computer systems developed. Some of them are:

1. C.mmp (16 PDP-11s interconnected by a crossbar)
2. IBM 370/168MP (dual processor with shared memory)
3. Tandem 16 (16 processors with dual common busses)
4. Cray X-MP (dual processors, shared memory, pipelining)
5. SPUR (6 to 12 RISC like LISP processors, common bus)

The individual processors within these systems have a von Neumann structure, although some systems like the Cray X-MP support a high degree of pipelining. Thus these systems suffer from many of the disadvantages found in the single processor von Neumann computer when processing symbolic information used in expert systems. Moreover the cost of implementing expert control for DHMS on these machines cannot be justified.

A special purpose processor which can over come the deficiency in the above architectures and cost effective can enhance the performance of the expert controller to a great extent. Design of such a system and the logical order of the deign process is discussed.

6.1.1 TOPOLOGY

A logically reconfigurable multiprocessor topology can support the critical timing requirements of real-time expert controllers. The multi processor aspect of the topology allows for high level parallelism in the system, which will facilitate system responsiveness. it also enables the system programmers to dedicate processing elements to specific tasks. This is often the case in real time control of DHMS, some processor can be used as dedicated low level controllers and others can be used for high level decision making, directing the low level controllers. Physical processor configuration for such a topology is illustrated in Figure 6.1

6.1.2 PROCESSOR STRUCTURE

A special hardware should be provided to support the processing requirements of expert controller. Since the rule-based paradigm has been chosen as the foundation of the knowledge representation, the modularity and incremental development features of rule based design aid in the organization of information in complex systems. Exploiting parallelism in a rule-based system will improve system performance, allowing the timing requirements to be achieved. A suggested design of processing element is illustrated in Figure 6.2. Each expert control units are interfaced with the environment (DHMS) and these expert control units may initiate actions by

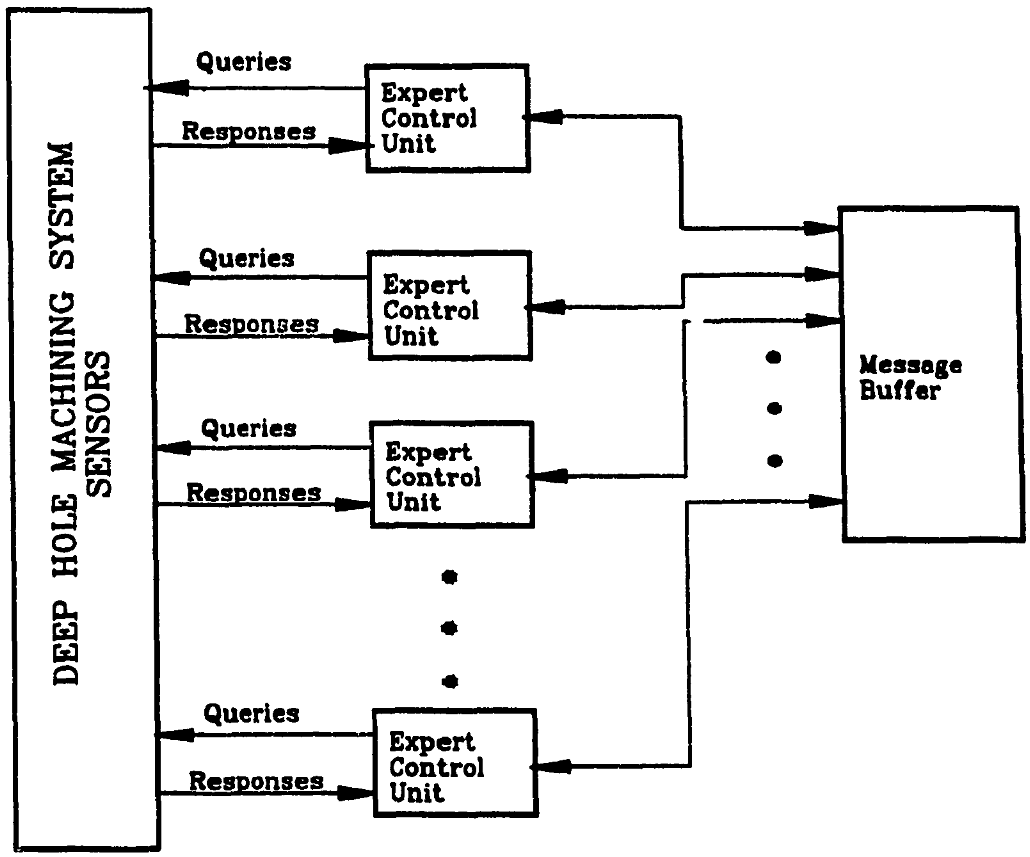


Figure 6.1: Physical processor configuration.

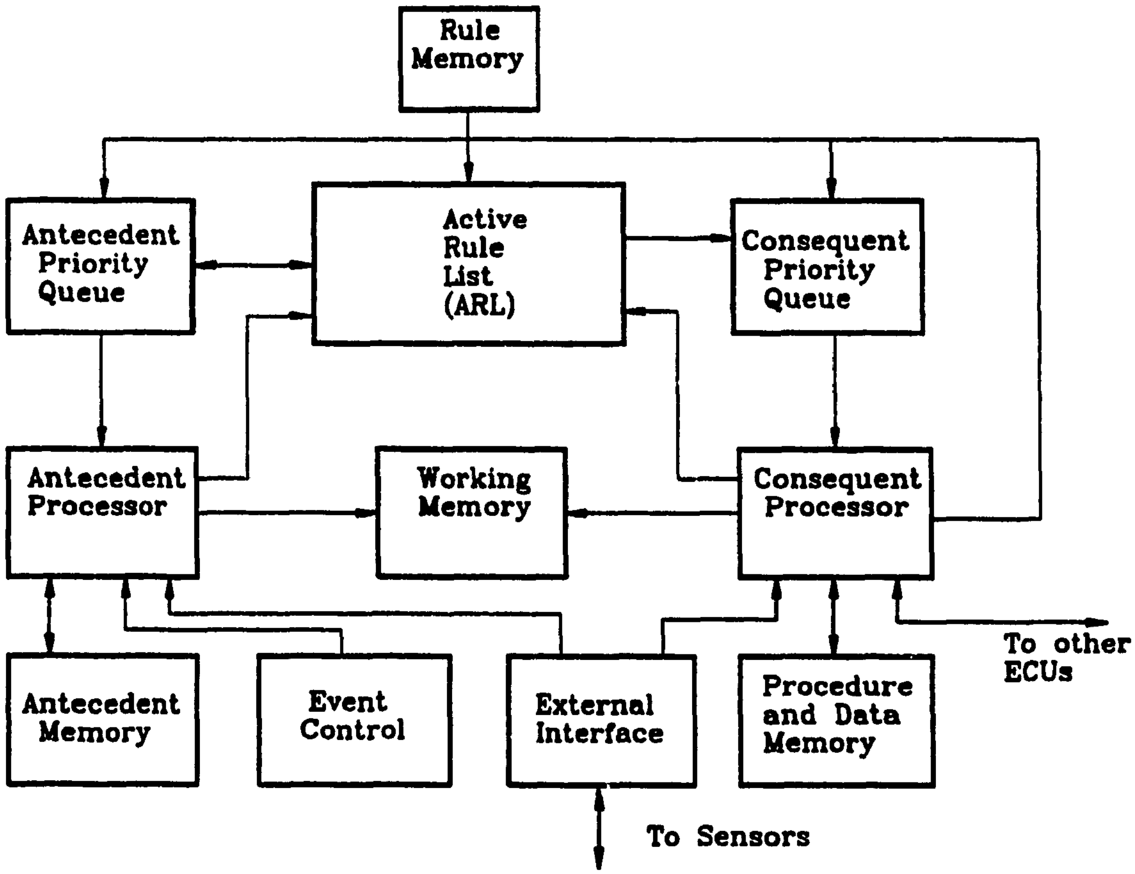


Figure 6.2: Processing element construction.

themselves or send a message to the message buffer to be picked up by a supervising expert system for further processing and action. The Active Rule List (ARL) controls the inferencing and the Antecedent processor and the consequent Processor evaluate the antecedents and perform actions specified by the consequents. The working Memory is used for holding results and intermediate values. Flow of data through the functional unit within the processor is illustrated in Figure 6.3.

6.1.3 INFERENCE MECHANISM

The inference engine used in the expert control system for DHMS is a forward chaining (data driven) system. Backward chaining performs a depth first search of the solution space starting from the system goal and decomposes the problem into subgoals. An extension to the backward chaining approach to improve system responsiveness is to allow the control mechanism to query the real-time environment directly for information. The system can also wait on information or the occurrence of an event to trigger inferencing.

6.2 DEEP HOLE MACHINING SYSTEM

The expert control was implemented on a centre lathe modified into deep hole boring machine. The variation in the motor speeds was achieved by means of AC inverters.

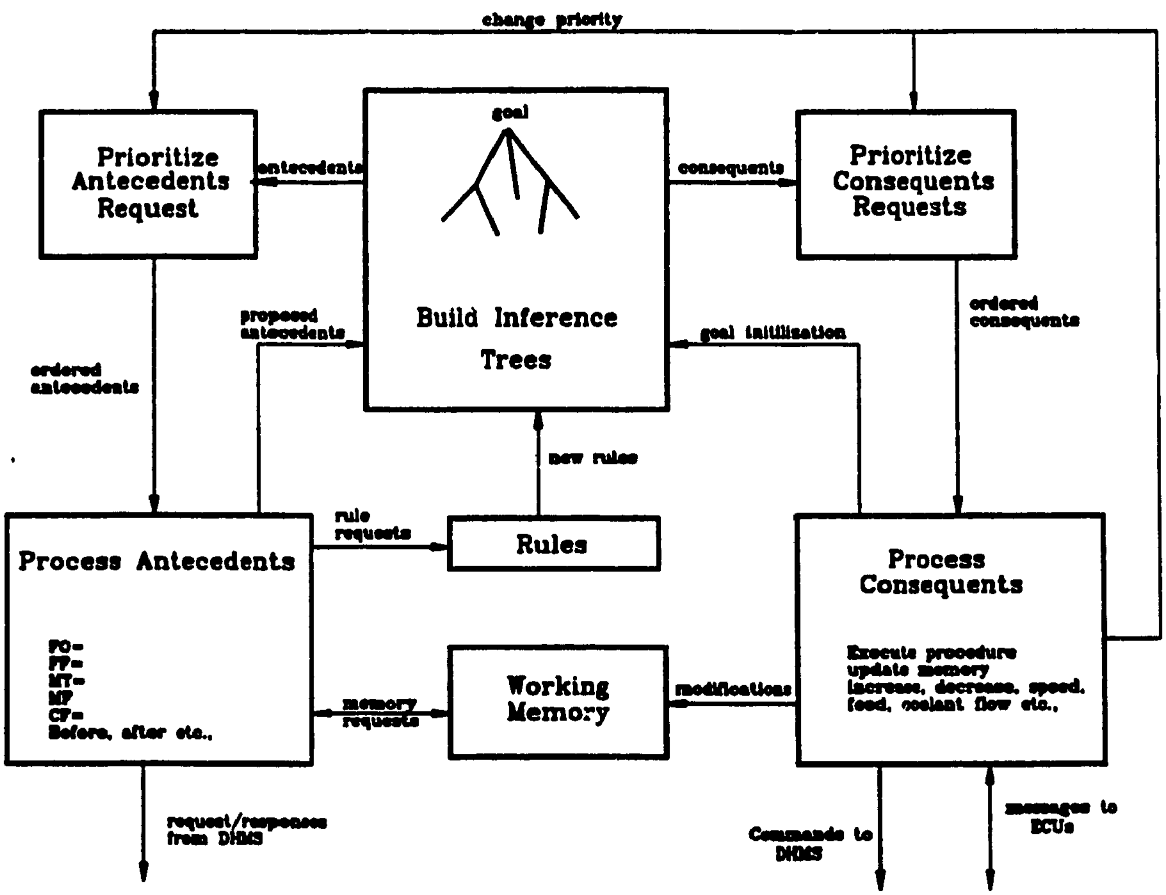


Figure 6.3: Data flow within the processing elements.

The deep hole boring tool was fed into the stationary workpiece. The feed is achieved by the feed motor and lead screw assembly. The lead screw assembly has very high friction coefficient and hence the replacement of lead screw by a ball screw is suggested.

The AC motors used on the deep hole machining system has many disadvantages. The motors have longer response time, slip, and need constant maintenance. In addition, an uncontrolled increase in the supply frequency beyond 120Hz can be dangerous and hence the motors have to be explosion free. To over come these draw backs Brush less servo motors are suggested. These motors meets the needs for fast response, accurate control and maintenance-free operation.

The system time block of the expert controller keeps track of the machining time by monitoring the change in speed and feed. Using the system time block to monitor the depth of the hole induces severe time tax on the expert controller. The out come of mathematical manipulation in the system time block depends on the efficiency of the system programmer and hence amenable to errors. Thus a linear encoder installed on the machine to track the movement of the cutting tool is more reliable and the out put can be monitored by an independent controller for sequence control and ending the machining cycle.

6.3 THE CHIP SENSOR

The sensor developed for sensing the chip factor has failed primarily due to the chip clogging at the nozzle and low swarf velocity. One way of overcoming this problem is to use ultrasonic sensors based on doppler effect. This can be achieved by using ultrasonic Doppler flow meters. The basic principle of operation employs the frequency shift (Doppler Effect) of an ultrasonic signal reflected by suspended chips. Figure 6.4 illustrates the construction of ultrasonic Doppler flow sensor. The transducer is mounted on the exterior of the pipe. It is driven by a high frequency oscillator in the transmitter, through an interconnecting cable. The transducer generates an ultrasonic signal which it transmits through the pipe into the flowing liquid. The transmitter measures the difference between its output and the input frequency and converts this different frequency into electronic pulses which are processed for an analog indication and voltage or current output signal. Additionally, the pulses are scaled and totalized for flow quantity. The transmitter also incorporates a circuitry which allows adjustment of the signal threshold. This permits elimination of undesirable ambient noises (both mechanical and electrical). As a result instrumentation is possible in a variety of locations subject to high levels of sonic, mechanical, and electrical noise.

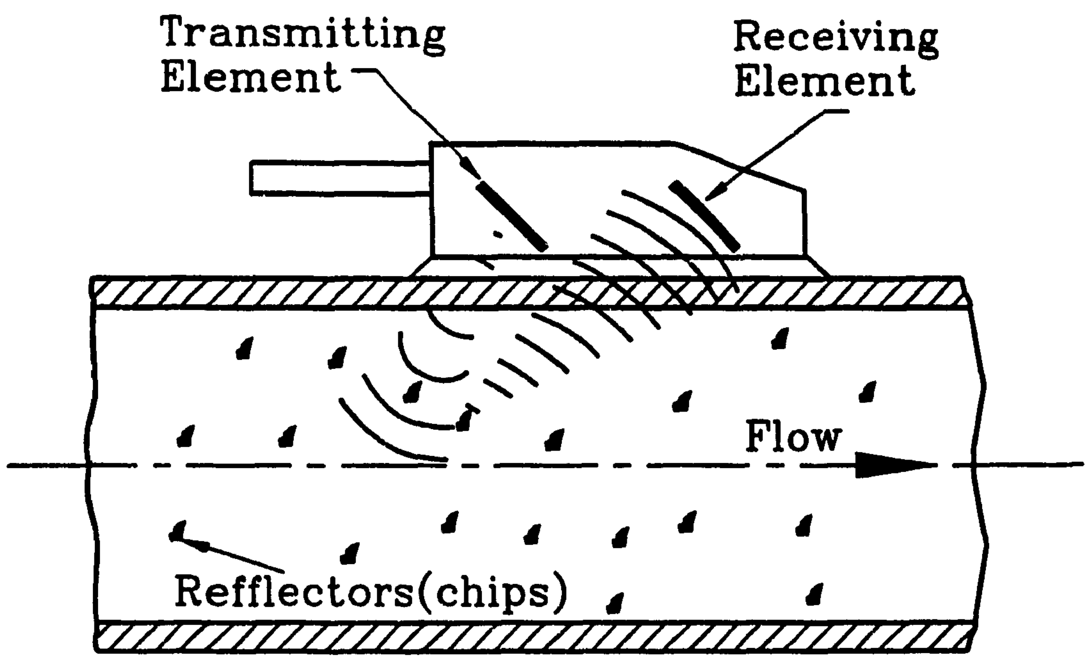


Figure 6.4: Ultra sonic Doppler flow sensor.

REFERENCES

1. Aaron, M Tenenbaum, Yedidyan Langsman and M.J. Augenstein, 1990, "Data Structures Using C", Prentice Hall.

2. Adeli, Hojjat, 1990, "Knowledge Engineering, Vol 1 and 2", McGraw Hill, New York.

3. Alty, J.L., and Johannsen, G., 1989, "Knowledge-based Dialogue for Dynamic Systems", Automatica, Vol. 25, No. 6, pp. 829-840.

4. Amitay, G., Malkin S., and Koren, Y., 1981, "Adaptive Control Optimization of Grinding", ASME Journal of Engineering for Industry, Vol. 103, NO.1, Feb, pp.102-111.

5. Astakhov, V.P., 1987, "The Symbolic Model for Operative Control in Deep Hole Drilling", Technology and Automation, Vol. 40, pp. 47-56. (In Russian).

6. Astakhov, V.P., 1990, "Modelling of Therm-hydraulic processes in the technological systems of the deep-hole machining" (in Russian). Metallorezushie stanky, Kiev, No18, p.70.

7. Astakhov, V.P., and Scorupco, A.M., 1982, Investigations of the chip-removal system in deep-hole drilling (in Russian). *Metallorezushie stanky,* Kiev, No10, p.58.

8. Astakhov V.P., and Ajrikjan A.L., 1983, Experimental setup for investigations of the hydraulic system of the ejector drills (in Russian). *USSR patent No 1006091*, GCNT USSR.

9. Astakhov V.P., Ajrikjan A.L., and Scorupco A.M., 1983, Method of the experimental investigations of the hydraulic system of the ejector drills (in Russian). *USSR patent No 1041232*, GCNT USSR.

10. Astrom, K.J., Anton, J.J., and Arzen K.E., 1986, "Expert Control", Automatica, Vol. 22, No. 3, pp 277-286.

11. Astakhov V.P., 1984, Fundamental evaluation of machine-tool cooling system for deep-hole drilling (in Russian). *Metallorezushie stanky*, Kiev, No12, p.63.

12. Astrom and Wittenmark, 1989, "Adaptive Control ", Addison-Wesley, Reading, Massachusetts.

13. Bendini, R., and Lisini, G., 1976, "Experiments on Adaptive Control of a Milling Machine", ASME Journal of Engineering for Industry, February,pp.239-245.

14. Bernhard Buse., 1987, "Ein neuartiges System zur automatischen Schwingungsbekampfung", tz fur Metallbearbeitung, 81 Jahrg, pp. 29-34.

15. Carpenter, D., 1980, "Adaptive Control", Proceedings of the Machine Tool Task Force Conference, Vol. 4, Oct, pp. 31-40.

16. Cawdery, D.O., 1977, "Design parameters of the Deep Hole Drilling Machine", Proceedings of the second International Conference on Deep Hole Drilling & Boring, 25-27 May, Burnel University.

17. Chandrashekhar, S, 1984, "An Analytical and Experimental Stochastic modelling of the Resultant Force System in BTA Deep-Hole Machining and its Influence on the Dynamics of the Machine Tool Workpiece System", Ph.D. thesis, Dept. of Mechanical Engineering, Concordia University, Montreal, Canada.

18. Chryssolouris, G., Guillot, M., and Domroese, M., 1988, "A Decision Making Approach to Machining Control", Transactions of ASME, journal of Engineering for Industry, Vol. 110, pp. 397-398.

19. Coney Y., and Simmers D., 1979, A study of fully developed, laminar axial flow and Taylor vortex flow by means of shear stress measurements. *Journal of Mechanical Engineering Science,* 21, Vol 1, p.13.

20. David, L., and Kumar, G., 1986, "Real-Time Expert System Development Techniques and Applications", IEEE, Paper No. CH2332-5/86/0000/0069.

21. Davis, Randall & Lenatt, Douglas, 1982, "Knowledge Based Systems and Artificial Intelligence", McGraw Hill, New York.

22. Dennis, J.B., 1988, "Data-flow Computation: A case study, Computer Architecture Concepts and Systems", North-Holland, New York.

23. Doraiswami, R., and Jiang, J., 1989, "Performance Monitoring in Expert Control Systems", Automatica, Vol. 25, No. 6, pp.799-811.

24. El-Khabeery, M.M., Saleh, S.M., and Ramadan, M.R., 1991, "Some Observations of Surface Integrity of Deep Drilling Holes", Wear, Vol. 142, No. 2, March, pp. 331-349.

25. Ernst, G.W., and Newell, 1969, A. GPS: A Case Study in Generality and Problem-Solving:, New York, Academic Pres.

26. Evers, D.C., Smith, D.M., and Staros, C.J., 1984, "Interfacing and Intelligent Decision-Maker to a Real-Time Control System", SPIE, Vol. 485, Applications of Artificial Intelligence, pp. 60-64.

27. Feigenbaum, E.A, 1980, "Knowledge Engineering: The applied side of Artificial Intelligence", Report STAN-CS-80-812, Department of Computer Science, Stanford University, USA.

28. Fink, P.,1977 "Economic Efficiency of Deep Hole Boring: Optimum Cutting Conditions", Proceedings of the second International Conference on Deep Hole Drilling & Boring, 25-27 May, Burnel University.

29. Field, J.S., 1982, "investigating the Frequency Response of Cutting Force Dynamometers by Transient Loading", International Journal of production Research, Vol. 20, No. 1, p. 57-64.

30. Fox, M., Lowenfeld S, and Kleinsoky P., 1983, "Techniques for Sensor Based Diagnosis", Carnegie Mellon University, Tech. Rept CMV-RJ-TR-83-7.

31. Galip Ulsoy, A., and Koren, Y., 1993, "Control of Machining Processes", Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 115, pp. 301-308.

32. Galip Ulsoy, A., Yoram Koren, and Fred Rasmussen, 1983, "Principal Developments in the Adaptive Control of Machine Tools", Transactions of the ASME, Journal of Dynamic Systems, measurement, and Control, Vol 105, pp. 107-111.

33. Garcia-Gardea, E., Burney, F.A., and Wu, S.M., 1979, "Determination of True Cutting Signal by Separation of Instrumentation Dynamics from measured Response", Trans. ASME, Journal of Engineering for Industry, Vol. 101, p. 264-271.

34. Gnatuk A.P., Astakhov V.P., and Chegodar A.A., 1989, Investigation of the change of the energy parameters of the coolant in the hydraulic system of the deep-hole drills (in Russian). *Izvestia Vuzov. Mashinistroenie*, Moscow, No12, p.113.

35. Greuner, B., 1977, "Deep Hole Boring and Finishing Techniques and Their Application To the Manufacture of Hydraulic Cylinders", Proceedings of the second International Conference on Deep Hole Drilling & Boring, 25-27 May, Burnel University, Germany.

36. Griffiths, B.J., 1982, "An Investigation into the Role of the Burnishing Pads in the Deep Hole Machining Process", Ph.D., Thesis, Dept. of Production Technology, Brunel University, UK.

37: Griffiths, B.J., 1977, "An Introduction to Deep Hole Drilling and Boring", Proceedings of the second International Conference on Deep Hole Drilling & Boring, 25-27 May, Burnel University, Germany.

38. Griffiths, B.J., 1986, "Surface Integrity of Holes Produced by the BTA Deep Drilling Process", International Conference on Deep hole Drilling and Boring, Brunel University, London, pp. 1-13.

39. Groover, M.P., 1980, "Automation Production System and Computer Aided manufacturing", Prentice Hall, Englewood Cliffs, N.J.

40. Groover, M.P., 1970, "A definition and Survey of Adaptive Control Machining", SME Paper No. MS70-561.

41. Grosse G., 1986, Technical guide to the thermal processes. Cambridge Univ. press.

42. Herath, J., Yamaguchi, Y., Saito, N., and Yuba, T., 1988, "Data flow Computing Models, Languages, and machines for Intelligence computations", IEEE Transactions on Software Engineering, Vol.14, No. 12, December.

43. Hodson, R.F., Kandel, A, 1991, "Real-Time Expert Systems Computer Architecture", CRC Press, Florida.

44. Hwang, K., Ghosh, R. Chowkwanyum, 1987, "Computer Architectures for Artificial Intelligence Processing", Computer, IEEE, Vol.20, No. 1, January.

45. Isermann, R. and K.H. Lachmann, 1985, "Parameter adaptive control with configuration aids and supervision functions", Automatica, Vol.21, pp 625-638.

46. Karl-Erik Arzen, 1989, "An Architecture for Expert System Based Feedback Control", Automatica, Vol. 25, No.6, pp.813-827.

47. Kegg, R.L., 1978, "Production Experience with Adaptive Controls", Proceedings of the 3rd NC Robot Automation Conference.

48. Knight, W.A., and Sadek, M.M., 1971, "The Correction for Dynamic Errors in Machine Tool Dynamometers", Annals of CIRP, Vol. 19, p.237-241

49. Koren, Y., 1983, "Computer Control of Manufacturing Systems", McGraw-Hill, New York.

50. Lattimer Wright, M., 1986, "Hexscon: A Hybrid Microcomputer-Based Expert System for Real-Time Control Applications", IEEE paper No. CH2332-5/86/0000/0049.

51. Landau, I.D., 1974, " A survey of Model Reference Adaptive Techniques-Theory and Application", Automatica, Vol. 10, pp. 353-379.

52. Ljubomir S. Lukic.,1987, "Prilog razvoju visokoproduktivnih obradnih sistema zp duboko busenje sa ACC upravlianjem", Produktivnost, Vol. 2, No. 3, pp 122-126 (in Serbo-Croatian).

53. Mathias, R.A., Boock, W., and Welch, A., 1980, "Adaptive Control: Monitoring and Control of Metal-cutting Process:, Proceedings of the Machine Tool Task Force Conference, Vol.4, Sect. 7.13, oct.

54. Neapolitan, R.E., 1986, "Forward Chaining Versus a graph approach as the inference engine in expert systems", Proc. Applications of Artificial Intelligence III, Spie Vol. 635, pp.62-69.

55. Newell, A., and Simon, H., 1972, "Human Problem solving", Englewood Cliffs, NJ, Prentice-Hall.

56. Newell, A., and H.A. Simon,1972, "Human Problem Solving", Prentice-Hall,

57. Nicholson, R., and Evans, B., 1979, " Considerations in the Selection, Application, Maintenance and Handling of Coolants for Deep Drilling", The proceedings of the Third international conference on Deep Hole Drilling and Boring, Brunel University, 10,11 May.

58. Nicholson, R., 1979, "A Comparison of Neat Oils and Watermix Fluids for Gundrilling", The proceedings of the Third International Conference on Deep hole Drilling and Boring, Brunel University, 10,11 May.

59. Osman, M.O.M., and V. Latinovic, 1975, "Trends in the Development of Deep-Hole Machining for Maximum Productivity", Pan-American Congress of Mechanical and Electrical Engineering and its Allied Branches, COPIMERA, Sao Paulo, Brazil, November.

60. Osman, M.O.M., and Sankar, T.S., 1975, "Profile Characterization of Manufactured Surfaces using Random Function Excursion Technique- Part II: Application", Trans. of ASME, J. of Engineering for Industry., Ser. B., p. 196-202.

61. Osman, M.O.M., and B., Greuner, 1974, " Development in BTA-System for Deep-Hole Boring", VD1-Tiefbohern in der Spananden Fertigung, Paper No. V11/1, Dusseldorf, Germany (In German).

62. Osman, M.O.M., and Latinovic, V.N., 1991, "On the Investigation of Flow Losses Due to Friction in Gundrilling", ASME Winter Annual Meeting, Atlanta, U.S.A.,

63. O'Neill, D.M., and Mullarkey, P.W., 1988, "A Knowledge-Based Approach to Real time Signal Monitoring",

64. Owen, J.V.,1993, "Feedback from the Cutting Edge", Manufacturing Engineering, January, Vol. 110, No.1, pp. 39-45

65. Parsaye, K., and Chignell, M., 1988, Expert Systems for Experts", John Wiley & Sons, Inc., New York.

66. Post, E.L., 1936, "Finite combinatory processes, formulation I", Journal of Symbolic Logic, Vol. 1, pp. 103-105.

67. Peter Streicher., 1975,"Tiefbohren der Metalle", Vogel-Verlag (in German).

68. Pospelov D., and Zagadskay N "Expert system for control of poorly defined process", (in Russian).

69. Pressman, R.S., and Williams, J.E., 1977, "Numerical Control and Computer Aided manufacturing", Wiley, New York.

70. Ramakrishna Rao, P.K., and Shunmugam, M.S., 1989, "Wear Studies in BTA Drilling", Wear, Vol 124, No. 1, pp-33-43.

71. Sakuma, K., Taguchi, K., and Katsuki, 1981, "Self Guiding Action of Deep-Hole Drilling Tools", Annals of CIRP, Vol. 30, No. 1, pp. 311-316.

72. Sakuma, K., Taguchi, K., and Katsuki, 1980, "Study on Deep Hole Drilling with Solid Boring Tool- The Burnishing Action of Guide Pads and Their Influence on Hole Accuracies", Bull. of JSME, , Vol. 23, No. 185, pp. 1921-1928.

73. Sakuma, K., Taguchi, K., and Katsuki, 1980, "Study on Deep-Hole Boring by a BTA System Solid Boring Tool- Behaviour of Tool and Its Effects on the Profile of the Machined Hole", Bull of JSPE, Vol. 14, No. 3, pp. 143-149.

74. Sankar, T.S., Osman, M.O.M., 1975, "Profile Characterization of Manufactured Surfaces Using Random Function Excursion Technique- Part I: Theory", Transactions of ASME, Journal of Engineering for Industry, Ser. B, p. 190-196.

75. Saunders, A.J., 1977, "The Introduction of Gun Drilling to Diesel Fuel Pump Element Barrels", Proceedings of the second International Conference on Deep Hole Drilling & Boring, 25-27 May, Burnel University, Germany.

76. SME, 1990, "Expert Systems", SME Blue Book Series, Produced by CASA/SME Technical Council, Michigan.

77. Sturenburg, H.O., 1977, "Deep Hole Drilling in Automotive Engine Production",Proceedings of the second International Conference on Deep Hole Drilling & Boring, 25-27 May, Burnel University,

78. Subramanya P.S, Osman, M.O.M., and Latinovic, V.N., 1990, "Expert Control of Turning Process", Proc. of the 5th International Conference on CAD/CAM Robotics and Factories of the Future, December 2-5, Norfolk, Virginia, pp.

79. Subramanya P.S., Latinovic, V.N., and Osman, M.O.M., 1991, "Adaptive Control of Machining Using Expert System", Proc. of the 6th International Conference on CAD/CAM, Robotics and Factories of the Future, August, London, England.

80. Subramanya P.S., 1989, "Computerized machinability Data Analysis for Nimonic 80A", M. Sc(Engg.) Thesis, Indian Institute of Technology, Madras, India.

81. Syrett, R.J., and Bult, H.J., 1979, "The Relative Performances of Cutting Fluids in Deep Drilling", The proceedings of the Third International Conference on Deep hole Drilling and Boring, Brunel University, 10,11 May.

82. Taylor G., 1963, Fluid friction between rotating cylinders. Proc. of Royal Soc.(London), Ser.A, No57, p.546.

83. Teltz, R and Elbestawi, M.A., 1993, "Hierarchical, Knowledge-Based Control in Turning", Transactions of ASME, Journal of Dynamic Systems, Measurement, and Control, March, Vol.115, pp.123-130.

84. Thurow, L., 1992, "Head-to-Head: The Coming Economic Battle Among Japan, Europe and America,", Morrow, New Youk, New York.

85. Tiberghien, J., 1984, " New Computer Architectures", Academic Press, London, 1984.

86. Tong, R.M, 1984, "A retrospective view of fuzzy control systems", Fuzzy Sets Syst.,Vol. 14, pp. 199-210.

87. Wick, C., 1977, "Automatic Adaptive Control of Machine Tools, Manufacturing Engineering, Sept, pp. 38-45.

88. Zwingmann, G., 1975, "Cutting Oils for Deep Hole Boring", International Conference on Deep Hole Drilling and Boring, Brunel university, 22-23 May.

APPENDIX 1

MACHINING RULES

#	PREMISE (IF)	CONCLUSION (THEN)
1	$(FO=1) \cup (FF=1) \cup (MT=1) \cup (MF=1) \cup (CF=1)$	INCR_V, INCR_S,TREND CONTROL
2	$(FO=1) \cup (FF=1) \cup (MT=1) \cup (MF=1) \cup (CF=2)$	INCR_V, INCR_S,TREND CONTROL
3	$(FO=1) \cup (FF=1) \cup (MT=1) \cup (MF=1) \cup (CF=3)$	INCR_S, SLOW
4	$(FO=1) \cup (FF=1) \cup (MT=1) \cup (MF=2) \cup (CF=1)$	INCR_V,INCR_S, TREND CONTROL
5	$(FO=1) \cup (FF=1) \cup (MT=1) \cup (MF=2) \cup (CF=2)$	INCR_S,INCR_V, GRADUAL
6	$(FO=1) \cup (FF=1) \cup (MT=1) \cup (MF=2) \cup (CF=3)$	INCR_Q, INCR_S, SLOW
7	$(FO=1) \cup (FF=1) \cup (MT=1) \cup (MF=3) \cup (CF=1)$	DECR_V, GRADUAL
8	$(FO=1) \cup (FF=1) \cup (MT=1) \cup (MF=3) \cup (CF=2)$	DECR_V, INCR_S, GRADUAL
9	$(FO=1) \cup (FF=1) \cup (MT=1) \cup (MF=3) \cup (CF=3)$	DECR_V, INCR_Q TO MAX.
10	$(FO=1) \cup (FF=1) \cup (MT=2) \cup (MF=1) \cup (CF=1)$	INCR_S SLOW, DECR_V GRADUAL

#	PREMISE (IF)	CONCLUSION (THEN)
11	$(FO=1)\cup(FF=1)\cup(MT=2)\cup(MF=1)\cup(CF=2)$	INCR_S, INCR_Q TOGETHER
12	$(FO=1)\cup(FF=1)\cup(MT=2)\cup(MF=1)\cup(CF=3)$	DECR_V, INCR_Q TO MAX
13	$(FO=1)\cup(FF=1)\cup(MT=2)\cup(MF=2)\cup(CF=1)$	INCR_S, GRADUAL
14	$(FO=1)\cup(FF=1)\cup(MT=2)\cup(MF=2)\cup(CF=2)$	INCR_S, GRADUAL
15	$(FO=1)\cup(FF=1)\cup(MT=2)\cup(MF=2)\cup(CF=3)$	DECR_S, INCR_Q
16	$(FO=1)\cup(FF=1)\cup(MT=2)\cup(MF=3)\cup(CF=1)$	DECR_V, INCR_S, GRADUAL
17	$(FO=1)\cup(FF=1)\cup(MT=2)\cup(MF=3)\cup(CF=2)$	DECR_V, INCR_Q
18	$(FO=1)\cup(FF=1)\cup(MT=2)\cup(MF=3)\cup(CF=3)$	DECR_V, INCR_Q
19	$(FO=1)\cup(FF=1)\cup(MT=3)\cup(MF=1)\cup(CF=1)$	DECR_V SLOW, INCR_S GRADUAL
20	$(FO=1)\cup(FF=1)\cup(MT=3)\cup(MF=1)\cup(CF=2)$	DECR_V, INCR_S, GRADUAL

#	PREMISE (IF)	CONCLUSION (THEN)
21	$(FO=1)\cup(FF=1)\cup(MT=3)\cup(MF=1)\cup(CF=3)$	DECR_V, INCR_S, INCR_Q, GRADUAL
22	$(FO=1)\cup(FF=1)\cup(MT=3)\cup(MF=2)\cup(CF=1)$	DECR_V, INCR_S, GRADUAL
23	$(FO=1)\cup(FF=1)\cup(MT=3)\cup(MF=2)\cup(CF=2)$	DECR_V, GRADUAL
24	$(FO=1)\cup(FF=1)\cup(MT=3)\cup(MF=2)\cup(CF=3)$	DECR_V, GRADUAL
25	$(FO=1)\cup(FF=1)\cup(MT=3)\cup(MF=3)\cup(CF=1)$	DECR_V, INCR_S, GRADUAL
26	$(FO=1)\cup(FF=1)\cup(MT=3)\cup(MF=3)\cup(CF=2)$	DECR_V, GRADUAL
27	$(FO=1)\cup(FF=1)\cup(MT=3)\cup(MF=3)\cup(CF=3)$	DECR_V,SLOW, INCR_S, GRADUAL & INCR_Q TO MAX.
28	$(FO=1)\cup(FF=2)\cup(MT=1)\cup(MF=1)\cup(CF=1)$	DECR_V, DECR_S, SLOW
29	$(FO=1)\cup(FF=2)\cup(MT=1)\cup(MF=1)\cup(CF=2)$	INCR_V, SLOW, INCR_S GRADUAL
30	$(FO=1)\cup(FF=2)\cup(MT=1)\cup(MF=1)\cup(CF=3)$	INCR_Q

#	PREMISE (IF)	CONCLUSION (THEN)
31	(FO=1)∪(FF=2)∪(MT=1)∪(MF=2)∪(CF=1)	INCR_V, INCR_S, SLOW
32	(FO=1)∪(FF=2)∪(MT=1)∪(MF=2)∪(CF=2)	INCR_V, INCR_S, SLOW
33	(FO=1)∪(FF=2)∪(MT=1)∪(MF=2)∪(CF=3)	INCR_V
34	(FO=1)∪(FF=2)∪(MT=1)∪(MF=3)∪(CF=1)	DECR_V, INCR_S, SLOW
35	(FO=1)∪(FF=2)∪(MT=1)∪(MF=3)∪(CF=2)	DECR_V, INCR_S, GRADUAL
36	(FO=1)∪(FF=2)∪(MT=1)∪(MF=3)∪(CF=3)	DECR_V, INCR_Q
37	(FO=1)∪(FF=2)∪(MT=2)∪(MF=1)∪(CF=1)	INCR_S, SLOW
38	(FO=1)∪(FF=2)∪(MT=2)∪(MF=1)∪(CF=2)	INCR_V, INCR_S, GRADUAL
39	(FO=1)∪(FF=2)∪(MT=2)∪(MF=1)∪(CF=3)	INCR_Q
40	(FO=1)∪(FF=2)∪(MT=2)∪(MF=2)∪(CF=1)	INCR_S, GRADUAL

#	PREMISE (IF)	CONCLUSION (THEN)
41	$(FO=1)\cup(FF=2)\cup(MT=2)\cup(MF=2)\cup(CF=2)$	INCR_S, GRADUAL
42	$(FO=1)\cup(FF=2)\cup(MT=2)\cup(MF=2)\cup(CF=3)$	INCR_Q
43	$(FO=1)\cup(FF=2)\cup(MT=2)\cup(MF=3)\cup(CF=1)$	DECR_V, INCR_S, GRADUAL
44	$(FO=1)\cup(FF=2)\cup(MT=2)\cup(MF=3)\cup(CF=2)$	DECR_V, INCR_S, SUDDEN
45	$(FO=1)\cup(FF=2)\cup(MT=2)\cup(MF=3)\cup(CF=3)$	DECR_V, SLOW
46	$(FO=1)\cup(FF=2)\cup(MT=3)\cup(MF=1)\cup(CF=1)$	DECR_V SLOW, INCR_S GRADUAL
47	$(FO=1)\cup(FF=2)\cup(MT=3)\cup(MF=1)\cup(CF=2)$	DECR_V SLOW, INCR_S GRADUAL
48	$(FO=1)\cup(FF=2)\cup(MT=3)\cup(MF=1)\cup(CF=3)$	DECR_V SUDDEN
49	$(FO=1)\cup(FF=2)\cup(MT=3)\cup(MF=2)\cup(CF=1)$	DECR_V, INCR_S, GRADUAL
50	$(FO=1)\cup(FF=2)\cup(MT=3)\cup(MF=2)\cup(CF=2)$	DECR_V, INCR_S, GRADUAL

#	PREMISE (IF)	CONCLUSION (THEN)
51	(FO=1)∪(FF=2)∪(MT=3)∪(MF=2)∪(CF=3)	DECR_V, GRADUAL
52	(FO=1)∪(FF=2)∪(MT=3)∪(MF=3)∪(CF=1)	DECR_V SUDDEN, DECR_S GRADUAL
53	(FO=1)∪(FF=2)∪(MT=3)∪(MF=3)∪(CF=2)	DECR_V SUDDEN
54	(FO=1)∪(FF=2)∪(MT=3)∪(MF=3)∪(CF=3)	DECR_V, INCR Q TO MAX
55	(FO=1)∪(FF=3)∪(MT=1)∪(MF=1)∪(CF=1)	DECR_V , INCR_S, SLOW
56	(FO=1)∪(FF=3)∪(MT=1)∪(MF=1)∪(CF=2)	DECR_V, INCR_S, SLOW
57	(FO=1)∪(FF=3)∪(MT=1)∪(MF=1)∪(CF=3)	DECR_V, INCR_S, INCR_Q,GRADUAL
58	(FO=1)∪(FF=3)∪(MT=1)∪(MF=2)∪(CF=1)	DECR_V, INCR_S, SLOW
59	(FO=1)∪(FF=3)∪(MT=1)∪(MF=2)∪(CF=2)	DECR_V GRADUAL
60	(FO=1)∪(FF=3)∪(MT=1)∪(MF=2)∪(CF=3)	DECR_V TO MIN, INCR_S GRADUAL

#	PREMISE (IF)	CONCLUSION (THEN)
61	(FO=1)∪(FF=3)∪(MT=1)∪(MF=3)∪(CF=1)	DECR_V, INCR_S, SLOW
62	(FO=1)∪(FF=3)∪(MT=1)∪(MF=3)∪(CF=2)	DECR_V, SLOW
63	(FO=1)∪(FF=3)∪(MT=1)∪(MF=3)∪(CF=3)	DECR_V TO MIN, INCR_Q
64	(FO=1)∪(FF=3)∪(MT=2)∪(MF=1)∪(CF=1)	DECR_V, SLOW
65	(FO=1)∪(FF=3)∪(MT=2)∪(MF=1)∪(CF=2)	DECR_V, INCR_S
66	(FO=1)∪(FF=3)∪(MT=2)∪(MF=1)∪(CF=3)	DECR_V, INCR_S, INCR_Q, GRADUAL
67	(FO=1)∪(FF=3)∪(MT=2)∪(MF=2)∪(CF=1)	DECR_V, INCR_S, SLOW
68	(FO=1)∪(FF=3)∪(MT=2)∪(MF=2)∪(CF=2)	DECR_V, GRADUAL
69	(FO=1)∪(FF=3)∪(MT=2)∪(MF=2)∪(CF=3)	DECR_V, INCR_S, INCR_Q, GRADUAL
70	(FO=1)∪(FF=3)∪(MT=2)∪(MF=3)∪(CF=1)	DECR_V SLOW, INCR_Q

#	PREMISE (IF)	CONCLUSION (THEN)
71	(FO=1)∪(FF=3)∪(MT=2)∪(MF=3)∪(CF=2)	DECR_V, DECR_S, GRADUAL
72	(FO=1)∪(FF=3)∪(MT=2)∪(MF=3)∪(CF=3)	DECR_V,INCR_Q, GRADUAL
73	(FO=1)∪(FF=3)∪(MT=3)∪(MF=1)∪(CF=1)	DECR_V,DECR_S, GRADUAL
74	(FO=1)∪(FF=3)∪(MT=3)∪(MF=1)∪(CF=2)	DECR_V, INCR_S, GRADUAL
75	(FO=1)∪(FF=3)∪(MT=3)∪(MF=1)∪(CF=3)	DECR_V,INCR_S, INCR_Q, SLOW
76	(FO=1)∪(FF=3)∪(MT=3)∪(MF=2)∪(CF=1)	DECR_V, INCR_S, GRADUAL
77	(FO=1)∪(FF=3)∪(MT=3)∪(MF=2)∪(CF=2)	DECR_V, GRADUAL
78	(FO=1)∪(FF=3)∪(MT=3)∪(MF=2)∪(CF=3)	DECR_V, INCR_Q, GRADUAL
79	(FO=1)∪(FF=3)∪(MT=3)∪(MF=3)∪(CF=1)	DECR_V TO MIN, SUDDEN
80	(FO=1)∪(FF=3)∪(MT=3)∪(MF=3)∪(CF=2)	DECR_V TO MIN, SUDDEN

#	PREMISE (IF)	CONCLUSION (THEN)
81	(FO=1)∪(FF=3)∪(MT=3)∪(MF=3)∪(CF=3)	DECR_V TO MIN, DECR_S, INCR_Q
82	(FO=2)∪(FF=1)∪(MT=1)∪(MF=1)∪(CF=1)	DECR_V, DECR_S, TO MIN
83	(FO=2)∪(FF=1)∪(MT=1)∪(MF=1)∪(CF=2)	INCR_V, INCR_S, SLOW
84	(FO=2)∪(FF=1)∪(MT=1)∪(MF=1)∪(CF=3)	INCR_S TO MAX, GRADUAL
85	(FO=2)∪(FF=1)∪(MT=1)∪(MF=2)∪(CF=1)	DECR_V TO MIN
86	(FO=2)∪(FF=1)∪(MT=1)∪(MF=2)∪(CF=2)	INCR_V AND INCR_Q
87	(FO=2)∪(FF=1)∪(MT=1)∪(MF=2)∪(CF=3)	INCR_S GRADUAL, INCR_Q
88	(FO=2)∪(FF=1)∪(MT=1)∪(MF=3)∪(CF=1)	INCR_S GRADUAL, INCR_Q
89	(FO=2)∪(FF=1)∪(MT=1)∪(MF=3)∪(CF=2)	DECR_V, INCR_Q, INCR_S, SLOW
90	(FO=2)∪(FF=1)∪(MT=1)∪(MF=3)∪(CF=3)	DECR_V GRADUAL, INCR_S SLOW

#	PREMISE (IF)	CONCLUSION (THEN)
91	$(FO=2)\cup(FF=1)\cup(MT=2)\cup(MF=1)\cup(CF=1)$	STOP_S, INCR_Q, IDLE 3SEC, START_S
92	$(FO=2)\cup(FF=1)\cup(MT=2)\cup(MF=1)\cup(CF=2)$	TREND CONTROL
93	$(FO=2)\cup(FF=1)\cup(MT=2)\cup(MF=1)\cup(CF=3)$	INCR_S TO MAX, DECR_V, GRADUAL
94	$(FO=2)\cup(FF=1)\cup(MT=2)\cup(MF=2)\cup(CF=1)$	STOP_S, INCR_Q, IDLE 3SEC, START_S
95	$(FO=2)\cup(FF=1)\cup(MT=2)\cup(MF=2)\cup(CF=2)$	DO NOTHING
96	$(FO=2)\cup(FF=1)\cup(MT=2)\cup(MF=2)\cup(CF=3)$	INCR_S, GRADUAL
97	$(FO=2)\cup(FF=1)\cup(MT=2)\cup(MF=3)\cup(CF=1)$	STOP_S, INCR_Q, IDLE, START_S
98	$(FO=2)\cup(FF=1)\cup(MT=2)\cup(MF=3)\cup(CF=2)$	DECR_V, INCR_Q, GRADUAL
99	$(FO=2)\cup(FF=1)\cup(MT=2)\cup(MF=3)\cup(CF=3)$	DECR_V, DECR_Q, SLOW
100	$(FO=2)\cup(FF=1)\cup(MT=3)\cup(MF=1)\cup(CF=1)$	STOP_S, INCR_Q, IDLE, START_S

#	PREMISE (IF)	CONCLUSION (THEN)
101	$(FO=2)\cup(FF=1)\cup(MT=3)\cup(MF=1)\cup(CF=2)$	DECR_V, SUDDEN
102	$(FO=2)\cup(FF=1)\cup(MT=3)\cup(MF=1)\cup(CF=3)$	DECR_V SUDDEN, INCR_Q
103	$(FO=2)\cup(FF=1)\cup(MT=3)\cup(MF=2)\cup(CF=1)$	STOP_S, IDLE, INCR_Q, START_S
104	$(FO=2)\cup(FF=1)\cup(MT=3)\cup(MF=2)\cup(CF=2)$	DECR_V, SUDDEN
105	$(FO=2)\cup(FF=1)\cup(MT=3)\cup(MF=2)\cup(CF=3)$	DECR_V SUDDEN, INCR_Q
106	$(FO=2)\cup(FF=1)\cup(MT=3)\cup(MF=3)\cup(CF=1)$	STOP_S, IDLE, INCR_Q, START_S
107	$(FO=2)\cup(FF=1)\cup(MT=3)\cup(MF=3)\cup(CF=2)$	DECR_V, DECR_S, SLOW
108	$(FO=2)\cup(FF=1)\cup(MT=3)\cup(MF=3)\cup(CF=3)$	DECR_V SUDDEN, INCR_Q
109	$(FO=2)\cup(FF=2)\cup(MT=1)\cup(MF=1)\cup(CF=1)$	STOP_S, IDLE, INCR_Q, START_S
110	$(FO=2)\cup(FF=2)\cup(MT=1)\cup(MF=1)\cup(CF=2)$	INCR_V, SLOW

#	PREMISE (IF)	CONCLUSION (THEN)
111	(FO=2)∪(FF=2)∪(MT=1)∪(MF=1)∪(CF=3)	INCR_Q
112	(FO=2)∪(FF=2)∪(MT=1)∪(MF=2)∪(CF=1)	STOP_S, IDLE, INCR_Q,START_S
113	(FO=2)∪(FF=2)∪(MT=1)∪(MF=2)∪(CF=2)	INCR_V, GRADUAL
114	(FO=2)∪(FF=2)∪(MT=1)∪(MF=2)∪(CF=3)	IN CR_V, DECR_S, GRADUAL, INCR_Q
115	(FO=2)∪(FF=2)∪(MT=1)∪(MF=3)∪(CF=1)	STOP_S, IDLE, INCR_Q, START_S
116	(FO=2)∪(FF=2)∪(MT=1)∪(MF=3)∪(CF=2)	INCR_S GRADUAL, DECR_V, SLOW
117	(FO=2)∪(FF=2)∪(MT=1)∪(MF=3)∪(CF=3)	DECR_V SUDDEN, INCR_Q
118	(FO=2)∪(FF=2)∪(MT=2)∪(MF=1)∪(CF=1)	STOP_S, IDLE, INCR_Q, START_S
119	(FO=2)∪(FF=2)∪(MT=2)∪(MF=1)∪(CF=2)	INCR_V, GRADUAL
120	(FO=2)∪(FF=2)∪(MT=2)∪(MF=1)∪(CF=3)	DECR_V, OR INCR_S, INCR_Q

#	PREMISE (IF)	CONCLUSION (THEN)
121	$(FO=2)\cup(FF=2)\cup(MT=2)\cup(MF=2)\cup(CF=1)$	FLUSH CHIPS
122	$(FO=2)\cup(FF=2)\cup(MT=2)\cup(MF=2)\cup(CF=2)$	DO NOTHING
123	$(FO=2)\cup(FF=2)\cup(MT=2)\cup(MF=2)\cup(CF=3)$	INCR_Q
124	$(FO=2)\cup(FF=2)\cup(MT=2)\cup(MF=3)\cup(CF=1)$	FLUSH CHIPS
125	$(FO=2)\cup(FF=2)\cup(MT=2)\cup(MF=3)\cup(CF=2)$	DECR_V GRADUAL
126	$(FO=2)\cup(FF=2)\cup(MT=2)\cup(MF=3)\cup(CF=3)$	DECR_V GRADUAL, INCR_Q
127	$(FO=2)\cup(FF=2)\cup(MT=3)\cup(MF=1)\cup(CF=1)$	FLUSH CHIPS
128	$(FO=2)\cup(FF=2)\cup(MT=3)\cup(MF=1)\cup(CF=2)$	DECR_V GRADUAL
129	$(FO=2)\cup(FF=2)\cup(MT=3)\cup(MF=1)\cup(CF=3)$	DECR_V GRADUAL, INCR_Q
130	$(FO=2)\cup(FF=2)\cup(MT=3)\cup(MF=2)\cup(CF=1)$	FLUSH CHIPS

#	PREMISE (IF)	CONCLUSION (THEN)
131	$(FO=2)\cup(FF=2)\cup(MT=3)\cup(MF=2)\cup(CF=2)$	DECR_V GRADUAL
132	$(FO=2)\cup(FF=2)\cup(MT=3)\cup(MF=2)\cup(CF=3)$	DECR_V, DECR_S, GRADUAL
133	$(FO=2)\cup(FF=2)\cup(MT=3)\cup(MF=3)\cup(CF=1)$	FLUSH CHIPS
134	$(FO=2)\cup(FF=2)\cup(MT=3)\cup(MF=3)\cup(CF=2)$	DECR_V, DECR_S, TO MIN
135	$(FO=2)\cup(FF=2)\cup(MT=3)\cup(MF=3)\cup(CF=3)$	DECR_V, DECR_S, SLOW
136	$(FO=2)\cup(FF=3)\cup(MT=1)\cup(MF=1)\cup(CF=1)$	FLUSH CHIPS
137	$(FO=2)\cup(FF=3)\cup(MT=1)\cup(MF=1)\cup(CF=2)$	DECR_V TO MIN
138	$(FO=2)\cup(FF=3)\cup(MT=1)\cup(MF=1)\cup(CF=3)$	DECR_V TO MIN, DECR_S
139	$(FO=2)\cup(FF=3)\cup(MT=1)\cup(MF=2)\cup(CF=1)$	FLUSH CHIPS
140	$(FO=2)\cup(FF=3)\cup(MT=1)\cup(MF=2)\cup(CF=2)$	DECR_V TO MIN

#	PREMISE (IF)	CONCLUSION (THEN)
141	(FO=2)∪(FF=3)∪(MT=1)∪(MF=2)∪(CF=3)	DECR_V TO MIN, DECR_S GRADUAL
142	(FO=2)∪(FF=3)∪(MT=1)∪(MF=3)∪(CF=1)	FLUSH CHIPS
143	(FO=2)∪(FF=3)∪(MT=1)∪(MF=3)∪(CF=2)	DECR_V GRADUAL
144	(FO=2)∪(FF=3)∪(MT=1)∪(MF=3)∪(CF=3)	DECR_V GRADUAL
145	(FO=2)∪(FF=3)∪(MT=2)∪(MF=1)∪(CF=1)	FLUSH CHIPS
146	(FO=2)∪(FF=3)∪(MT=2)∪(MF=1)∪(CF=2)	DECR_V GRADUAL
147	(FO=2)∪(FF=3)∪(MT=2)∪(MF=1)∪(CF=3)	DECR_V GRADUAL
148	(FO=2)∪(FF=3)∪(MT=2)∪(MF=2)∪(CF=1)	FLUSH CHIPS
149	(FO=2)∪(FF=3)∪(MT=2)∪(MF=2)∪(CF=2)	DECR_V GRADUAL
150	(FO=2)∪(FF=3)∪(MT=2)∪(MF=2)∪(CF=3)	DECR_V GRADUAL

#	PREMISE (IF)	CONCLUSION (THEN)
151	$(FO=2)\cup(FF=3)\cup(MT=2)\cup(MF=3)\cup(CF=1)$	FLUSH CHIPS
152	$(FO=2)\cup(FF=3)\cup(MT=2)\cup(MF=3)\cup(CF=2)$	DECR_V, DECR_S, GRADUAL
153	$(FO=2)\cup(FF=3)\cup(MT=2)\cup(MF=3)\cup(CF=3)$	DECR_V, SUDDEN, DECR_S GRADUAL
154	$(FO=2)\cup(FF=3)\cup(MT=3)\cup(MF=1)\cup(CF=1)$	FLUSH CHIPS
155	$(FO=2)\cup(FF=3)\cup(MT=3)\cup(MF=1)\cup(CF=2)$	DECR_V, SLOW
156	$(FO=2)\cup(FF=3)\cup(MT=3)\cup(MF=1)\cup(CF=3)$	DECR_V, SLOW, INCR_Q
157	$(FO=2)\cup(FF=3)\cup(MT=3)\cup(MF=2)\cup(CF=1)$	FLUSH CHIPS
158	$(FO=2)\cup(FF=3)\cup(MT=3)\cup(MF=2)\cup(CF=2)$	DECR_V SLOW
159	$(FO=2)\cup(FF=3)\cup(MT=3)\cup(MF=2)\cup(CF=3)$	DECR_V, DECR_S, GRADUAL
160	$(FO=2)\cup(FF=3)\cup(MT=3)\cup(MF=3)\cup(CF=1)$	FLUSH CHIPS

#	PREMISE (IF)	CONCLUSION (THEN)
161	$(FO=2)\cup(FF=3)\cup(MT=3)\cup(MF=3)\cup(CF=2)$	DECR_V SUDDEN
162	$(FO=2)\cup(FF=3)\cup(MT=3)\cup(MF=3)\cup(CF=3)$	DECR_V SUDDEN, DECR_S GRADUAL
163	$(FO=3)\cup(FF=1)\cup(MT=1)\cup(MF=1)\cup(CF=1)$	EMERGENCY STOP
164	$(FO=3)\cup(FF=1)\cup(MT=1)\cup(MF=1)\cup(CF=2)$	DECR_S GRADUAL, INCR_V SLOW
165	$(FO=3)\cup(FF=1)\cup(MT=1)\cup(MF=1)\cup(CF=3)$	DECR_V, GRADUAL
166	$(FO=3)\cup(FF=1)\cup(MT=1)\cup(MF=2)\cup(CF=1)$	EMERGENCY STOP
167	$(FO=3)\cup(FF=1)\cup(MT=1)\cup(MF=2)\cup(CF=2)$	DECR_S GRADUAL, INCR_V SLOW
168	$(FO=3)\cup(FF=1)\cup(MT=1)\cup(MF=2)\cup(CF=3)$	DECR_S GRADUAL
169	$(FO=3)\cup(FF=1)\cup(MT=1)\cup(MF=3)\cup(CF=1)$	EMERGENCY STOP
170	$(FO=3)\cup(FF=1)\cup(MT=1)\cup(MF=3)\cup(CF=2)$	DECR_S GRADUAL, INCR_V SLOW

#	PREMISE (IF)	CONCLUSION (THEN)
171	$(FO=3)\cup(FF=1)\cup(MT=1)\cup(MF=3)\cup(CF=3)$	DECR_S GRADUAL
172	$(FO=3)\cup(FF=1)\cup(MT=2)\cup(MF=1)\cup(CF=1)$	EMERGENCY STOP
173	$(FO=3)\cup(FF=1)\cup(MT=2)\cup(MF=1)\cup(CF=2)$	DECR_ SLOW, INCR_V GRADUAL
174	$(FO=3)\cup(FF=1)\cup(MT=2)\cup(MF=1)\cup(CF=3)$	DECR_ S SLOW, INCR_V GRADUAL, INCR_Q
175	$(FO=3)\cup(FF=1)\cup(MT=2)\cup(MF=2)\cup(CF=1)$	EMERGENCY STOP
176	$(FO=3)\cup(FF=1)\cup(MT=2)\cup(MF=2)\cup(CF=2)$	DECR_ SLOW, INCR_V GRADUA
177	$(FO=3)\cup(FF=1)\cup(MT=2)\cup(MF=2)\cup(CF=3)$	DECR_S, DECR_V, GRADUAL, INCR_Q
178	$(FO=3)\cup(FF=1)\cup(MT=2)\cup(MF=3)\cup(CF=1)$	EMERGENCY STOP
179	$(FO=3)\cup(FF=1)\cup(MT=2)\cup(MF=3)\cup(CF=2)$	DECR_V, DECR_S, SLOW
180	$(FO=3)\cup(FF=1)\cup(MT=2)\cup(MF=3)\cup(CF=3)$	DECR_V, DECR-S, INCR_Q

#	PREMISE (IF)	CONCLUSION (THEN)
181	$(FO=3)\cup(FF=1)\cup(MT=3)\cup(MF=1)\cup(CF=1)$	EMERGENCY STOP
182	$(FO=3)\cup(FF=1)\cup(MT=3)\cup(MF=1)\cup(CF=2)$	DECR_V, DECR_S, SLOW
183	$(FO=3)\cup(FF=1)\cup(MT=3)\cup(MF=1)\cup(CF=3)$	DECR_V, DECR_S, SLOW, INCR_Q
184	$(FO=3)\cup(FF=1)\cup(MT=3)\cup(MF=2)\cup(CF=1)$	EMERGENCY STOP
185	$(FO=3)\cup(FF=1)\cup(MT=3)\cup(MF=2)\cup(CF=2)$	DECR_V, SLOW, DECR_S, GRADUAL
186	$(FO=3)\cup(FF=1)\cup(MT=3)\cup(MF=2)\cup(CF=3)$	DECR_V, DECR_S, GRADUAL, INCR_Q
187	$(FO=3)\cup(FF=1)\cup(MT=3)\cup(MF=3)\cup(CF=1)$	EMERGENCY STOP
188	$(FO=3)\cup(FF=1)\cup(MT=3)\cup(MF=3)\cup(CF=2)$	DECR_V, DECR_S, SLOW
189	$(FO=3)\cup(FF=1)\cup(MT=3)\cup(MF=3)\cup(CF=3)$	DECR_V, DECR_S, SLOW, INCR_Q
190	$(FO=3)\cup(FF=2)\cup(MT=1)\cup(MF=1)\cup(CF=1)$	EMERGENCY STOP

#	PREMISE (IF)	CONCLUSION (THEN)
191	(FO=3)∪(FF=2)∪(MT=1)∪(MF=1)∪(CF=2)	DECR_S, SLOW, INCR_V GRADUAL
192	(FO=3)∪(FF=2)∪(MT=1)∪(MF=1)∪(CF=3)	DECR_S, INCR_Q
193	(FO=3)∪(FF=2)∪(MT=1)∪(MF=2)∪(CF=1)	EMERGENCY STOP
194	(FO=3)∪(FF=2)∪(MT=1)∪(MF=2)∪(CF=2)	DECR_S, INCR_V, SLOW
195	(FO=3)∪(FF=2)∪(MT=1)∪(MF=2)∪(CF=3)	DECR_S, INCR_V, SLOW
196	(FO=3)∪(FF=2)∪(MT=1)∪(MF=3)∪(CF=1)	EMERGENCY STOP
197	(FO=3)∪(FF=2)∪(MT=1)∪(MF=3)∪(CF=2)	DECR_V SLOW
198	(FO=3)∪(FF=2)∪(MT=1)∪(MF=3)∪(CF=3)	DECR_V, DECR_S, SLOW
199	(FO=3)∪(FF=2)∪(MT=2)∪(MF=1)∪(CF=1)	EMERGENCY STOP
200	(FO=3)∪(FF=2)∪(MT=2)∪(MF=1)∪(CF=2)	DECR_V, DECR_S, SLOW

#	PREMISE (IF)	CONCLUSION (THEN)
201	(FO=3)∪(FF=2)∪(MT=2)∪(MF=1)∪(CF=3)	DECR_V, SLOW, INCR_Q
202	(FO=3)∪(FF=2)∪(MT=2)∪(MF=2)∪(CF=1)	EMERGENCY STOP
203	(FO=3)∪(FF=2)∪(MT=2)∪(MF=2)∪(CF=2)	DECR_S, SLOW, INCR_V GRADUAL
204	(FO=3)∪(FF=2)∪(MT=2)∪(MF=2)∪(CF=3)	DECR_S, INCR_Q
205	(FO=3)∪(FF=2)∪(MT=2)∪(MF=3)∪(CF=1)	EMERGENCY STOP
206	(FO=3)∪(FF=2)∪(MT=2)∪(MF=3)∪(CF=2)	DECR_S, SLOW
207	(FO=3)∪(FF=2)∪(MT=2)∪(MF=3)∪(CF=3)	DECR_S SLOW, INCR_Q
208	(FO=3)∪(FF=2)∪(MT=3)∪(MF=1)∪(CF=1)	EMERGENCY STOP
209	(FO=3)∪(FF=2)∪(MT=3)∪(MF=1)∪(CF=2)	DECR_V, DECR_S, GRADUAL
210	(FO=3)∪(FF=2)∪(MT=3)∪(MF=1)∪(CF=3)	DECR_V, DECR_S, SLOW, INCR_Q

#	PREMISE (IF)	CONCLUSION (THEN)
211	$(FO=3)\cup(FF=2)\cup(MT=3)\cup(MF=2)\cup(CF=1)$	EMERGENCY STOP
212	$(FO=3)\cup(FF=2)\cup(MT=3)\cup(MF=2)\cup(CF=2)$	DECR_V, DECR_S, GRADUAL
213	$(FO=3)\cup(FF=2)\cup(MT=3)\cup(MF=2)\cup(CF=3)$	DECR_S TO MIN, DECR_V SLOW, INCR_Q
214	$(FO=3)\cup(FF=2)\cup(MT=3)\cup(MF=3)\cup(CF=1)$	EMERGENCY STOP
215	$(FO=3)\cup(FF=2)\cup(MT=3)\cup(MF=3)\cup(CF=2)$	DECR_V, DECR_S, SLOW
216	$(FO=3)\cup(FF=2)\cup(MT=3)\cup(MF=3)\cup(CF=3)$	DECR_S TO MIN, DECR_V SLOW, INCR_Q
217	$(FO=3)\cup(FF=3)\cup(MT=1)\cup(MF=1)\cup(CF=1)$	EMERGENCY STOP
218	$(FO=3)\cup(FF=3)\cup(MT=1)\cup(MF=1)\cup(CF=2)$	DECR_S SLOW, INCR_V, SLOW
219	$(FO=3)\cup(FF=3)\cup(MT=1)\cup(MF=1)\cup(CF=3)$	DECR_V, DECR_S, SLOW, INCR_Q
220	$(FO=3)\cup(FF=3)\cup(MT=1)\cup(MF=2)\cup(CF=1)$	EMERGENCY STOP

#	PREMISE (IF)	CONCLUSION (THEN)
221	(FO=3)∪(FF=3)∪(MT=1)∪(MF=2)∪(CF=2)	DECR_S, SLOW, INCR_V GRADUAL
222	(FO=3)∪(FF=3)∪(MT=1)∪(MF=2)∪(CF=3)	DECR_S, SLOW, INCR_Q
223	(FO=3)∪(FF=3)∪(MT=1)∪(MF=3)∪(CF=1)	EMERGENCY STOP
224	(FO=3)∪(FF=3)∪(MT=1)∪(MF=3)∪(CF=2)	DECR_S, SLOW, DECR_V GRADUAL
225	(FO=3)∪(FF=3)∪(MT=1)∪(MF=3)∪(CF=3)	DECR_S, DECR_V SLOW, INCR_Q
226	(FO=3)∪(FF=3)∪(MT=2)∪(MF=1)∪(CF=1)	EMERGENCY STOP
227	(FO=3)∪(FF=3)∪(MT=2)∪(MF=1)∪(CF=2)	DECR_S, SLOW
228	(FO=3)∪(FF=3)∪(MT=2)∪(MF=1)∪(CF=3)	DECR_V, INCR_Q
229	(FO=3)∪(FF=3)∪(MT=2)∪(MF=2)∪(CF=1)	EMERGENCY STOP
230	(FO=3)∪(FF=3)∪(MT=2)∪(MF=2)∪(CF=2)	DECR_S SLOW

#	PREMISE (IF)	CONCLUSION (THEN)
231	$(FO=3)\cup(FF=3)\cup(MT=2)\cup(MF=2)\cup(CF=3)$	DECR_V SLOW, INCR_Q
232	$(FO=3)\cup(FF=3)\cup(MT=2)\cup(MF=3)\cup(CF=1)$	EMERGENCY STOP
233	$(FO=3)\cup(FF=3)\cup(MT=2)\cup(MF=3)\cup(CF=2)$	DECR_V, DECR_S GRADUAL
234	$(FO=3)\cup(FF=3)\cup(MT=2)\cup(MF=3)\cup(CF=3)$	DECR_V, DECR_S GRADUAL, INCR_Q
235	$(FO=3)\cup(FF=3)\cup(MT=3)\cup(MF=1)\cup(CF=1)$	EMERGENCY STOP
236	$(FO=3)\cup(FF=3)\cup(MT=3)\cup(MF=1)\cup(CF=2)$	DECR_S SLOW, DECR_V GRADUAL
237	$(FO=3)\cup(FF=3)\cup(MT=3)\cup(MF=1)\cup(CF=3)$	DECR_S, DECR_V, SLOW
238	$(FO=3)\cup(FF=3)\cup(MT=3)\cup(MF=2)\cup(CF=1)$	EMERGENCY STOP
239	$(FO=3)\cup(FF=3)\cup(MT=3)\cup(MF=2)\cup(CF=2)$	DECR_V, DECR_S, SLOW
240	$(FO=3)\cup(FF=3)\cup(MT=3)\cup(MF=2)\cup(CF=3)$	DECR_V GRADUAL, DECR_S SLOW, INCR_Q

#	PREMISE (IF)	CONCLUSION (THEN)
241	$(FO=3)\cup(FF=3)\cup(MT=3)\cup(MF=3)\cup(CF=1)$	E-STOP_S, QUICK TOOL RETURN
242	$(FO=3)\cup(FF=3)\cup(MT=3)\cup(MF=3)\cup(CF=2)$	DECR_V, DECR_S, SLOW
243	$(FO=3)\cup(FF=3)\cup(MT=3)\cup(MF=3)\cup(CF=3)$	DECR_V, DECR_S, SLOW

APPENDIX 2

SOURCE CODES FOR EXPERT CONTROL

This program highlights only the important aspects of expert controller programming. The complete source code is much larger and beyond the scope of this appendix.

```
/*
 * ioutil.h
 *
 * This include file contains the function declarations for the data acquisition system
 *
 * Turbo C Version.
 *
 */

#ifdef LINT_ARGS

/* function declarations for those who want strong type checking
 * on arguments to library function calls
 */

int getavail(int);
int putavail(int);
int inready(int);
int outready(int);
int fgetavail(FILE *);
int fputavail(FILE *);
void binary (int);
int write_ioctl(int,char far *,int);
int read_ioctl(int,char far *,int);
void SetInput(FILE *,int);
void SetOutput(FILE *,int);

#else
```

```
#endif /* LINT_ARGS */
```

```
/*
 * procutil.h
 *
 */

#ifdef LINT_ARGS            /* arg. checking enabled */
int sendcmd (char *, char *, unsigned int, unsigned int *);
int sendlist (char *, char *, int);
int bsendcmd (char *, char *, unsigned int, unsigned int *);
int bsendlist (char *, char *, int);
int cdefcmd (char *, char *, unsigned int, unsigned int *, int);
int cdeflist (char *, char *, int, int);

#else

#endif /* LINT_ARGS */
```

```
#ifdef LINT_ARGS

int getclk (unsigned char *);
int chgclk (unsigned char *);
void delay (unsigned int);

#endif  /* LINT_ARGS */
```

```
/*
 *config.h - system configuration header file
 *
LAST UPDATE
       21 January 1992

*************************************************************************/

#define TICKRATE    20                    /* clock ticks per second (>18) */
```

```
#define MAXHARDTIMERS        1      /* number of hardware timer available */

#define MAXDTIMERS           4      /* number of soft countdown timers */
#define MAXITIMERS           2      /* number of soft interval timers */

#define MAXUSRSTACK     2048        /* max size of each user stack */
#define MINUSRSTACK       32        /* min size of each user stack */

#define SYSBUFSIZ         64        /* size of each system buffer in bytes */
#define MAXSYSBUF          8        /* total number of system buffers */

#define MAXTASKS          16        /* maximum number of resident tasks */
#define MAXEVENTS          8        /* maximum number of events/semaphores */
#define MAXSEMS            8        /* maximum number of semaphores */

#define MAXQUANTUM      1000        /* maximum quantum in milliseconds */
#define MINQUANTUM        50        /* minimum quantum in milliseconds */

#define PCONTROL  1000              /* minimum priority of control tasks */
#define PSERVER          100        /* miminum priority of server tasks */
#define PSLICE            10        /* default priority of slice tasks */
#define PZERO              0        /* floor on task priority */
#define PIDLER            -1        /* Idler priority, must be < PZERO */

#define SLC_TIMER            0      /* timer reserved for slice scheduling */
#define STL_TIMER            1      /* timer reserved for STL scheduling */
```

```
/*
 * c_lib.c
 *

 * Turbo C Version.
 *
 */

#include <ioutil.h>

FILE *DapOut, *DapIn;
```

```
Delay (t)
int t;
{
        long int i;
        for (i=0; i < t*20; i++)
                ;
}

FinFlush (f)
FILE *f;
{
        do {
                while (fgetavail(f))
                        fgetc(f);
                delay(200);
        } while (fgetavail(f));
}

void SetInput (Inf, Num)
FILE *Inf;
int Num;
{
        char s[20];
        int len;
        len = sprintf (s, "I%d", Num);
        write_ioctl (fileno(Inf), s, len);
}

void SetOutput (Inf, Num)
FILE *Inf;
int Num;
{
        char s[20];
        int len;
        len = sprintf (s, "O%d", Num);
        write_ioctl (fileno(Inf), s, len);
}

InitDap ()
```

```
{
        if ((DapIn = fopen("ACCEL","rb")) == NULL) |
           ((DapOut = fopen("ACCEL","wt")) == NULL)) {
                printf("Error: DAP device driver not installed\n");
                exit(1);
        }
        binary(fileno(DapIn));
        write_ioctl(fileno(DapIn),"S,M00",5);
}

LeaveDap ()
{
        write_ioctl(fileno(DapIn),"R",1);
        fclose(DapIn);
        fclose(DapOut);
}

StartCommands ()
{
        fprintf (DapOut, "reset \n");
        fflush (DapOut);
        delay(1000);
        FinFlush (DapIn);
}

CheckErrors ()
{
        int i=1;
        fprintf (DapOut, "display errorq \n");
        fflush (DapOut);
        fscanf (DapIn,"%d",&i);
        if (i != 0) {
                printf ("Error in DAPL command list i=%d\n",i);
                exit(1);
        }
}
```

```
/*
```

```
 * Microstar I/O support -- CPU-independant Time Delay Function
 * Borland Turbo C Version.
*/
#include <dos.h>
#include <stdio.h>
#include <clock.h>
#define TRUE 1
#define FALSE 0

int getclk (clock)
unsigned char *clock;
{
        union REGS inregs, outregs;
        inregs.h.ah = 0x2c;
        intdos(&inregs,&outregs);
        clock[0] = 0;
        clock[1] = 0;
        clock[2] = 0;
        clock[3] = 0;
        clock[4] = outregs.h.ch;
        clock[5] = outregs.h.cl;
        clock[6] = outregs.h.dh;
        clock[7] = outregs.h.dl;
        return(0);
}

void increment (tclock,ms)
unsigned char *tclock;
unsigned int ms;
{
        unsigned int clock[8],i;
        for (i=0; i<8; i++)
                clock[i] = (int) tclock[i];
        clock[7] = clock[7] + (ms/10);
        if ( clock[7] > 99) {
                clock[6] =  clock[6] + clock[7] / 100;
                clock[7] =  clock[7] % 100;
        }
        if ( clock[6] > 59) {
```

```
                clock[5] = clock[5] + clock[6] / 60;
                clock[6] = clock[6] % 60;
        }
        if ( clock[5] > 59) {
                clock[4] = clock[4] + clock[5] / 60;
                clock[5] = clock[5] % 60;
        }
        if ( clock[4] > 23) {
                clock[3] = clock[3] + clock[4] / 24;
                clock[4] = clock[4] % 24;
        }
        for (i=0; i<8; i++)
                tclock[i] = (char) clock[i];
}

int time_gt (clock1, clock2)
unsigned char *clock1;
unsigned char *clock2;
{
        int i;
        for (i=1;i<8;i++)
                if (clock1[i] < clock2[i])
                        return(FALSE);
                else if (clock1[i] > clock2[i])
                        return(TRUE);
        return(FALSE);
}

void delay (ms)
unsigned int ms;
{
        unsigned char clock[8],nclock[8];
        getclk(clock);
        increment(clock,ms);
        do {
                getclk(nclock);
        } while (time_gt(clock,nclock));
}
```

```
/***********************************************************************

FILE
   com.c - IBM PC non bios COM programming

ENTRY ROUTINES
   cominit  - initialize com
   comienb  - enable com interrupts
   comidis  - disable com interrupts
   comin    - com input
   comout   - com output

PERTICULAR  ROUTINES
   combase  - get base address of COM device

REMARKS
   This module supplies COM programming routines that are independent
   of the BIOS. The BIOS limit of only 2 COM devices are overcomed.
   It is assumed that all COM devices use the INS8250 chip.

LAST UPDATE
     24 June 1990

***********************************************************************/

/***********************************************************************
                     I M P O R T S
***********************************************************************/

#include "envir.h"          /* environment declarations */
#include "inout.h"          /* i/o function mapping */
#include "com.h"            /* exported module declarations */

/***********************************************************************
                  P R I V A T E    D A T A
***********************************************************************/

/************ table of baud rates and associated divisor ************/
```

```
#define BAUDTABSIZ      16      /* number of entries in baud table */

static int baudtab[BAUDTABSIZ][2] = {
    {    50, 0x900 },
    {    75, 0x600 },
    {   110, 0x417 },
    {   135, 0x359 },           /* actually 134.5 baud */
    {   150, 0x300 },
    {   300, 0x180 },
    {   600, 0x0C0 },
    {  1200, 0x060 },
    {  1800, 0x040 },
    {  2000, 0x03A },
    {  2400, 0x030 },
    {  3600, 0x020 },
    {  4800, 0x018 },
    {  7200, 0x010 },
    {  9600, 0x00C },
    { 19200, 0x006 }            /* baud rate not recommended by IBM */
};

/*************************************************************************
          F O R W A R D   D E C L A R A T I O N S
*************************************************************************/

#ifdef ANSI
int combase(int);
#else
int combase();
#endif

/*************************************************************************
              E N T R Y   R O U T I N E S
*************************************************************************/

/*-------------------------------------------------------------------
FUNCTION
   COMINIT - initialize com device
```

```
SYNOPSIS
   int cominit(com, baud, parity, wordsize, stopbits, rts, dtr)
   int com, baud, parity, wordsize, stopbits, rts, dtr;

PARAMETERS
   com      - com device (1 for COM1, 2 for COM2 etc...)
   baud     - baud rate (135.5 is specified as 135)
   parity   - 0 - none, 1 - odd, 2 - even
   wordsize - 5, 6, 7, 8
   rts      - initial RTS state (0 or 1)
   dtr      - initial DTR state (0 or 1)

RETURNS
   0 if all is well, -1 otherwise

REMARKS
   Wider range of baud rates than possible with DOS's MODE command.

LAST UPDATE
   4 May 1990
-----------------------------------------------------------------*/

cominit(com, baud, parity, wordsize, stopbits, rts, dtr)
int com, baud, parity, wordsize, stopbits, rts, dtr;
{
   int base;           /* base address */
   char lcr;           /* LCR control byte */
   char mcr;           /* MCR control byte */
   int i;              /* iteration variable */


   if ((base = combase(com)) < 0)     /* invalid com specifier */
      return(-1);

   for (i = 0; i < BAUDTABSIZ; i++)
      if (baudtab[i][0] == baud)     /* baud rate is supported */
         break;

   if (i < BAUDTABSIZ)
```

```
{
   out(base+LCR, 0x80);          /* access divisor latch */

   out(base, baudtab[i][1] & 0xFF);
   out(base+1, baudtab[i][1] >> 8);

   out(base+LCR, 0x00);          /* reset DLAB */
}
else
   return(-1);


lcr = 0;                         /* initialize LCR byte */

switch (wordsize)
{
   case 5:   lcr &= ~0x01;     break;
   case 6:   lcr |= 0x01;      break;
   case 7:   lcr |= 0x02;      break;
   case 8:   lcr |= 0x03;      break;
   default:  return(-1);       /* invalid word size */
}

switch (parity)
{
   case 0:   lcr &= ~0x08;     break;    /* no parity */
   case 1:   lcr &= ~0x10;     break;    /* odd parity */
   case 2:   lcr |= 0x10;      break;    /* even parity */
   default:  return(-1);                 /* invalid parity */
}

switch (stopbits)
{
   case 1:   lcr &= ~0x04;     break;    /* 1 stop bit */
   case 2:   lcr |= 0x04;      break;    /* 1.5 if wordsize=5 */
   default:  return(-1);                 /* error */
}

out(base+LCR, lcr);              /* set line control */
```

```
    mcr = (char)in(base+MCR);

    if (rts && dtr)
        out(base+MCR, mcr | 0x03);
    else if (dtr)
        out(base+MCR, mcr | 0x01);
    else if (rts)
        out(base+MCR, mcr | 0x02);

    return(0);
}



/*-----------------------------------------------------------------
FUNCTION
    COMIENB  -  enable com interrupts

SYNOPSIS
    int comienb(com, intrps)
    int com, intrps;

PARAMETERS
    com      -  com id
    intrps  -  which interrupts to enable (defined in com.h)

RETURNS
    0  if all is well,  -1 otherwise

LAST UPDATE
    4 May 1990
-----------------------------------------------------------------*/

int comienb(com, intrps)
int com, intrps;
{
    int base;                     /* base address */
```

```
    if ((base = combase(com)) >= 0)
    {
        out(base+MCR, in(base+MCR) | 0x08);     /* enable them */
        out(base+IER, intrps & 0x0F);           /* select interrupts */

        return(0);
    }
    else                        /* com id error */
        return(-1);

}



/*-----------------------------------------------------------------
FUNCTION
    COMIDIS  -  disable com interrupts

SYNOPSIS
    int comidis(com, intrps)
    int com, intrps;

PARAMTERS
    com     - com id
    intrps -  what interrupts to disable

RETURNS
    0  if all is well,  -1 otherwise

LAST UPDATE
    4 May 1990
-----------------------------------------------------------------*/

int comidis(com, intrps)
int com, intrps;
{
    int base;               /* base address */
```

```
    if ((base = combase(com)) >= 0)
    {
        intrps &= 0x0F;             /* mask off extraneous bits */

        if (intrps == 0)            /* disable all interrupts */
        {
            out(base+IER, 0);
            out(base+MCR, in(base+MCR) & ~0x08);
        }
        else
        {
            out(base+IER, (~intrps) & 0x0F);
        }

        return(0);
    }
    else
        return(-1);

}


#if 0       /* taken out to save code space */

/*-----------------------------------------------------------------
FUNCTION
    COMOUT - non interrupt driven com character output

SYNOPSIS
    int comout(com, c)
    int com;
    char c;

PARAMETERS
    com  - com id
    c    - output character

RETURNS
    0 if all is well, -1 otherwise
```

```
LAST UPDATE
   4 May 1990
-------------------------------------------------------------------*/

int comout(com, c)
int com;
char c;
{
   int base;          /* com base address */
   int mcr;           /* MCR address */
   int lsr;           /* LSR address */


   if ((base = combase(com)) >= 0)
   {
      mcr = base+MCR;
      lsr = base+LSR;

      while (!(in(lsr) & THRE))   /* wait for THRE */
         ;

      out(base, c);

      return(0);
   }
   else
      return(-1);

}



/*-------------------------------------------------------------------
FUNCTION
   COMIN  -  non interrupt com input

SYNOPSIS
   int comin(com)
   int com;
```

```
PARAMETERS
   com - com id

RETURNS
   input char cast to int if all is well, -1 otherwise

LAST UPDATE
   4 May 1985
-----------------------------------------------------------------*/

int comin(com)
int com;
{
   int base;           /* com base address */
   int lsr;            /* LSR address */
   int c;              /* input value */


   if ((base = combase(com)) >= 0)
   {
      lsr = base+LSR;

      while (!(in(lsr) & DAV))      /* wait for DAV */
         ;

      c = in(base);
   }
   else
      c = -1;

   return(c);
}

#endif


/***********************************************************************
            P R I V A T E   R O U T I N E S
***********************************************************************/
```

```
/*-----------------------------------------------------------------
FUNCTION
   COMBASE  -  get com base address from com id

SYNOPSIS
   static combase(com)
   int com;

PARAMETERS
   com  -  com id  (1 for COM1, 2 for COM2 etc...)

RETURNS
   base address if all is well,  -1 otherwise

LAST UPDATE
   24 June 1985
-----------------------------------------------------------------*/

static int combase(com)
int com;
{
   int base;              /* base address */


   switch (com)
   {
      case 1:         /* COM1 */

         base = COM1BASE;
         break;

      case 2:         /* COM2 */

         base = COM2BASE;
         break;

      default:

         base = -1;
```

```
   }

   return(base);
}
```

```
;
; Microstar I/O support -- assembly language DOS interface
;

;
DOS         EQU   21H

DGROUP            GROUP _DATA

_DATA       SEGMENT WORD PUBLIC 'DATA'
TEMP        DW  0       ; this temporary variable is used by the
                        ;   getavail and putavail functions
_DATA       ENDS

            ASSUME    CS:_TEXT,SS:DGROUP

_TEXT             SEGMENT BYTE PUBLIC 'CODE'

      IFDEF LARGE                   ; large memory model
PROC_ENTRY        MACRO PROC_NAME
PROC_NAME         PROC FAR
            PUBLIC PROC_NAME
            PUSH BP
            MOV BP,SP
            ADD BP,2
            ENDM

PROC_EXIT       MACRO PROC_NAME
            SUB BP,2
            MOV SP,BP
            POP BP
            RET
```

```
PROC_NAME       ENDP
                ENDM
        ENDIF
        IFDEF SMALL                     ; small memory model
PROC_ENTRY      MACRO PROC_NAME
PROC_NAME       PROC NEAR
                PUBLIC PROC_NAME
                PUSH BP
                MOV BP,SP
                ENDM

PROC_EXIT  MACRO PROC_NAME
                MOV SP,BP
                POP BP
                RET
PROC_NAME       ENDP
                ENDM
        ENDIF

        PROC_ENTRY _getavail
                PUSH  DS
                MOV   AX,SS
                MOV   DS,AX
                MOV   TEMP,0
                MOV   DX,OFFSET DGROUP:TEMP
                MOV   AH,68
                MOV   AL,2
                MOV   BX,[BP+4]
                MOV   CX,2
                INT   DOS
                MOV   AX,TEMP
                JNC   A1
                 MOV AX,0
A1:             POP  DS
        PROC_EXIT _getavail

        PROC_ENTRY _putavail
                PUSH  DS
                MOV   AX,SS
```

```
            MOV   DS,AX
            MOV   TEMP,1
            MOV   DX,OFFSET DGROUP:TEMP
            MOV   AH,68
            MOV   AL,2
            MOV   BX,[BP+4]
            MOV   CX,2
            INT   DOS
            MOV   AX,TEMP
            JNC   A2
             MOV AX,0
A2:         POP  DS
        PROC_EXIT _putavail

        PROC_ENTRY _inready
            MOV  AH,68
            MOV  AL,6
            MOV  BX,[BP+4]
            INT  DOS
            CMP  AL,255
            JNE  A3
             MOV AX,1
             JMP A4
A3:          MOV AX,0
A4:
        PROC_EXIT _inready

        PROC_ENTRY _outready
            MOV  AH,68
            MOV  AL,7
            MOV  BX,[BP+4]
            INT  DOS
            CMP  AL,255
            JNE  A5
             MOV AX,1
             JMP A6
A5:          MOV AX,0
A6:
        PROC_EXIT _outready
```

```
        PROC_ENTRY _binary
                MOV  AH,68
                MOV  AL,0
                MOV  BX,[BP-4]
                INT  DOS
                MOV  AH,68
                MOV  AL,1
                MOV  DH,0
                OR   DL,32
                INT  DOS
        PROC_EXIT _binary

        PROC_ENTRY _write_ioctl
                PUSH  DS
                MOV   DX,[BP-6]
                MOV   DS,[BP-8]
                MOV   AH,68
                MOV   AL,3
                MOV   BX,[BP-4]
                MOV   CX,[BP-10]
                INT   DOS
                MOV   AX,[BP-10]
                JNC   A7
                 MOV AX,0
A7              POP  DS
        PROC_EXIT _write_ioctl

        PROC_ENTRY _read_ioctl
                PUSH  DS
                MOV   DX,[BP+6]
                MOV   DS,[BP-8]
                MOV   AH,68
                MOV   AL,2
                MOV   BX,[BP-4]
                MOV   CX,[BP-10]
                INT   DOS
                JNC   A8
                 MOV AX,0
A8              POP  DS
```

```
        PROC_EXIT _read_ioctl

_TEXT           ENDS

                END
```

```
/*
   Microstar I/O support -- C language stream I/O interface

   Borland Turbo C Version

#define LINT_ARGS 1
#include <stdio.h>
#include <ioutil.h>

int fgetavail (f)
FILE *f;
{
        return(f->level - getavail(f->fd));
}

int fputavail (f)
FILE *f;
{
        if ((f->level) < 0)
                return(putavail(f->fd) - (f->level) - 1);
        else
                return(putavail(f->fd) - (f->bsize));
}
```

```
/*
   Microstar I/O support -- Custom Command Definition Functions
   Borland Turbo C Version
*/
#define LINT_ARGS 1
#include <stdlib.h>
#include <stdio.h>
#include <ioutil.h>
```

```
#include <procutil.h>
#include <string.h>
#include <math.h>
#include <io.h>
#include <clock.h>
#define TRUE 1
#define FALSE 0

static void finflush(f)
FILE *f;
{
        while (fgetavail(f))
                getc(f);
}

void SetOutput (Inf, Num)
FILE *Inf;
int Num;
{
        char s[20];
        int len;
        len = sprintf (s, "O%d", Num);
        write_ioctl (fileno(Inf), s, len);
}

static char *trim (name)
char *name;
{
        int i;

        while (name[0] == ' ')
                name++;
        if (name[0] == '\0')
                return(name);
        i = strlen(name) - 1;
        while (name[i] == ' ') {
                name[i] = '\0';
                i--;
        }
```

```
        return(name);
}

void strcpy2 (dest, src, max_len)
char *dest, *src;
int max_len;
{
        int i;
        for (i=0; i < (max_len-1) & src[i] != '\0'; i++)
                dest[i] = src[i];
        dest[i] = '\0';
}

void add_suffix (name, dest, max_len, suffix)
char *name, *dest, *suffix;
int max_len;
{
        int i;

        strcpy2(dest,name,max_len);
        if (dest[0] == '\0')
                return;
        i = 0;
        if (dest[i+1] == ':')
                i = i + 2;
        if (dest[i] == '.')
                i = i + 1;
        if (dest[i] == '.')
                i = i + 1;
        if (dest[i] == '\0')
                return;
        if (strchr(&(dest[i]), '.') == NULL) {
                i = strlen(dest);
                strcpy2(&(dest[i]), suffix, max_len-i);
        }
}

void get_root (name, dest, max_len)
char *name, *dest;
```

```
int max_len;
{
        int i;
        char *p,*q,*r;

        dest[0] = '\0';
        if (name[0] == '\0')
                return;
        i = 0;
        if (name[i+1] == ':')
                i = i + 2;
        if (name[i] == '.')
                i = i + 1;
        if (name[i] == '.')
                i = i + 1;
        if (name[i] == '\0')
                return;
        p = name + i;
        if ((q = strchr(p,'.')) == NULL)
                q = p + strlen(p);
        if (p == q)
                return;
        q = q - 1;
        for (r=q; r != p & *r != '\\'; r--)
                ;
        if (*r == '\\')
                r = r + 1;
        q = q + 1;
        for (i=0; i < (max_len-1) & r != q; )
                dest[i++] = *r++;
        dest[i] = '\0';
}

#define LINE_SIZE 200
#define MAX_FNAME 30

int cdefcmd (dap_name, in_fname, ssize, len, bin_flag)
char *dap_name, *in_fname;
unsigned int ssize, *len;
```

```
int bin_flag;
{
        char root[MAX_FNAME],fname[MAX_FNAME],line[LINE_SIZE],*s;
        FILE *inf, *dap_in, *dap_out, *dap_binout;
        int i,s_error,pad;

        if ((dap_in = fopen(dap_name,"rb")) == NULL)
                return(-1);
        if ((dap_out = fopen(dap_name,"wt")) == NULL)
                return(-1);
        if ((dap_binout = fopen(dap_name,"wb")) == NULL)
                return(-1);
        binary(fileno(dap_in));
        binary(fileno(dap_binout));
        write_ioctl(fileno(dap_in),"S,M00",5);

        fprintf(dap_out,"D ERRORQ\n");
        fflush(dap_out);
        delay(100);
        for (i=0; (i<5) && (fgetavail(dap_in)==0); i++)
                delay(200);
        if (fgetavail(dap_in) == 0)
                return(-2);
        finflush(dap_in);

        add_suffix(in_fname,fname,MAX_FNAME,".BIN");
        get_root(fname,root,MAX_FNAME);

        fprintf(dap_out,"ERASE %s\n",root);
        fflush(dap_out);
        finflush(dap_in);

        if ((inf = fopen(fname,"rb")) == NULL)
                return(1);

        *len = (unsigned int) filelength(fileno(inf));

        s_error   FALSE;
        finflush(dap_in);
```

```
if (bin_flag) {
        fprintf(dap_out,"OPTION DECIMAL=ON,ERRORQ=OFF\n");
        fflush(dap_out);
        delay(200);
        finflush(dap_in);
        pad = ((*len % 2) == 1);
        fprintf(dap_out,"BDOWNLOAD %s %u %u $BININ\n", root, ssize,
                pad ? *len+1 : *len);
        fflush(dap_out);
        delay(200);
        while (inready(fileno(dap_in)))
                if (getc(dap_in) == '*')
                        s_error = TRUE;
        SetOutput(dap_out,1);
        i = 0;
        while ((i < *len) & !s_error) {
                char c;
                c = (char) getc(inf);
                putc(c, dap_binout);
                i++;
        }
        if (pad)
                putc(0, dap_binout);
        fclose(inf);
        fflush(dap_binout);
        SetOutput(dap_out,0);
        fprintf(dap_out,"OPTION ERRORQ=ON\n");
        fflush(dap_out);
        delay(100);
        finflush(dap_in);
} else {
        fprintf(dap_out,"OPTION DECIMAL=OFF,ERRORQ=OFF\n");
        fflush(dap_out);
        delay(200);
        finflush(dap_in);
        fprintf(dap_out, "DOWNLOAD %s 0%X 0%X\n", root, ssize, *len);
        fflush(dap_out);
        delay(100);
        s_error = FALSE;
```

```
            while (inready(fileno(dap_in)))
                    if (getc(dap_in) == '*')
                            s_error = TRUE;
            i = 1;
            while ((i <= *len) & !s_error) {
                    fprintf(dap_out, " %X", getc(inf));
                    if (i % 8 == 0) {
                            putc('\n', dap_out);
                            fflush(dap_out);
                    }
                    while (inready(fileno(dap_in)))
                            if (getc(dap_in) == '*')
                                    s_error = TRUE;
                    i = i + 1;
            }
            fclose(inf);
            putc('\n', dap_out);
            fflush(dap_out);
            fprintf(dap_out,"OPTION DECIMAL=ON,ERRORQ=ON\n");
            fflush(dap_out);
            delay(100);
            finflush(dap_in);
    }
    write_ioctl(fileno(dap_in),"R",1);
    fclose(dap_in);
    fclose(dap_out);
    fclose(dap_binout);
    if (s_error)
            return(3);
    else
            return(0);
}

#define LINESIZE 100

static int execute_line (dap_name, listf, print, bin_flag)
char *dap_name;
FILE *listf;
int print, bin_flag;
```

```
{
        char line[LINESIZE], *lp, fname[MAX_FNAME], s_str[20];
        unsigned int ssize, len;
        int temp;

        if feof(listf)
                return(0);
        if (fgets(line, LINESIZE, listf) == NULL)
                return(feof(listf) ? 0 : 5);
        lp = trim(line);
        if (lp[0] == '\0' | lp[0] == '\n')
                return(0);
        if (sscanf(lp, "%29s %19s", fname, s_str) == EOF)
                return(5);
        if ((ssize = atoi(s_str)) == 0)
                return(5);
        if ((temp = cdefcmd (dap_name, fname, ssize, &len, bin_flag)) != 0)
                return(temp);
        if (print)
                printf(". %s: transfer completed (%u bytes)\n", fname,
                        len);
        return(0);
}


int cdeflist (dap_name, in_lname, print, bin_flag)
int print;
char *dap_name, *in_lname;
int bin_flag;
{
        char lname[MAX_FNAME];
        FILE *listf;
        int s_error;

        add_suffix(in_lname,lname,MAX_FNAME,".TXT");
        if ((listf = fopen(lname,"rt")) == NULL)
                return(4);
        s_error = 0;
        while (!feof(listf) & !s_error)
```

```
        s_error = execute_line(dap_name, listf, print, bin_flag);
    fclose(listf);
    return(s_error);
}

int bsendlist (dap_name, in_lname, print)
int print;
char *dap_name, *in_lname;
{
    return(cdeflist(dap_name, in_lname, print, TRUE));
}

int bsendcmd (dap_name, in_fname, ssize, len)
char *dap_name, *in_fname;
unsigned int ssize, *len;
{
    return(cdefcmd(dap_name, in_fname, ssize, len, TRUE));
}

int sendlist (dap_name, in_lname, print)
int print;
char *dap_name, *in_lname;
{
    return(cdeflist(dap_name, in_lname, print, FALSE));
}

int sendcmd (dap_name, in_fname, ssize, len)
char *dap_name, *in_fname;
unsigned int ssize, *len;
{
    return(cdefcmd(dap_name, in_fname, ssize, len, FALSE));
}
```

```
/*************************************************************************

FILE
  cntrl4.c - Motor Speed for Multiple Motors
```

```
ENTRY ROUTINES
    control  -  execute control algorithm
    init     -  interpret command line for initialization
    chng     -  interpret command line for speed change
    setv     -  interpret command line for setpoint
    setsmp   -  interpret command line for sampling rate
    setmot   -  allocate and initialize motor structures

PRIVATE ROUTINES

    pimotor  -  apply motor control
    cpmotor  -  copy motor descriptors
    getv     -  get  reading
    mact     -  send controller output to motor actuator

REMARKS
    This routine implements multiple motor control; each motor has
    completely independent sampling interval, control parameters, etc.

LAST UPDATE
    1 March 1994


    Copyright (c) 1994  Perdur S.Subramanya

*************************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include "envir.h"
#include "ioutil.h"
#include "dap.h"
#include "time0.h"

#include "user.h"

#ifdef SIMRT
```

```
#ifdef ANSI
extern void tstep(void);
extern void sim_init(int);
extern void sim_reset(char *);
#else
extern void tstep();
extern void sim_init();
extern void sim_reset();
#endif
#endif

/***********************************************************************
        MODULE  DATA  STRUCTURES  AND  VARIABLES
***********************************************************************/


#define MTODAC      1.0         /* output units to DAC units */

#define TIME_GRAIN  0.001       /* left time granularity */

/*------------------------------------------------------*/
/*  This structure describes the control and configurational */
/*  parameters for velocity control of  motors.       */
/*------------------------------------------------------*/

typedef struct motor {
   int ichan;      /* A/D channel number */
   int mchan;       /* D/A channel number */
   double v;       /* velocity */
   double vref;    /* velocity setpoint */
   double kp;       /* proportional chng */
   double ki;      /* integral chng */
   double ival;    /* integrator accumulated value */
   double vcon;     /* constant in control equation */
   double m;        /* controller output */
   double mmin;     /* lower limit for controller output */
   double mmax;     /* upper limit for controller output */
   double tsamp;  /* sample interval */
   double tleft;  /* time left to next sample */
```

```
} MOTOR;

static MOTOR *pm0;      /* pointer to the structure for motor #0 */
static int nmotors;     /* number of motors */

/*------------------------------------------------------------*/
/*  motdef is a set of default values that is used to initialize  */
/*  newly allocated motor control structure.                   */
/*------------------------------------------------------------*/

static MOTOR motdef = {
        0,      /* A/D channel defaults to 0 */
        0,      /* D/A channel defaults to 0 */
      0.0,      /* zero initial velocity */
   1000.0,      /* default velocity setpoint */
      1.0,      /* kp defaults to 1.0 */
      0.0,      /* no integrator by default, P control by default */
      0.0,      /* zero integration sum */
      0.0,      /* constant term in control equation */
      0.0,      /* no controller output initially */
  -2048.0,      /* lower output limit */
   2047.0,      /* upper output limit */
      0.5,      /* default sampling interval of 0.5 seconds */
      0.0       /* time left to next sample */
};

/*************************************************************************
          F O R W A R D   D E C L A R A T I O N S
*************************************************************************/

#ifdef ANSI

double  getv(int);
void    mact(int, double);
void    pimotor(MOTOR *);
void    cpmotor(MOTOR *, MOTOR *);

#else
```

```
double  getv();
void    mact();
void    pimotor();
void    cpmotor();

#endif

/**************************************************************************
              E N T R Y   R O U T I N E S
**************************************************************************/

/*----------------------------------------------------------------
PROCEDURE
   CONTROL  -  this function does the control

SYNOPSIS
   void control(cline)
   char cline[];


LAST UPDATE
   1 March 1994
      use ANSI features
----------------------------------------------------------------*/

void control(cline)
char *cline;
{
   MOTOR *pm;          /* pointer to motor data structure */
   long i;             /* iteration counter */
   int m;              /* motor number index */


   prologue();         /* prepare scheduling kernel */

   pm = pm0;           /* set pm to point at first motor structure */

   for (m = 0; m < nmotors; m++)
   {
```

```
        pm->ival = 0.0;     /* initialize the integrator */

        SetCntrlTask("motor", pimotor, (void *)pm, 1000,
                    (long)(pm->tsamp * 1000.0 + 0.5));

        pm++;               /* next motor */
    }

    start();            /* start the scheduler */

}


/*-----------------------------------------------------------------
PROCEDURE
    INIT - process system initialization command

SYNOPSIS
    void init(cline)
    char *cline;

REMARKS
    This routine must be called before control() is called.

LAST UPDATE
    1 March 1994
        change sim_init()
-----------------------------------------------------------------*/

void init(cline)
char *cline;
{

    ad_init();          /* initialize A/D */
    da_init();          /* initialize D/A */

#ifdef SIMRT
    sim_reset(cline);
```

```
#endif

}


/*---------------------------------------------------------------
PROCEDURE
   Chng  -  interpret input line for controller chngs specification

SYNOPSIS
   void chng(cline)
   char *cline;

LAST UPDATE
   1 March 1994
      use ANSI features
---------------------------------------------------------------*/

void chng(cline)
char *cline;
{
   int mn;                    /* motor number */
   double kp, ki, vcon;   /* temporary chng and constant terms */


   if (sscanf(cline, "%d %lf %lf %lf", &mn, &kp, &ki, &vcon) == 4)
   {
      if ((mn >= 0) && (mn < nmotors))
      {
         /*
          * Set the appropriate fields of the motor descriptor.
          * Note that (pm0 + mn) is a pointer to the descriptor
          * for the mn-th motor; hence the use of "->" to
          * dereference the descriptor fields.
          */

         (pm0 + mn)->kp = kp;
         (pm0 + mn)->ki = ki;
```

```
            (pm0 + mn)->vcon = vcon;

#ifdef DEBUG
            printf("<chng> motor = %d, ", mn);
            printf("kp = %lf, ki = %lf, vcon = %lf\n", kp, ki, vcon);
#endif
        }
        else
            printf("chng: invalid motor number: %d\n", mn);
    }
    else
        printf("chng: invalid or insufficient parameters\n");

}



/*------------------------------------------------------------------
PROCEDURE
    SETV  -  get velocity setpoint from command line

SYNOPSIS
    void setv(cline)
    char *cline;

PARAMETER
    cline  -  pointer to null terminated input command line

LAST UPDATE
    1 March 1994
        use ANSI features
------------------------------------------------------------------*/

void setv(cline)
char *cline;
{
    int mn;          /* motor number */
    double vref;     /* temporary holder for setpoint value */
```

```
    if (sscanf(cline, "%d %lf", &mn ,&vref) == 2)
    {
        if ((mn >= 0) && (mn < nmotors))
        {
            (pm0 + mn)->vref = vref;        /* record velocity reference */

#ifdef DEBUG
            printf("<setv> motor = %d, vref = %lf\n", mn, vref);
#endif
        }
        else
            printf("setv: bad motor number: %d\n", mn);
    }
    else
        printf("setv: invalid or insufficient parameters\n");

}


/*-----------------------------------------------------------------
PROCEDURE
    SETSMP  -  set sampling interval from command line specification

SYNOPSIS
    void setsmp(cline)
    char *cline;

PARAMETER
    cline  -  pointer to null terminated input command line

LAST UPDATE
    1 March 1994
        use ANSI features
-----------------------------------------------------------------*/

void setsmp(cline)
char *cline;
{
```

```
   int mn;              /* motor number */
   double tsamp;        /* temporary holder for sample time */


   if (sscanf(cline,"%d %lf", &mn, &tsamp) == 2)
   {
      if ((mn >= 0) && (mn < nmotors))
      {
         (pm0 + mn)->tsamp = tsamp;       /* set sample interval */

#ifdef DEBUG
         printf("<setsmp> motor = %d, tsamp = %f\n", mn, tsamp);
#endif
      }
      else
         printf("setsmp: bad motor number: %d\n", mn);
   }
   else
      printf("setsmp: invalid or insufficient parameters\n");

}



/*-----------------------------------------------------------------
PROCEDURE
   SETMOT  -  allocate and initialize data structures for the number
              of motors requested by the user.
SYNOPSIS
   void setmot(void)

REMARKS
   This routine must be called as the first action of the main program.

   Motor descriptors initialized from defaults specified in "motdef"

LAST UPDATE
   1 March 1994
      use ANSI features
```

```
-------------------------------------------------------------*/

void setmot()
{
   MOTOR *pm;          /* pointer to motor data structures */
   int nm;             /* number of motors */
   char cbuf[20];      /* input line */
   int damin, damax;   /* D/A output limits */
   int done;           /* flag, if set, o.k. to return */
   int i;              /* loop index */


   do
   {
      done = 1;

      /*
       * Prompt user and get response line. Because scanf() does not
       * remove the newline character at the end of lines, fgets() is
       * used instead since the newline character may cause trouble
       * with other console functions.
       */

      fputs("How many motors? ", stdout);
      fgets(cbuf, 20, stdin);          /* get at most 19 characters */

      if (sscanf(cbuf, "%d", &nm) == 1)  /* decode input */
      {
         da_limits(&damin, &damax);      /* find DAC limits */

         motdef.mmin = (double)damin / MTODAC;   /* set output limits */
         motdef.mmax = (double)damax / MTODAC;

         /*
          * Dynamically allocate memory for the required number of
          * motor structures. Note that calloc() is used instead of
          * malloc() since calloc() is suppose to guarantee that the
          * allocated block of memory has the correct alignment.
          * Note also the use of "casts" in the allocation statement;
```

```
         * though not strictly necessary, it is recommended as good
         * programming style.
         */

        if ((pm0 = (MOTOR *)calloc(nm, sizeof(MOTOR))) == (MOTOR *)NULL)
        {
           printf("INSUFFICIENT MEMORY: ");
           printf("cannot allocate %d motor blocks\n", nm);
           exit(1);
        }

        for (i = 0, pm = pm0; i < nm; i++, pm++)
        {
           cpmotor(&motdef, pm);   /* init with defaults from motdef */

           pm->ichan = i;  /* set both A/D and D/A channels to i; the */
           pm->mchan = i;  /*   actual channels depends on your setup */
        }

        nmotors = nm;         /* record number of motor descriptors */

#ifdef SIMRT
        sim_init(nm);   /* if real-time simulation is to be used, */
#endif                  /*   initialize the simulation module.    */

        if (nm == 1)
           printf("\nmotor 0 has been initialized\n");
        else
           printf("\nmotors 0 to %d have been initialized\n", nm - 1);
     }
     else
     {
        printf("invalid entry\n");
        done = 0;
     }
  }
  while (!done);

}
```

```
/***********************************************************************
               P A R T I C U L A R   R O U T I N E S
***********************************************************************/

/*------------------------------------------------------------------
PROCEDURE
   EXMOTOR  -  apply expert control for motor

SYNOPSIS
   static void exmotor(pm, prnt)
   MOTOR *pm;
   int prnt;

PARAMETERS
   pm    -  pointer to motor structure requiring control
   prnt  -  1- gradual,2- slow,3-sudden

LAST UPDATE
   2 March 1994
      use ANSI features
   1March 1994
      modify for  background scheduler
------------------------------------------------------------------*/

static void exmotor(pm, prnt)
MOTOR *pm;
{
 x = prnt;
switch(x) {
case 1: mact_gradl(pm->mchan, pm->m);
break;
case 2: mact_slow(pm->mchan,pm->m);
break;
case 3: mact_sudden(pm->mchan, pm->m);
break;
  fprintf(report,"motor = %d, v = %7.3lf, m = %7.3lf\n", mn, pm->v, pm->m);

 }
```

```
/*-----------------------------------------------------------------
PROCEDURE
   CPMOTOR  -  copy one motor descriptor to another

SYNOPSIS
   static void cpmotor(from, to)
   MOTOR *from, *to;

PARAMETERS
   from  -  pointer to source
   to    -  pointer to destination

REMARKS
   Standard (K&R) C does not allow entire structures to be assigned
   (this will change with the proposed ANSI X3J11 standards). This
   restriction is very inconvenient and this routine gets around this
   by doing a byte-by-byte copy of the descriptor contents. This
   technique may not work on all computers, but it is reasonably
   robust and is much easier than doing field by field assignments.

   If you are using an ANSI conforming C compiler, just do a structure
   assignment. The compiler will take care of the rest.

LAST UPDATE
   2 March 1994
      use ANSI features
-----------------------------------------------------------------*/
static void cpmotor(from, to)
MOTOR *from, *to;
{
#ifdef ANSI     /* can do structure assignment */
   *to = *from;
#else           /* do explicit byte by byte copy */

   register char *src, *dst;   /* source and destination pointers */
   unsigned count;             /* byte count */
```

```
    src = (char *)from;     /* make fast copy of source pointer */
    dst = (char *)to;       /* make fast copy of destination pointer */

    for (count = 0; count < sizeof(MOTOR); count++)
        *dst++ = *src++;

#endif

}
/*-------------------------------------------------------------------
FUNCTION
    GETV  -  get current motor velocity

SYNOPSIS
    static double getv(voltage)
    int channel;

PARAMETER
    voltage  -  current voltage output to the controller
RETURNS
    current motor velocity

LAST UPDATE
    2 March 1994
        use ANSI features
-------------------------------------------------------------------*/

static double getv(channel)
int channel;
{

    return((double)ad_wread(channel) * V);
}



/*-------------------------------------------------------------------
PROCEDURE
    mact_slow, mact_gradl, mact_sudden  -  send controller output to D/A converter
```

```
SYNOPSIS
   static void mact_slow(channel, output)
   static void mact_gradl(channel, output)
   static void mact_sudden (channel, output)
   int channel;
   double output;

PARAMETERS
   channel -  D/A channel number
   output  -  controller output


LAST UPDATE
   2 March 1994
      use ANSI features
----------------------------------------------------------------------*/

static void mact_gradl(channel, output)
int channel;
double output;
{
   out=0.5;
  while  (output != out) {
     (da_write(channel, (int)((out ) + 0.5));
sleep(2);
}
}

static void mact_slow(channel, output)
int channel;
double output;
{
  out=0.5;
  while  (output != out) {
     (da_write(channel, (int)((out ) + 0.5));
sleep(0.5);
}
}
```

```
static void mact_sudden(channel, output)
int channel;
double output;
{

     (da_write(channel,output ));
}
```

```
/* file clust.c : clustering algorithm for classification of chip sizes
*********/
#include <stdio.h>
#include <ctype.h>
#include <math.h>

#define NMXPATTERN  100
#define NMXATTR     100

int ninput;
int ninattr;
float threshold;
int testing;
float pattern[NMXPATTERN][NMXATTR];
float b[NMXPATTERN][NMXATTR];
int cluster_tbl[NMXPATTERN][NMXATTR];
float x[NMXATTR];
float ed[NMXPATTERN];
int active_nodes;
FILE *in, *out;
int debug_flag;
int total_time;

user_session()
{
int i,j;
char file_name[20];

if ((in = fopen(file_name,"r")) == NULL)
{
```

```
  printf("\n cannot open input file \n");
  exit(0);
}

testing = 0;
total_time = 0;
}

read_all_patterns()
{
  int i,j;

  for (i = 0; i < ninput; i++)
    for (j = 0; j< ninattr; j++)
    fscanf( in, "%e", &pattern[i][j]);

  fclose(in);
}
first_node()
{
  int i;

  for (i = 0; i < ninattr; i++)
    b[0][i] = pattern[0][i];

    active_nodes = 1;
    cluster_tbl[0][0] = 1;
}

form_new_node(input_no)
int input_no;
{
  int i;

  for (i = 0; i < ninattr; i++)
    b[active_nodes][i] = pattern[input_no][i];


  cluster_tbl[active_nodes][0] = 1;
```

```
  cluster_tbl[active_nodes][1] = input_no;
  active_nodes++;
}

compute_euc_dist()
{
 int i,j;


  for (j = 0; j < active_nodes; j++)
  {
    ed[j] = 0.0;
    for(i = 0; i < ninattr; i++)
    ed[j] = ed[j] + ((b[j][i]-x[i]) * (b[j][i]-x[i]));

 }


int compare_min_ed(input_no)
{
 int i, cluster_no;
 float min;

 min = 10000;

 for (i = 0; i < active_nodes; i++)
   if (ed[i] < min)
   {
    min = ed[i];
    cluster_no = i;
    }
    sqrt(ed[cluster_no]), cluster_no, input_no);

 if (sqrt(ed[cluster_no]) <= threshold)
   return(cluster_no);
 else return(-99);
}
update_wts(cluster_no,input_no)
int cluster_no;
```

```
int input_no;
{
  int i,no_member;
  float n,m;

  n = cluster_tbl[cluster_no] [0];
  m = n +1;
  if (testing == 0)
  {
    for (i = 0; i < ninattr; i++)
      b[cluster_no][i] = ((n/m) * b[cluster_no] [i])
        +((1/m) * x[i]);
  }
  no_member = ++(cluster_tbl[cluster_no][0]);
  cluster_tbl[cluster_no][no_member] = input_no;
}

report()
{
int i,j,k, nbcol;

printf("\n\n");
printf("+--------+--------+---------\n");
printf("|     Name_of_Pattgern      \n");
printf("+--------+--------+---------\n");

k=0;
nbcol = 6;
for (i= 0; i < ninput; i++)
{
  if (cluster_tbl[i][0] == 0) break;
  printf("| %-4d | %-3d |", i, cluster_tbl[i][0]);
  for (j = 1; j < ninput+1; j++)
  {
   if ((cluster_tbl[i][j] != 0) || ((i ==0) && (j == 1)))
    {
     if (k > nbcol)
      {
        printf("|\n|      |      |");
```

```
        k = 0;
      }
      printf( " %-3d", cluster_tbl[i][j]);
      k++;
    }
  }
  for (j = k; j <= nbcol; j++) printf("   ");
  printf("|\n");
  printf("----------------------------------\n");
  k = 0;
  }
}

print_bot_up_wts()
{
  int i,j;
  char *file_name = "weights.dat";

  if (( out = fopen(file_name, "w")) == NULL)
  {
    printf("\n Cannot open input data file \n");
    exit(0);
  }
  printf("\n\nBottom_up weights ");
  for (i = 0; i <active_nodes; i++)
  {
  printf("\n Cluster %d\n ",i);
  for (j = 0; j<ninattr; j++)
  {
    printf("%.3f ", b[i][j]);
    fprintf(out,"%.3f ", b[i][j]);
  }
  printf("\n");
  }
  fclose(out);
}
main()
{
int cluster_no, q;
```

```
int i;


read_all_patterns();
first_node();

for (q = 1; q < ninput; q++)
{
 for (i = 0; i < ninattr; i++)
  x[i] = pattern[q][i];
  compute_euc_dist();
  if (( cluster_no + compare_min_ed(q)) >=0)
   update_wts(cluster_no,q);
  else
    form_new_node(q);
  }
  report();
  print_bot_up_wts();
  }
```

```
/*
FILE  expert.c  : Expert system for determining the action
*/

#include <stdio.h>

#define        NINFO   4
#define         NULL   0


struct rule {
        int info1;
        int info2;
        int info3;
        int info4;
        int info5;
```

```
        int action;
        struct rule *next;
        };


typedef struct rule *RULEPTR;


makelist(p,a,b,c,d,e,f) /* only for entering the knowledge into the K.B */
RULEPTR p;
int a,b,c,d,e,f;
{
int i;

p = (RULEPTR) malloc(sizeof (struct rule));
        printf("enter FO  : ");
        scanf("%d", &a);
        printf("enter FF  : ");
        scanf("%d", &b);
        printf("enter MT  : ");
        scanf("%d", &c);
        printf("enter MF  : ");
        scanf("%d", &d);
        printf("enter CF  : ");
        scanf("%d", &e);
        printf("enter ACTION  : ");
        scanf("%d", &f);
        p->info1 = a;
        p->info2 = b;
        p->info3 = c;
        p->info4 = d;
        p->info5 = e;
        p->action = f;
        p->next = p;
}                                        /*end make list */

insafter(p,a,b,c,d,e,f)
RULEPTR p;
int a,b,c,d,e,f;
```

```
{
  RULEPTR q;
      if (p==NULL) {
      printf("void insertion\n");
      exit(1);

                  }

      q=(RULEPTR) malloc(sizeof(struct rule));
      printf("enter FO  : ");
      scanf("%d", &a);
      printf("enter FF  : ");
      scanf("%d", &b);
      printf("enter MT  : ");
      scanf("%d", &c);
      printf("enter MF  : ");
      scanf("%d", &d);
      printf("enter CF  : ");
      scanf("%d", &e);
      printf("enter ACTION  : ");
      scanf("%d", &f);
      q->info1 = a;
      q->info2 = b;
      q->info3 = c;
      q->info4 = d;
      q->info5 = e;
      q->action = f;
      q->next = p->next;
      p->next =q;
}

/* this part of the code is to be used in the controller */


RULEPTR search(list, a,b,c,d,e,f) /* searching the knowledge base */
RULEPTR list;
int a,b,c,d,e,f;

{
```

```
RULEPTR p;
for (p = list; p !=NULL; p = p->next)
  if (( p->info1 == a) &&
     ( p->info2 == b) &&
     ( p->info3 == c) &&
     ( p->info4 == d) &&
     ( p->info5 == e) )
   /*  printf("act : %d", p->action); */
     return (p);
     return (NULL);
     }

testkb() /* knowledge base testing */
{

RULEPTR r;
int NUMRULES, j;
int FO,FF,MT,MF,CF, ACTN,dummy;
printf("enter number of rules : "); /* formation of rule base */
scanf("%d", &NUMRULES);
makelist(r,FO,FF,MT,MF,CF,ACTN);
for(j=0; j<=NUMRULES+1; j++);
   {

        insafter(r,FO,FF,MT,MF,CF,ACTN);
   }
      printf("search\n");
      printf("enter FO  : ");
      scanf("%d", &FO);
      printf("enter FF  : ");
      scanf("%d", &FF);
      printf("enter MT  : ");
      scanf("%d", &MT);
      printf("enter MF  : ");
      scanf("%d", &MF);
      printf("enter CF  : ");
      scanf("%d", &CF);
      if (search(r,FO,FF,MT,MF,CF,dummy)!=NULL)
      printf("success, result is : %d " ,r->action);
```

```
        scanf("%d");
}


SendCommands ()
{
        StartCommands();
        fprintf (DapOut, " idefine a 2          \n");
        fprintf (DapOut, "  set 0 s0            \n");
        fprintf (DapOut, "  set 1 s1            \n");
        fprintf (DapOut, "  set 2 s2            \n");
        fprintf (DapOut, "  set 3 s3            \n");
        fprintf (DapOut, "  set 4 s4            \n");
        fprintf (DapOut, "  set 5 s5            \n");
        fprintf (DapOut, "  set 6 s6            \n");
        fprintf (DapOut, "  set 7 s7            \n");

        fprintf (DapOut, "  time 1000            \n");
        fprintf (DapOut, "  print               \n");
        fprintf (DapOut, "  end                 \n");
        CheckErrors();
}

main () /* main control program */
{
        int i,y1,y2;
        char lname[20];
        FILE *logfile;

        InitDap();


        SendCommands();
        fprintf (DapOut, "start a \n");
        fflush (DapOut);

        for (;;) {
                fscanf (DapIn,"%d %d %d %d %d",&y1,&y2, &y3,&y4,&y5);
                make_list(fo,ff, mt, mf, cf);
```

```
            expert(fo,ff,mt,mf, cf);
            cntrl(motor1, motor2);
            reprot(logfile);
                    }

        fprintf (DapOut, "stop system \n");
        fflush (DapOut);
        LeaveDap();
}
```

APPENDIX 3

ON THE DESIGN OF DEEP HOLE DRILLS WITH EJECTORS

1. Basic Energy Parameters

The schematic diagram of the ejector with the annular nozzle is shown in Figure A3.1. The operating regime of this ejector is characterized by the following parameters:

a) Dimensional Parameters: The most relevant dimensional parameters are: re-circulation flow rate, Q_0; operating flow rate, Q_1; operating hydraulic head, H_1; swarf flow rate past ejector, Q_2; $= Q_0 + Q_1$; swarf head past ejector, H_2 ; cross-sectional area of the ejector nozzle, A_1; cross-sectional area of the mixing chamber, A_2.

b) Non-dimensional Parameters:

- relative head

$$h = H_2/H_2 \qquad \textbf{(A1)}$$

- relative flow rate

$$q = Q_0/Q_1 \qquad \textbf{(A2)}$$

- modulus of the geometrical similarity

$$m = A_2/A_1 \qquad \textbf{(A3)}$$

- hydraulic efficiency

$$\eta_E = q\,\frac{h}{1 - h} \qquad \textbf{(A4)}$$

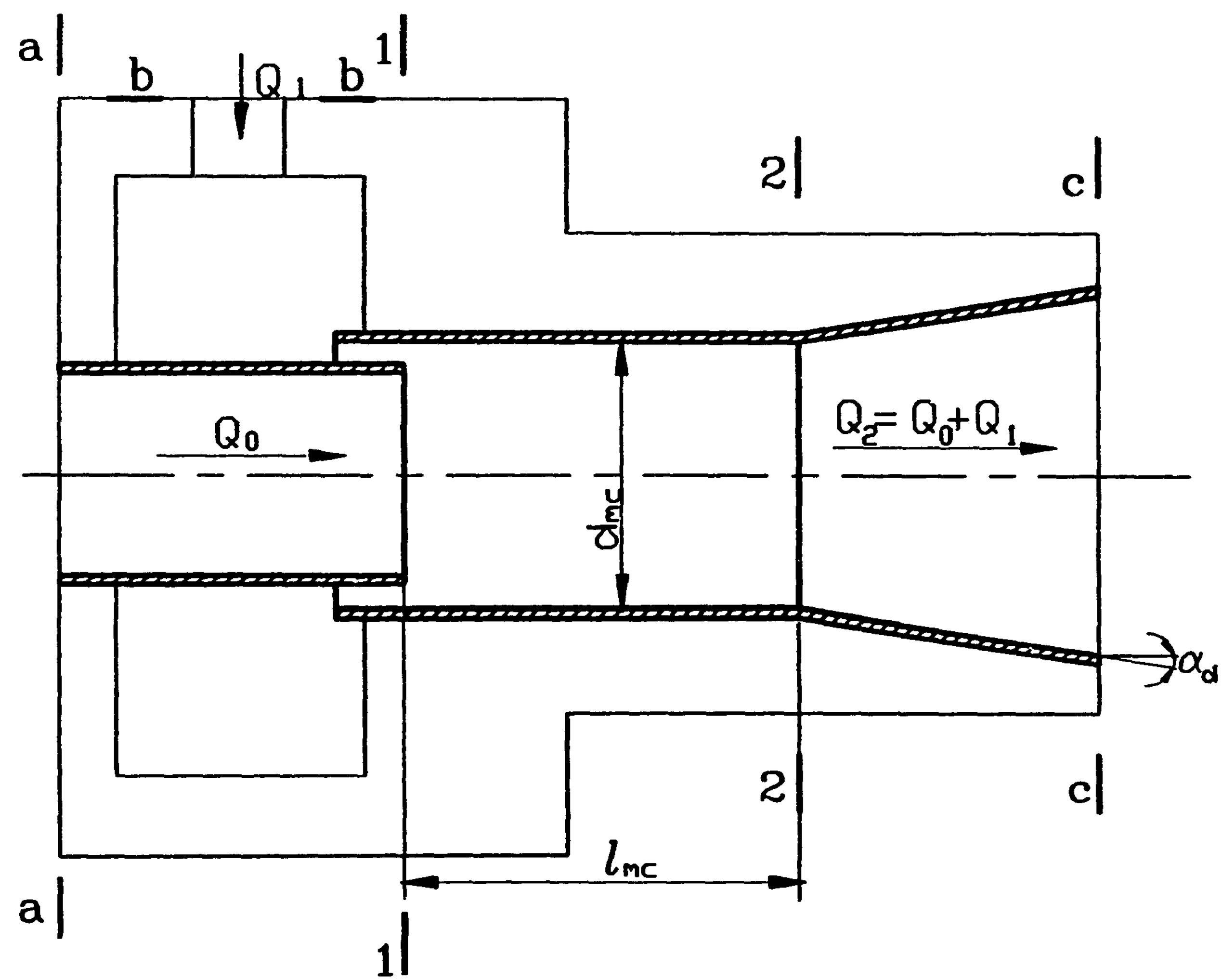


Figure A3.1: Schematic diagram of ejector with annular nozzle

2. Operating Energy Characteristic of the Ejector

The operating energy characteristic of the ejector can be derived by using two different approaches: **a)** the energy analysis and **b)** the impulse theorem [Astakhov, 1990]. Using the energy analysis seems to be reasonable, as it avoids those problems connected with separating the influence of each ejector's design parameters on its energy characteristic [Gnatjuk et al., 1990].

Basic energy relations can be obtained by analyzing the flows of cutting fluid in sections a-a, b-b, c-c, (Figure A3.1):

$$H_2 + \left(\frac{p_c}{\rho_c g} + \frac{\alpha_c v_a^2}{2g}\right) - \left(\frac{p_a}{\rho_c g} + \frac{\alpha_a v_a^2}{2g}\right) \quad \textbf{(A5)}$$

$$H_1 + \left(\frac{p_b}{\rho_c g} + \frac{\alpha_b v_b^2}{2g}\right) - \left(\frac{p_a}{\rho_c g} + \frac{\alpha_a v_a^2}{2g}\right) \quad \textbf{(A6)}$$

Then, the relative head is:

$$h = \frac{H_2}{H_1} = \frac{\left(\frac{p_c}{\rho_c g} + \frac{\alpha_c v_c^2}{2g}\right) - \left(\frac{p_a}{\rho_c g} + \frac{\alpha_a v_a^2}{2g}\right)}{\left(\frac{p_b}{\rho_c g} + \frac{\alpha_b v_b^2}{2g}\right) - \left(\frac{p_a}{\rho_c g} + \frac{\alpha_a v_a^2}{2g}\right)} \quad \textbf{(A7)}$$

Here, α_a, α_b and α_c are velocity co-efficients in sections a-a, b-b and c-c, respectively. The parameters of the flows in sections a-a, b-b, and c-c can be expressed through the parameters of the flow in the beginning of the mixing chamber (section 1-1, Figure A3.1), as follows:

$$\frac{p_b}{\rho_c g} + \frac{\alpha_b v_b^2}{2g} = \frac{p_1}{\rho_c g} + \frac{\zeta_N v_N^2}{2g} \quad \textbf{(A8)}$$

$$\frac{p_a}{\rho_c g} + \frac{\alpha_a v_a^2}{2g} = \frac{p_1}{\rho_c g} + \frac{\alpha_N v_N^2}{2g} + \frac{\zeta_{in} v_N^2}{2g} \quad \textbf{(A9)}$$

$$\frac{p_c}{\rho_c g} + \frac{\alpha_c v_c^2}{2g} = \frac{p_1}{\rho_c g} + \frac{\alpha_1 v_1^2}{2g} - \left(\zeta_{mc} + \zeta_d\right)\frac{v_2^2}{2g} \quad \textbf{(A10)}$$

Substituting Equations (A8) - (A10) to Equation (A7), considering that the fluid velocity at the ejector nozzle, v_N, is much higher than the velocity at section 1-1, v_1 and finally, assuming that the kinetic energy of the cutting fluid flow in the section 1-1 is approximately equal to the kinetic energy of the operating cutting fluid flow [Astakhov and Scorupco, 1982], we arrive at:

$$h = \frac{m^2 - (\zeta_{mc} + \zeta_d)(1 + q)^2 - q^2(1 + m)^2(1 + \zeta_{in})}{(1 + \zeta_N)m^2 - q^2(1 + m)^2(1 + \zeta_{in})} \qquad \textbf{(A11)}$$

This equation represents the operating energy characteristic of the ejector when the energy parameters of the re-circulation and operating flows are independent. This case corresponds to the ejector drill with the "independent" ejector as shown in Figure 3.13 [Petrosjan et al., 1981]. Here, the cutting fluid necessary for the transportation of chips and the cooling and lubricating of the cutting edge and supporting pads is fed to the machining zone through the pressure head and the annular channel between the boring bar and the inner tube. The other cutting fluid flow (from the other pump) is fed to the barb fitting, passes through the ejector nozzle and creates a partial vacuum in the mixing chamber. This causes the cutting fluid from the machining zone to be sucked back along with the chips. As our experience shows, this design is preferable because it saves energy and provides the possibility to regulate the operating regime in the wide limits, depending on the particular tool geometry and workpiece material. Nevertheless, the current market still only offers those ejector drills using the "dependent" ejector design, (Figure A3.1). In such drills, the energy parameters of the re-circulation and operating flow rates are dependent. If we express the flow parameters in the different sections through the flow parameters at the beginning of the mixing chamber (section 1-1, Figure A3.1), we have:

$$\frac{p_a}{\rho_c g} + \frac{\alpha_a v_a^2}{2g} = \frac{p_1}{\rho_c g} + \frac{\alpha_a v_N^2}{2g} - \zeta_\Sigma \frac{v_N^2}{2g} \qquad \textbf{(A12)}$$

$$\frac{p_b}{\rho g} + \frac{\alpha_b v_b^2}{2g} = \frac{p_1}{\rho g} + \frac{\alpha_N v_N^2}{2g} - \zeta_\Sigma \frac{v_N^2}{2g} \qquad \textbf{(A13)}$$

$$\frac{p_c}{\rho_c g} + \frac{\alpha_c v_c^2}{2g} = \frac{p_1}{\rho_c g} + \frac{\alpha_1 v_1^2}{2g} - (\zeta_{mc} + \zeta_\Sigma)\frac{v_1^2}{2g} \qquad \textbf{(A14)}$$

Here ζ_Σ is the total pressure loss co-efficient of the re-circulation flow rate [Suhorucov and Astakhov, 1983].

Expressing velocities v_N, v_2 and v_0, in Equations (A12) to (A14) through the corresponding flow rates and assuming that $\alpha_N = \alpha_* = \alpha_1 = 1$, we obtain:

$$h = \frac{m^2 - (\zeta_{mc} + \zeta_d)(1+q)^2 - q^2(1+m)^2(1+\zeta_{in})}{(1+\zeta_N)m^2 - q^2(1+m)^2(1+\zeta_{in})} \quad \textbf{(A15)}$$

Equations A11 and A15 are the main equations for calculating the ejector's energy characteristics during operation. Our experimental investigations show that the value of $\zeta_{in} = 0.10 - 0.12$ can be accepted in calculations.

3, Selection Of Ejector Nozzle Type

Through the analysis of the different types of ejector nozzles (see Appendix A3-1), we recommend the combined ejector nozzle which contains both the conical and cylindrical parts. The ejector drill with this nozzle is shown in Figure.A3.2, where the selected angle of the conical part (α) is equal to 10° - 15°, based on the following:

1. If α)10°, it leads to an increase in the axial size of the entire ejector and the connection between the tubular parts of the inner tube is weakened.

2. If α)15°, it leads t› increased energy loss in the nozzle and the mixing chamber because the operating flow mixes with the re-circulation flow without primary adjacent velocity.

In order to understand the chips behavior while the re-circulation and operating flows are mixing, the flow structures in the mixing chamber are investigated. The experiments were conducted on different types of ejector nozzles and the best results occur with combined and vortex-combined ejector nozzles. The flow structures of the nozzles in the mixing chamber are shown in Figure A3.3 (a) - (c). Here, it can be seen that the presence of chips in the cutting fluid (in the re-circulation flow) has a significant effect on the flow structure in the mixing chamber. While the chips are present, there appears a re-circulation zone in the

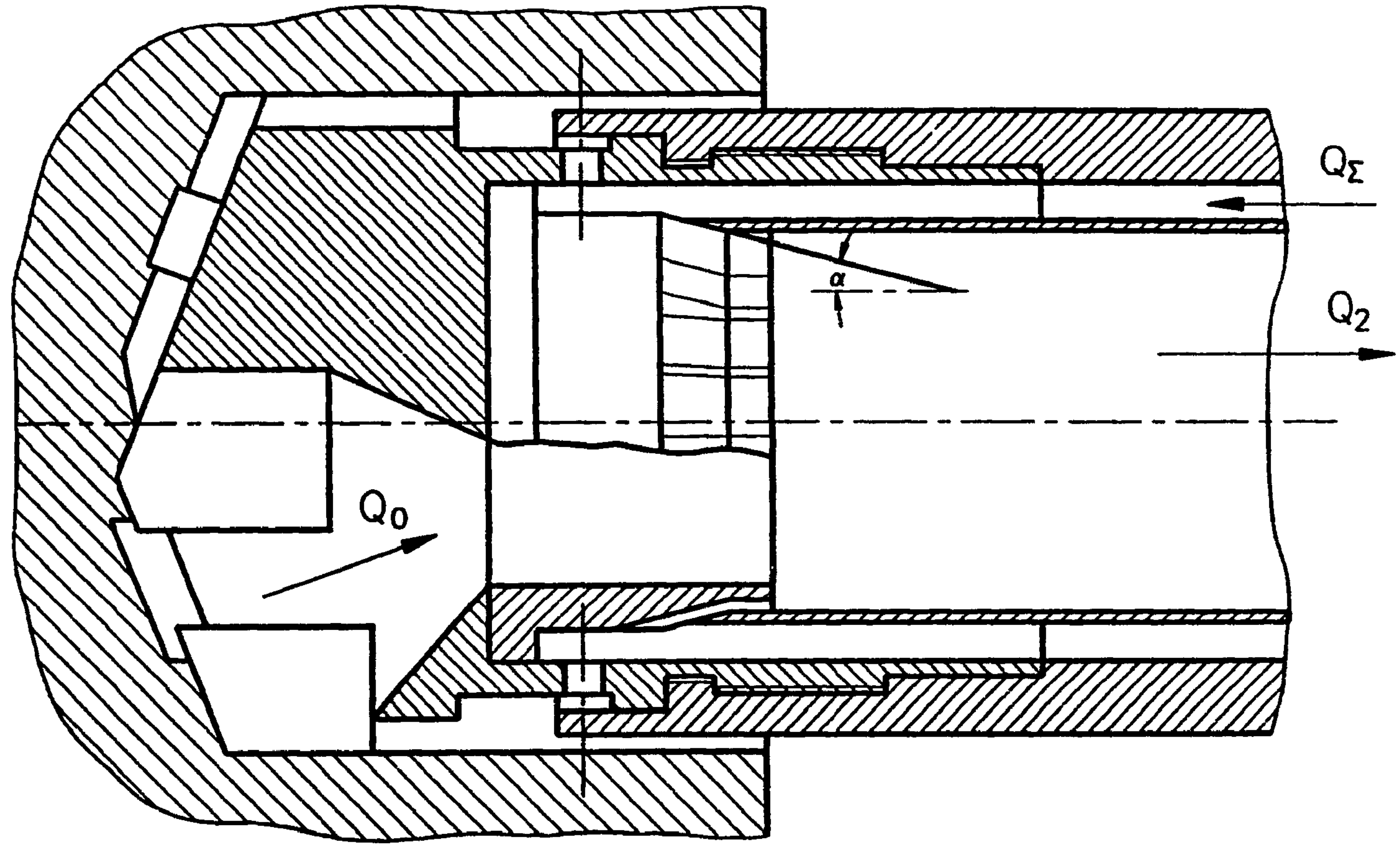


Figure A3.2: Ejector drill with combined nozzle

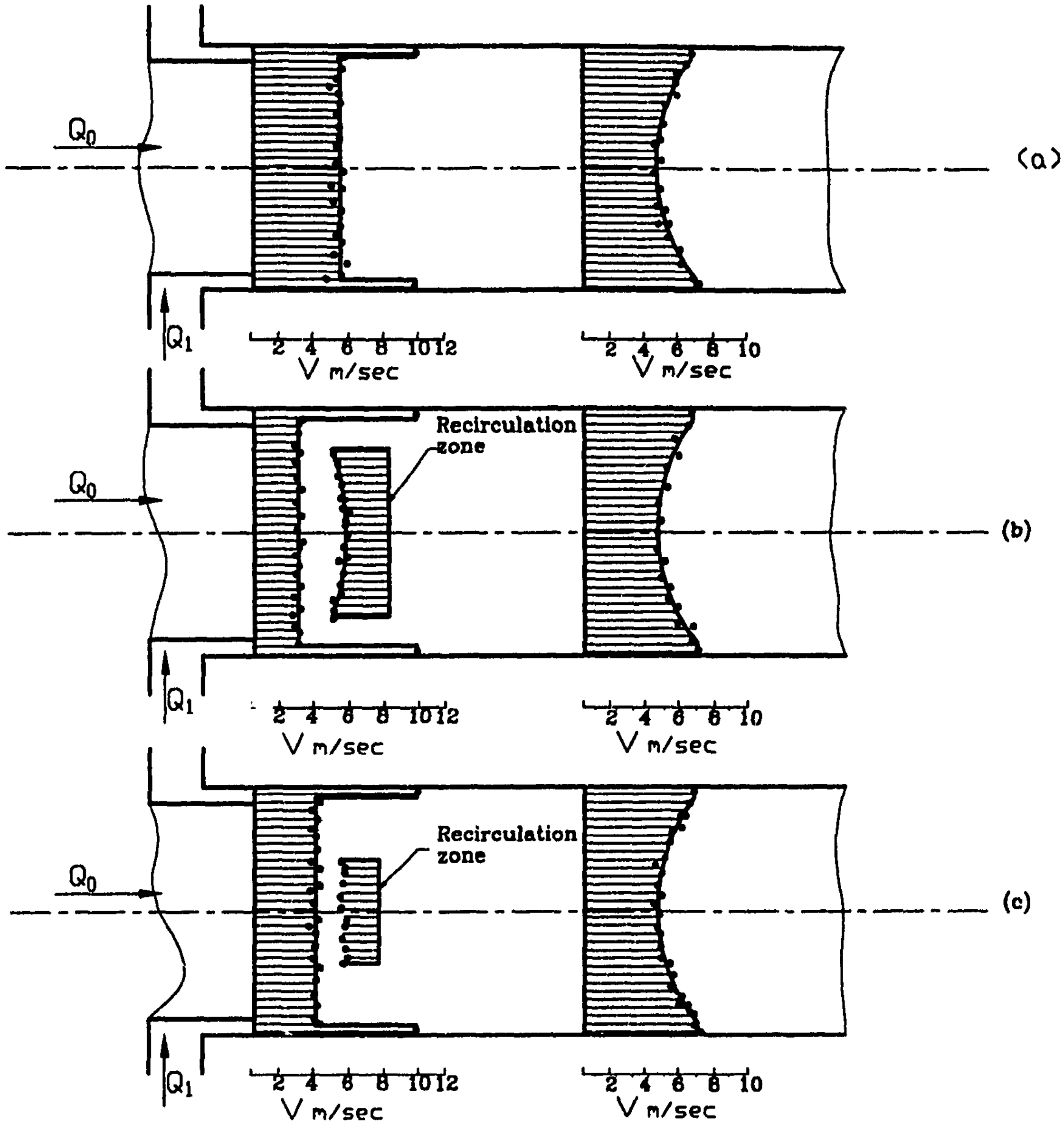


Figure A3.3 : Flow structure in the mixing chamber.

mixing chamber. The flow in this zone is in opposite direction to the re-circulation and operating flows, and contact between the chips from the machining zone and the re-circulation zone take place. Usually when machining medium carbon steel, the chips do not contract and the re-circulation zone does not effect the drilling process. On the other hand, machining hard materials (stainless steels, low carbon alloys, etc.) results in chip contraction, causing the influence of the re-circulation zone to halt the drilling process altogether. This will also usually cause drill head damage if the drilling machine does not have the safety system equipped with a transducer for the chips in the chip removal channels. Therefore, while drilling the hard materials, it is reasonable to use the vortex-combined ejector nozzle. The ejector drill with this nozzle is shown in Figure A3.4. Here, angle α of the conical part of the nozzle, along with the combined nozzle and wing-angle β is selected as equal to 20° - 30°. This choice is based on the following considerations:

1. If $\beta \langle 20°$, then the recalculation zone is growing, reducing the effectiveness of the nozzle.

2. If $\beta \rangle 30°$, then the tangential component of the operating flow velocity is growing, leading to increased energy loss during the mixing of the re-circulation and operating flows.

4. The Length And Diameter Of The Mixing Chamber

The length of the mixing chamber, l_{mc}, (Figure A3.1) is one of the main design parameters of the ejector [Astakhov and Scorupco, 1982]. Experiments were conducted to establish optimum conditions with minimum energy loss in the mixing chamber. These experiments used the special sectional mixing chamber set-up [Astakhov and Scorupco, 1982]. During the initial stages, the unique energy characteristic, h = f(q), became apparent and emphasized the ejector to be dependent of the hydraulic head of the operating flow (Figure A3.5). In the second stage, the section-type mixing chamber was used and the experimental characteristics h =- f(q,m), under different ratio l_{mc}/d_{mc} are obtained (Figure A3.6 (a) -(d)). The optimal length of the mixing chamber was chosen, corresponding to the maximum relative head, h, under constant relative flow rate, q; that is, under maximum energy

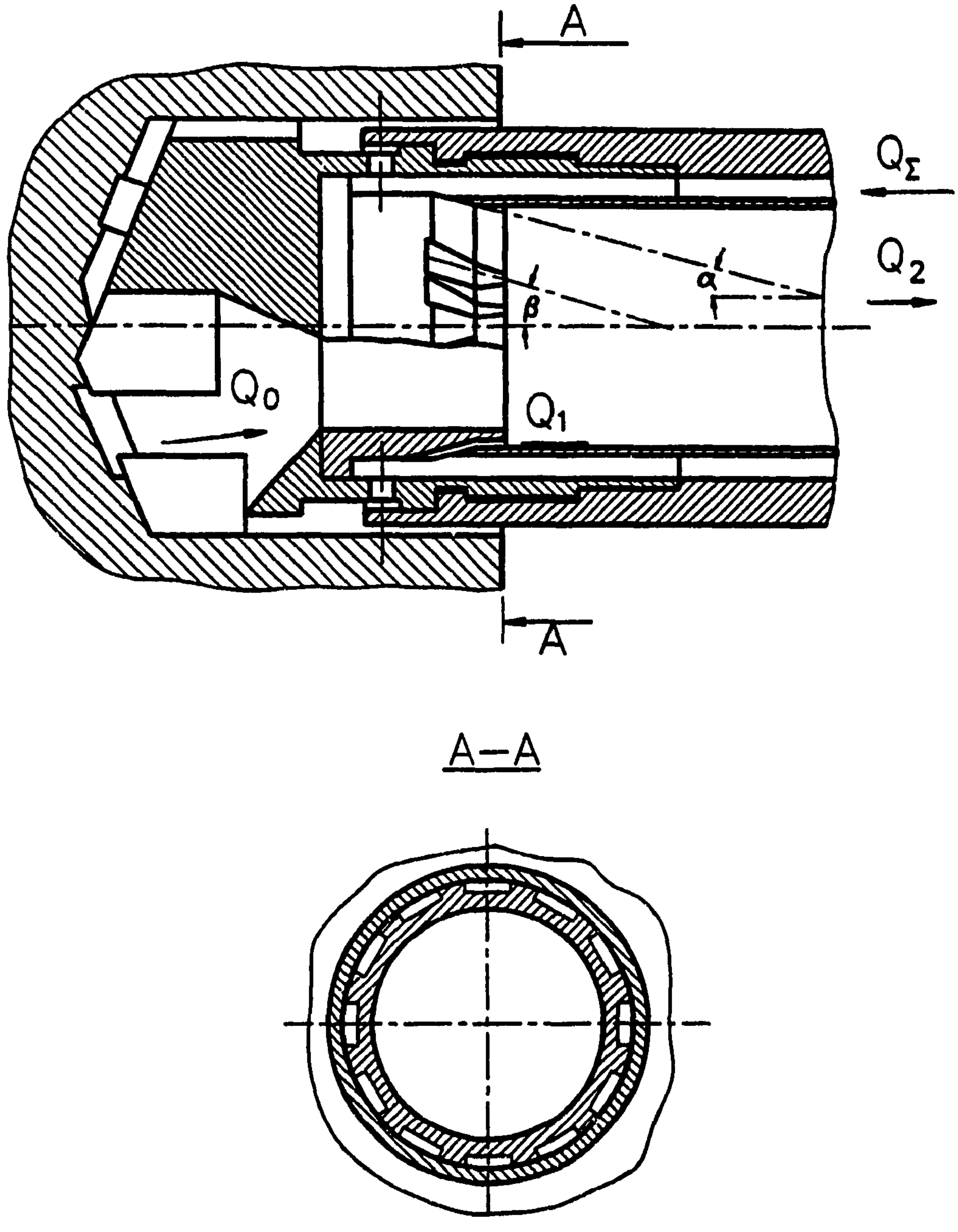


Figure A3.4: Ejector drill with vortex-combined ejector nozzle.

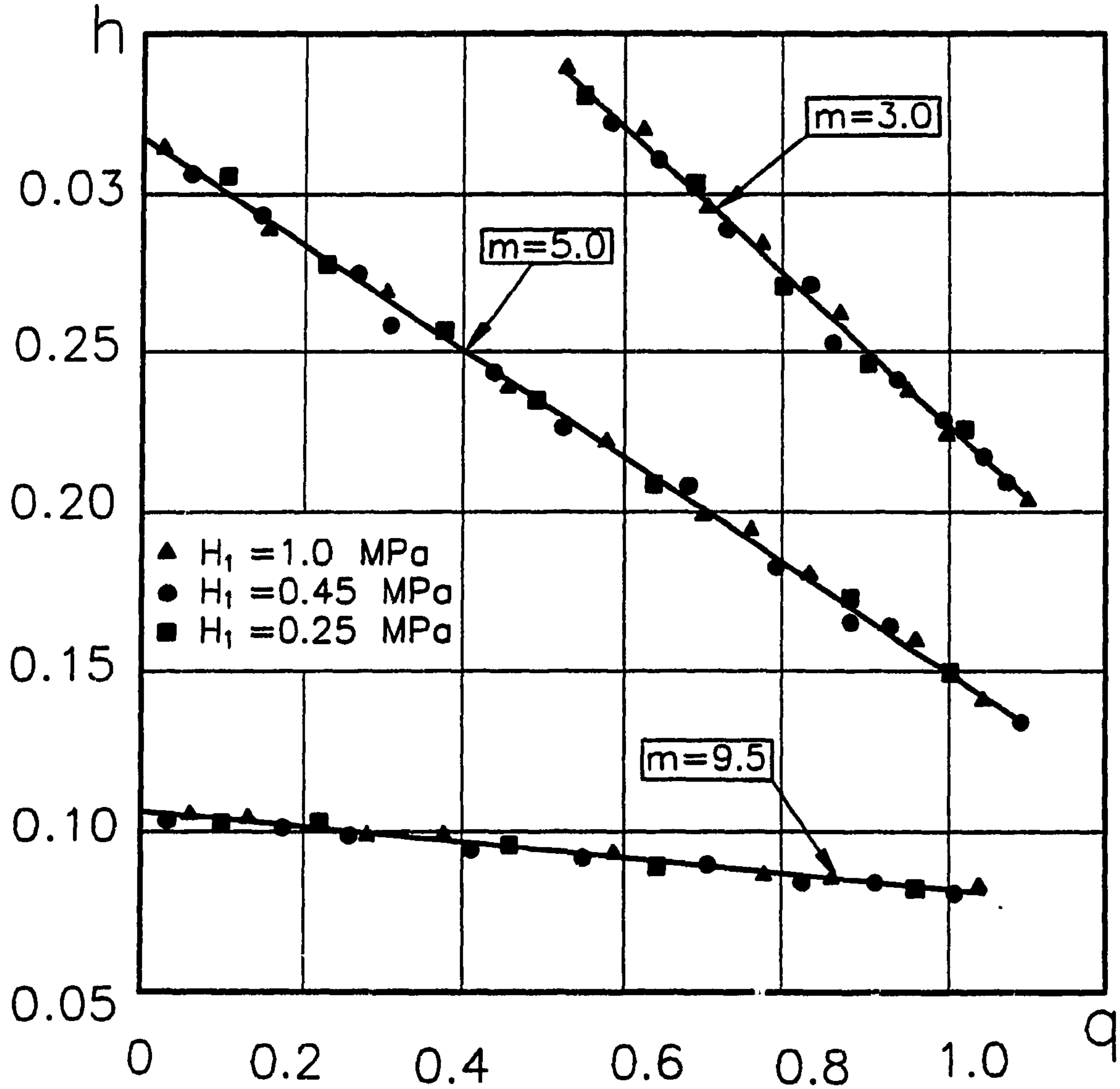


Figure A3.5: Effect of the operating head, H_1, on the characteristic h=f(q).

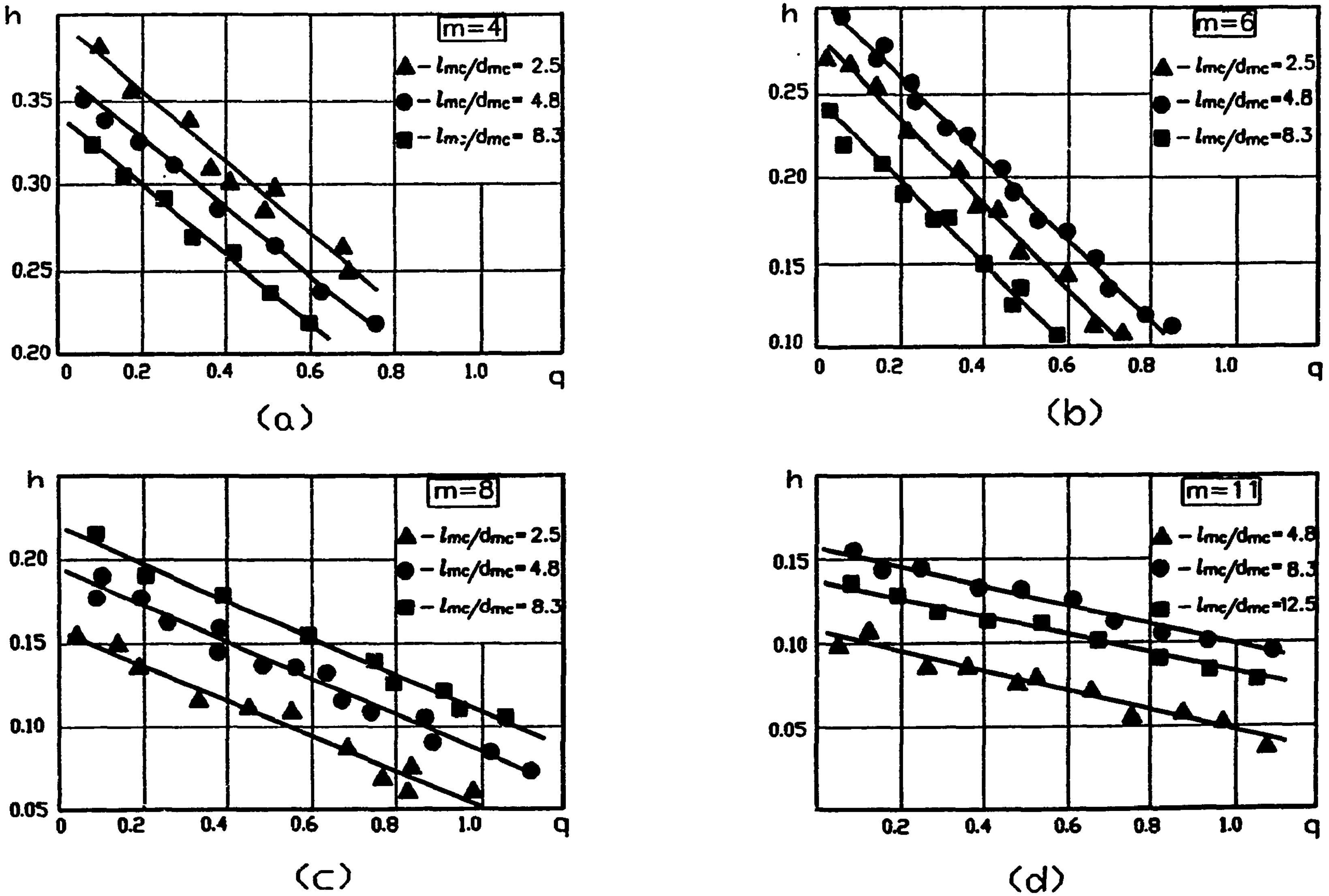


Figure A3.6: Effect of the ratio l_m/d_{mc} on the characteristic h=f(q).

efficiency of the ejector under given modulus m. Using these points, the relationship $(1_{mc}/d_{mc})_{opt}$ = f(m) was obtained (Figure A3.7).

5. Efficiency Of Ejectors With Annular Nozzles - General Energy Characteristics

The efficiency of ejectors with annular nozzles, η_E = f(q,m) is shown in Figure A3.8. It is observed that maximum efficiency is achieved when m = 3.5 -4.0. It should be noted that in commercial ejector drills, the modulus is usually)10. Based upon maximum efficiency, the general energy characteristic h - f(q,m) is obtained (Figure A3.9).

6. Position Of Ejectors In The Hydraulic System Of Drills

Positioning of the ejector nozzle in the hydraulic system of drills plays a vital role in chip removal conditions. Currently, ejector drills are commercially available with two nozzle locations:

1. The nozzle located in the connector. In this case, the ejector has the minimal length of the re-circulation channel, therefore the maximal suction capability. Simultaneously, the operating pressure in not enough to reach optimal operating conditions because: a: pressure is defined by the hydraulic resistance of the machining zone only and b: there is significant pressure loss along the annular channel, between the connector and the drill head.

2. The nozzle located in the drill head. Here, the operating pressure can be high enough. However, this position of the ejector has its own disadvantages: a. the length of the drill is restricted by the suction capability of the ejector nozzle and as a rule, cannot exceed two meters; b. the length of the mixing chamber is less than optimal, reducing the effectiveness of the ejector.

Hence, the positioning of the ejector along the inner tube has be defined through specific hydraulic calculation.

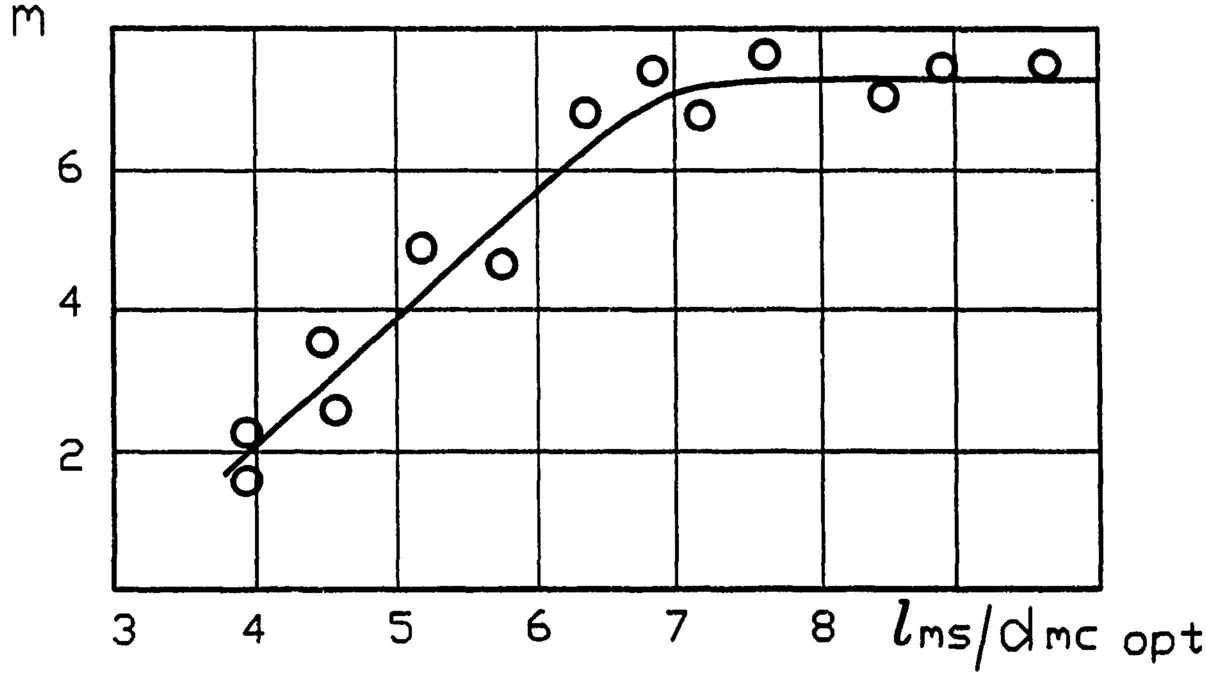


Figure A3.7: Effect of the modulus, m, on the ratio $(l_{ms}/d_{mc})_{opt}$.

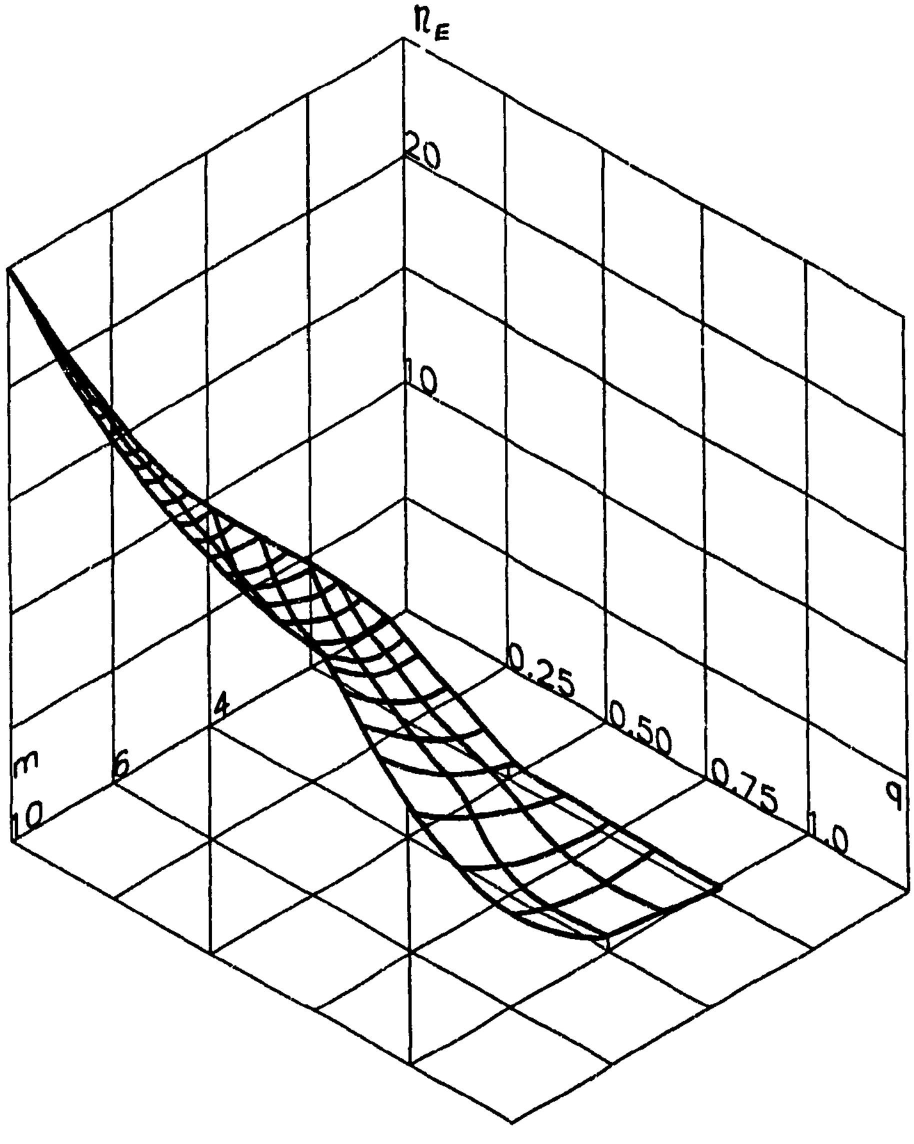


Figure A3.8: Influence of relative flow rate, q, and the modulus, m, on the energy efficiency of the ejector.

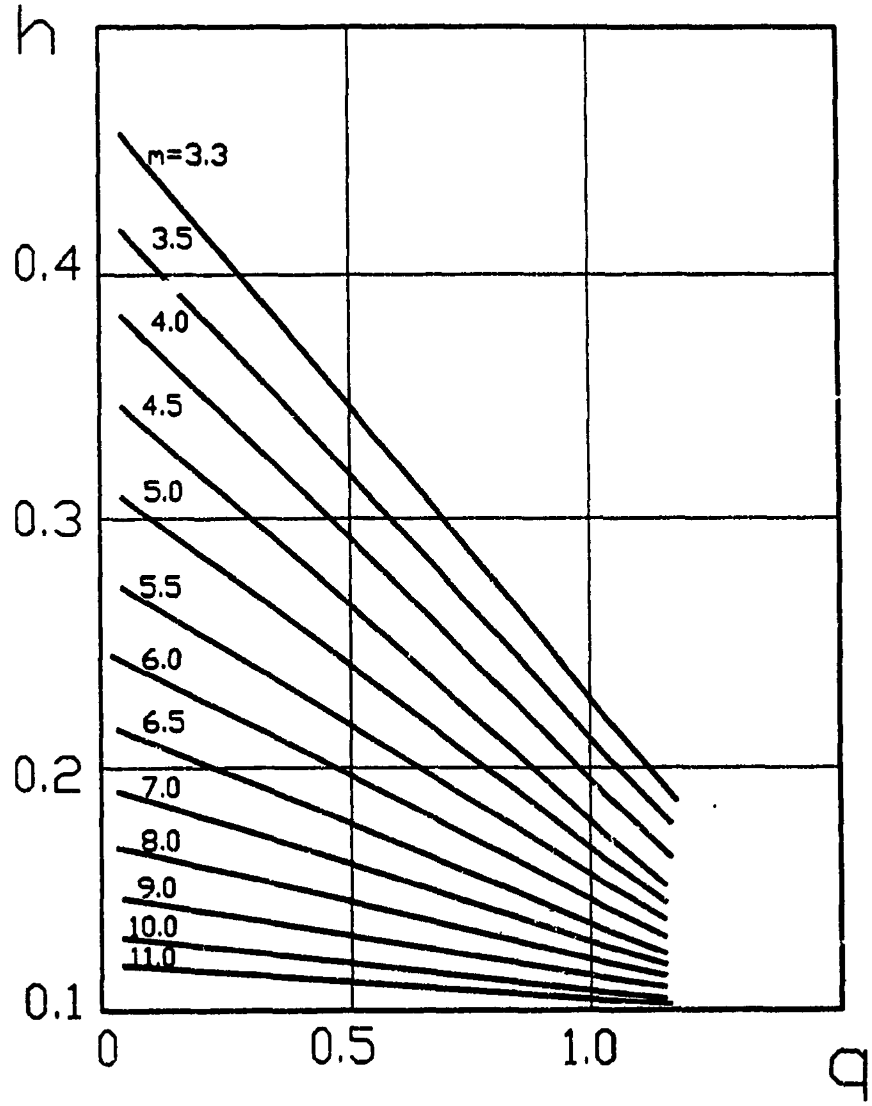


Figure A3.9: General energy characteristics of the ejectors with annular nozzles under maximum hydraulic efficiency.

The schematic diagram of the ejector drill is shown in Figure 3.33. Inlet coolant annular clearance can be classified into three regions: 1) AB is the part from the inlet section (corresponding to Point A), to the ejector nozzle (Point B); 2) BD corresponds to the nozzle, and connects the annular channel (D_b - d_{b2}) to the chip removal channel; 3) BC runs from the ejector nozzle (Point B) to the exit section of the outlet holes of the drill head (Point C). The equations for the hydraulic heads of these regions are:

$$H_{AB} = (Z_B - Z_A) + S_{AB}Q^2 \quad \text{(A16)}$$

$$H_{BC} = (Z_C - Z_B) + S_{BC}Q^2 \quad \text{(A17)}$$

$$H_{BD} = S_{BD}Q^2 \quad \text{(A18)}$$

where z_A, z_B, and z_c are the elevations of points A, B and C, respectively, from datum m; S_{AB}, S_{BC} and S_{BD} are the hydraulic-resistant co-efficients of the above-mentioned parts, s/m^2.

The hydraulic resistance co-efficient for Section AB is:

$$S_{AB} = \left(\zeta_c + \lambda_{ag1} \frac{l_e}{d_{e1}} \right) \frac{1}{12.28 \left(D_b^2 - d_{b1}^2 \right)^2} \quad \text{(A19)}$$

where ζ_c is the pressure loss co-efficient of the connector; λ_{adl} is the friction factor for the annular channel, restricted by diameters D_b and d_{b1}; l_e is the length of the annular channel from the inlet to Point B, m; d_{e1} is the equivalent diameter of the annular channel, m. This diameter can be calculated as follows:

$$d_{e1} = \sqrt{D_b^2 - d_{b1}^2} \quad \text{(A20)}$$

The hydraulic-resistant co-efficient for Section BC is:

$$S_{BC} = \lambda_{ag1} \frac{l_o}{d_{e2}} \frac{1}{128 \left(D_b^2 - d_{b2}^2 \right)^2} + \frac{\zeta_{0h}}{2gA_h^2} \quad \text{(A21)}$$

where λ_{ad1} is the friction factor for the annular clearance, restricted by diameters D_b and d_{b2}; 1_o and d_{e2} are the length and equivalent diameter of this clearance, m; ζ_{0h} is the pressure loss co-efficient of the outlet holes of the drill head; A_h is the total cross-sectional area of these holes.

The hydraulic-resistant co-efficient for Section BD is:

$$S_{BD} = \frac{1}{128\left[\left(d_{b1} - \delta_2\right)^2 - d\frac{2}{b2}\right]^2 \mu} \qquad \text{(A22)}$$

where δ_2 is the thickness of the inner tube (Figure 3.33); μ is the discharge co-efficient of the ejector nozzle.

Sections BC and BD are connected in parallel. The Equations (A17) and (A18) can be summarized by the coolant flow rate:

$$Q_{\Sigma_{BCD}} = \sqrt{\frac{H_{BC} - \left(z_c - z_B\right)}{S_{BC}}} + \sqrt{\frac{H_{BD}}{S_{BD}}} \qquad \text{(A23)}$$

and the hydraulic heads on these sections are $H_{BC} = H_{BD} = H_{\Sigma BCD}$. For horizontal drills $z_c = z_B$ and the flow rate is:

$$Q_{\Sigma_{BCD}}{}^2 = \frac{H_{\Sigma_{BCD}}}{S_{BC}} + 2H_{\Sigma_{BCD}}\sqrt{\frac{1}{S_{BC}S_{BD}}} + \frac{H_{\Sigma_{BCD}}}{S_{BD}} \qquad \text{(A24)}$$

It is obvious that the flow rate $Q_{\Sigma BCD}$ is equal to the inlet flow rate Q_Σ. Then:

$$H_{\Sigma_{BAD}} = \left(\frac{1}{\frac{1}{S_{BC}} + 2\sqrt{\frac{1}{S_{BC}S_{BD}}} + \frac{1}{S_{BD}}} + S_{AB}\right)Q_\Sigma{}^2 \qquad \text{(A25)}$$

Sections AB and Σ_{BCI} are connected in series. Allowing $z_A = z_B$ (horizontal drill location), Equations (A16) and (A25) summarize the hydraulic heads as follows:

$$H_{\Sigma} = \left(\frac{1}{\frac{1}{S_{BC}} + 2\sqrt{\frac{1}{S_{BC}S_{BD}}} + \frac{1}{S_{BD}}} + S_{AB} \right) Q_{\Sigma}^{2} \qquad \textbf{(A26)}$$

Equation (A26) reveals the relationship between the inlet hydraulic head (H_{Σ}) and the flow rate (Q_{Σ}). Equation (A25) defines the hydraulic head at the front end of the ejector nozzle.

The pressure loss co-efficients ζ_c ζ_{oh}, and the friction factor are determined experimentally by using a special method [Astakhov et al., 1983] and a tailored experimental setup [Astakhov and Airikjan, 1983] (Figures 3 .34 & 3. 35).

The equations for the co-efficients of hydraulic resistance for Sections AB and BC (Equations (A19) and (A21)) are written for the most common location of the ejector nozzle, between the connector and the drill head. Using these equations, it is possible to define the position of the ejector in the hydraulic system of a drill and to calculate the inlet flow rate and pressure of the cutting fluid, as shown in Appendix A3-2.

7. Conclusions

Ejectors with annular nozzles can be used to improve the chip removal in deep-hole drills. A comprehensive analysis of ejectors used in deep-hole drills is provided to better understand the role and place of the ejector in tool design. Three non-dimensional characteristics define the operating regime of the ejector: relative head, relative flow rate, and modulus of the geometrical similarity. The energy principle is employed to derive equations for the operating energy characteristic for both independent and dependent ejectors. Analysis of the different types of ejector nozzles provides support to the recommendation for deep-hole drills, the combined (with the angle of the conical part equaling 12°) and vortex-combined (with the angle of the conical part equaling 12° and the wing angle equaling 20° - 30°) ejector nozzles. It was confirmed that the operating energy characteristic for ejectors with annular nozzles is independent from the operating head. The maximum hydraulic efficiency is achieved when the modulus of the geometrical similarity is selected within the range of 3.5 - 4.0 and the relative flow rate is selected between 0.4 - 0.6. Positioning of the ejector nozzle in the hydraulic system of the drills plays a vital role in the chip removal conditions and must be calculated the using the proposed method.

Appendix A3-1

Investigation of the different types of ejector nozzles.

Energy evaluations of the different types of design of ejector nozzles can be done using relation, h = f(q). However, the design evaluation depends on the configuration and the length of the mixing chamber, whether there is a diffuser, etc. Consequently, it is difficult to separate the influence of the ejector nozzle itself. Because of this, a different method of evaluation is used.

The discharge co-efficient of the ejector nozzle can be found using this recognized relation [Engleson, 1952]:

$$\mu = \frac{1}{\sqrt{\alpha_N + \zeta_N}} \tag{A27}$$

assuming uniform inlet nozzle flow $(\alpha_N = 1)$, we have:

$$\mu = \frac{1}{\sqrt{1 + \zeta_N}} \tag{A28}$$

When there are similarities of velocities and pressure fields under different Reynolds numbers, the energy loss in the ejector nozzle is characterized by the Euler number alone [Astakhov, 1984]. This number depends on the geometric characteristic of the nozzle and can be represented as:

$$Eu = \frac{\Delta p}{\rho_c \, v_N^2} \tag{A29}$$

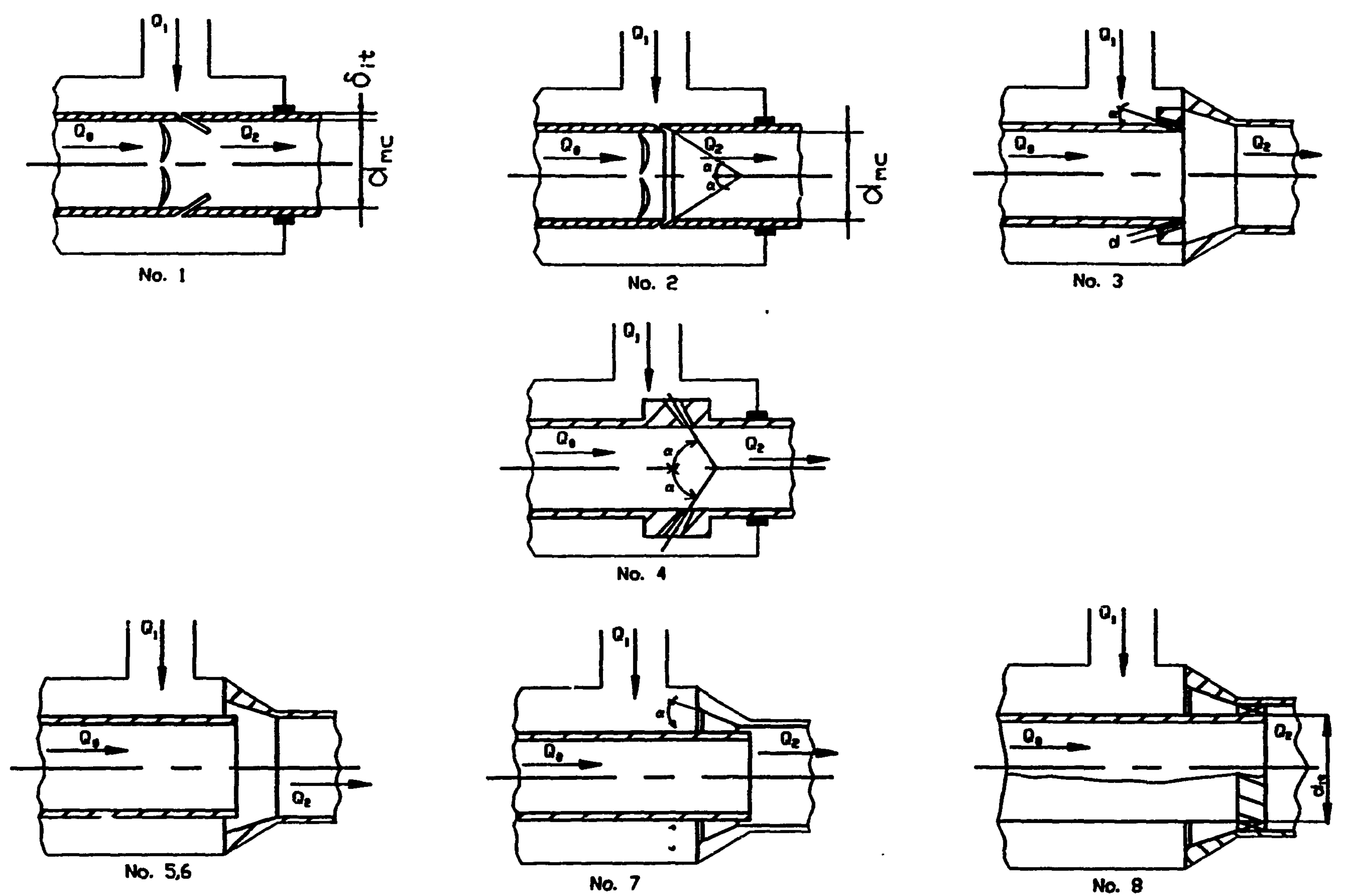


Figure A3.10: Types of ejector nozzles used in the experiments.

Table A3-1. Modulus Of Geometrical Similarity, Pressure-Loss Co-efficient, And Discharge Co-efficient Of The Ejector Nozzles.

No.	Type Of Nozzle	Geometrical Characteristic	ξ_N	μ
1	Venturi	$m = \frac{\pi d_{mc}^2}{4nA_n}$ where n is the number of nozzles; A_n is the nozzle's cross-sectional area.	0.73	0.64
2	Conical at the Inner Tube	$m = \frac{(d_{mc}/2)^2}{\delta_0 \ B}$ $B = \delta_0 \cos^2\alpha + (d_{mc}/2) \cos\alpha$ $B = \delta_0 \cos^2\alpha + (d_{mc}/2)/\tan\alpha$ Here, δ_0 is the nozzle width along the drill axis; α is the nozzle angle.	0.47	0.72
3	Multiple Jet	$m = \frac{d_{mc}^2}{nd_j^2}$ Here, d_j is the diameter of the jet holes.	0.54	0.69
4	Vortex Multiple Jet		0.62	0.67
5	Conical $\alpha = 22.5°$	$m = \frac{\pi d_{mc}^2}{4A_N}$	0.07	0.94
6	Conical $\alpha = 45°$			
7	Combined	$m = \frac{d_{mc}^2}{d_{mc}^2 - d_{it}^2}$	0.12	0.90
8	Vortex combined		0.33	0.78

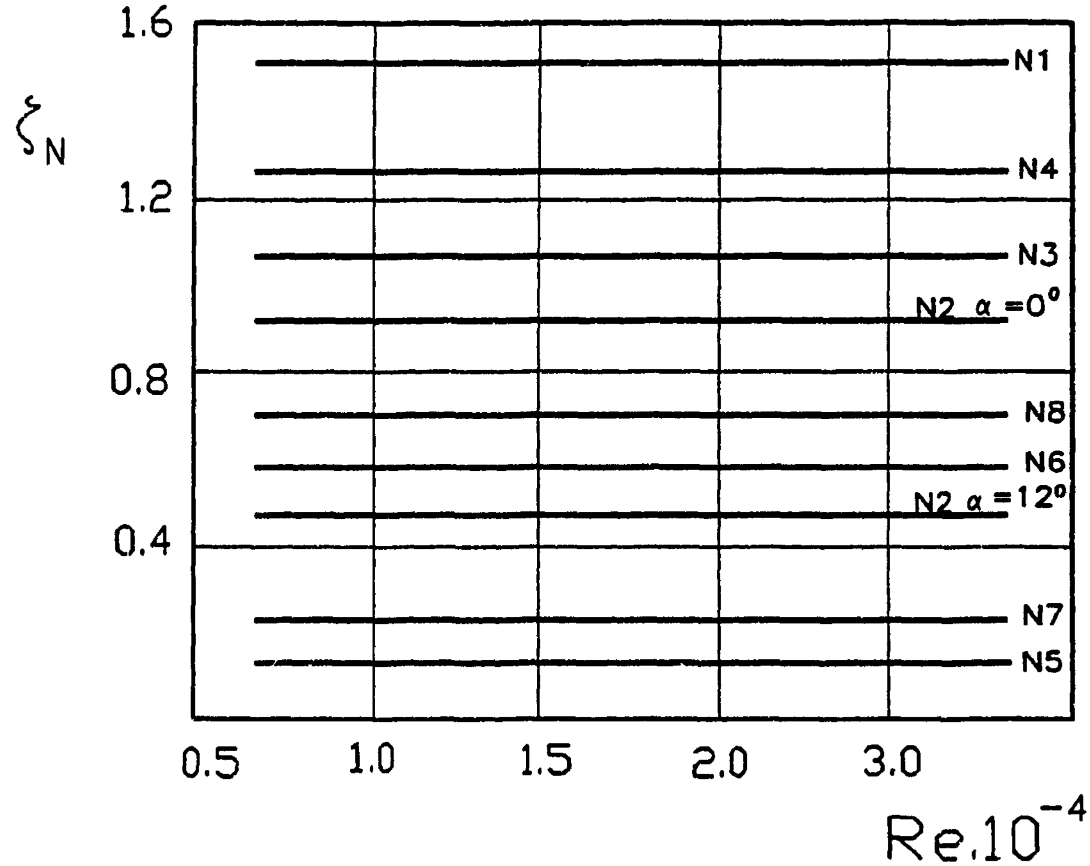


Figure A3.11: Influence of Reynolds number, Re, on the pressure loss coefficient-ζ_n.

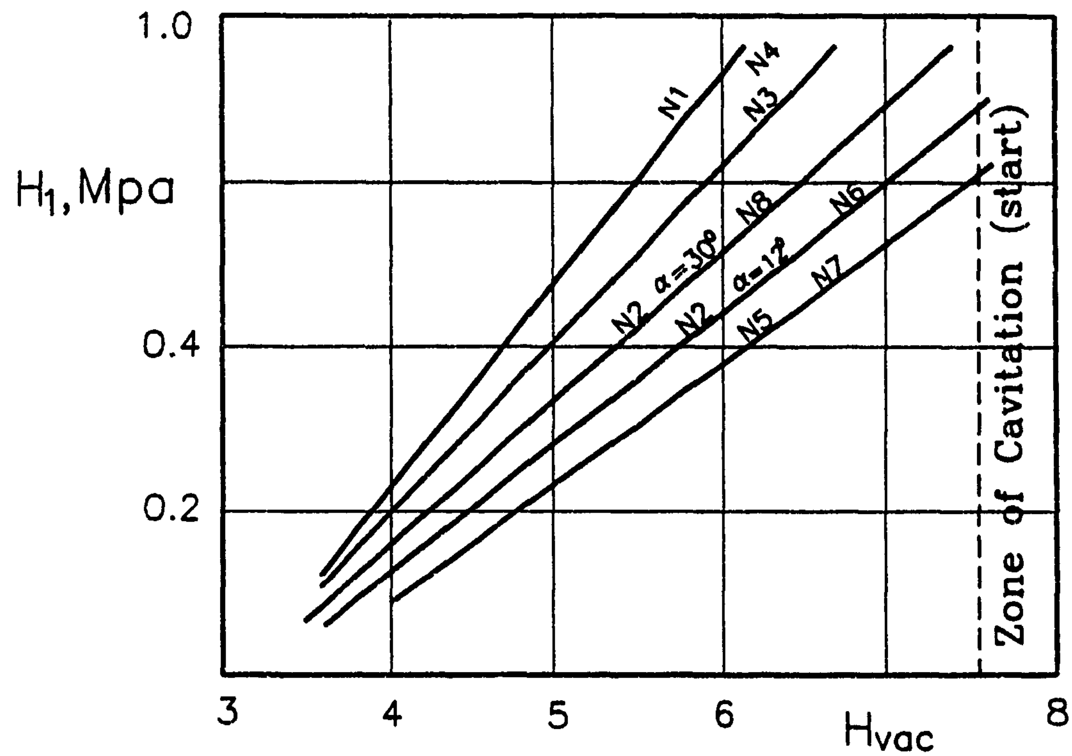


Figure A3.12: Influence of the operating head, H_1, on the vacuum in the mixing chamber, H_{vac}.

Here, Δp and v_N are the pressure drop and the cutting fluid velocity in the nozzle.

The pressure loss co-efficient, ζ_N, is selected as the estimator of the ejector nozzle design and can be expressed through the Euler number as:

$$\zeta_N = 2\ Eu \tag{A30}$$

Substituting Equation (A30) to Equation (A28), we get:

$$\mu = \frac{1}{\sqrt{1 + 2\ Eu}} \tag{A31}$$

Thus, the Euler number can be chosen as the initial characteristic of the nozzle, which defines the pressure loss and discharge co-eefficients.

The experiments were conducted in order to study:

1. The influence of the nozzle's design on its hydraulic resistance.
2. The suction capabilities of the different types of nozzles.

The eight different types of ejector nozzles are identified for study (Figure A3.10, Table A3.1). For each type, the experimental relation ζ_N = f(Re) is obtained (Figure A3.11). It can be seen from this figure that the minimum pressure loss corresponds to nozzle No. 5, with the conical cantor line under α = 22.5°. However, this design is non-technological. Even small error of the axial position of the inner tube leads to a change in the nozzle size, that is a change of modulus m in range of 15-7. Nozzle No. 2, with α = 12° is also characterized by a small pressure-loss co-efficient, but it is difficult to make this nozzle using the thin inner tube. Based upon these considerations, a new design of ejector nozzle is proposed (Figure A3.10, No.7). This nozzle has a rounded, smooth entry, with conical and cylindrical parts. The Euler number for this nozzle (Eu = 0.90) is close to that for nozzle No. 5, (Eu = 0.94).

Efficiency of the ejector nozzle was evaluated according to its suction capability, which was defined by the value of the negative pressure (vacuum) depending upon the inlet head, H_1. The results of the study are shown in Figure A3.12. Analysis of these results shows a correlation between the pressure-loss coefficient and suction capability of the ejector nozzle; that is, nozzles which are characterized by similar pressure-loss co-efficients have approximately the same suction capability. Nozzles Nos. 5 and 7 have the largest suction capabilities. As demonstrated in Figure A3.12, the suction capability increases with an increase in the inlet head. The restriction factor for suction capability is cavitation in the mixing chamber, which occurs when the value of the vacuum exceeds 0.078 MPa. As indicated in the experiments, this value does not depend on the design parameters of the ejector nozzle or the ejector. The value of vacuum created by the ejector nozzle for a given operating head can be found in Figure A3.13, and this value is one of the most important energy characteristics of the ejector. The most common error is that this value is considered when defining the position of the ejector nozzle in relation to the machining zone. The correct positioning of the ejector nozzle can be verified by the following inequality:

$$H_{vac} \geq \Delta h_s \tag{A32}$$

Here, Δh_s is the head loss of the cutting fluid-chip mixture, swarf (including the elevation of the nozzle position, relative to the machining zone), along the distance between the machining zone and the ejector nozzle.

Losses due to change of location, past the mixing chamber, can be reduced by installing a diffuser between the mixing chamber and the other part of the chip removal channel (Figure 3.33). The co-efficient of the hydraulic loss in diffusion is determined by the non-dimensional characteristics h= f(q, α_d). Under a small diffuser angle, α_d (up to 8°) and ratio l_d/d_{mc} = 2 - 4, the hydraulic resistance co-efficient in the diffuser is ζ_d = 0.5 - 0.7 (lower values correspond to the smaller ratios l_d/d_{mc}) for a wide range of ejector parameters.

In order to define the energy loss in the mixing chamber, the equation for energy balance was used. The nozzles Nos. 1, 7, and 8 were selected for this investigation and the following results for the pressure-loss co-efficients were obtained:

$$\zeta_{mc1} = 1.497 \cdot m^{1.407} \cdot q^{0.157 \cdot \log m - 0.661} \tag{A33}$$

$$\zeta_{mc7} = 0.0241 \cdot m^{2.451} \cdot q^{-1.661} \tag{A34}$$

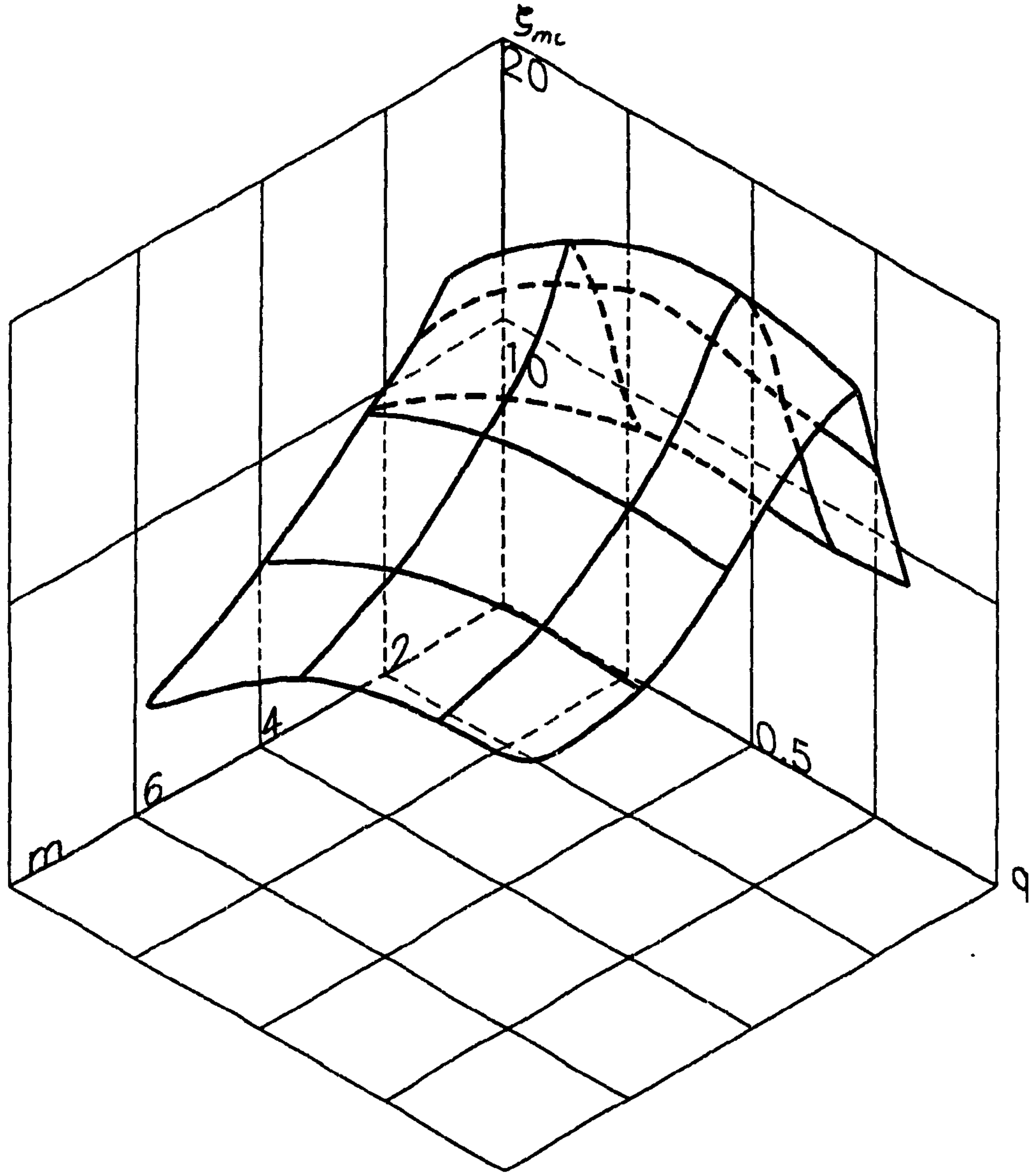


Figure A3.13: Effect of modulus m, and relative flow rate, q, on the pressure loss coefficient ζ_{mc} for nozzle No. 7.

$$\zeta_{mc8} = 0.0138 \cdot m^{2.331} \cdot q^{0.079 \cdot \log m + 1.468} \quad \textbf{(A35)}$$

The relationship $\zeta_{mc} = f(q, m)$ for nozzle No. 7 is shown in Figure A3.13. It can be seen that an increase in the modulus m leads to increased energy loss in the mixing chamber. Due to an increase in the value of m, the velocities of re-circulation and operating flows increase, resulting in energy loss.

Therefore, the experimental comparison of the different types of ejector nozzles shows that the combined ejector nozzle (nozzle No. 7) is the most efficient among the samples used.

Appendix A3-2

Calculation of the ejector position in the hydraulic system of the ejector drill.

Calculation of the ejector position can be carried out as follows:

1. Calculate the critical cutting fluid velocity, V_{cr}, as the velocity corresponding to the beginning of reliable chip transportation [Astakhov, V.P., 1989].

2. Calculate the re-circulation flow rate as:

$$Q_0 = V_{cr} \cdot A_{in} \quad \textbf{(A36)}$$

3. Calculate the cross-sectional area of the ejector nozzle, using the selected modulus m = 3.5 - 4.0, then calculate S_{BD} (Equation (A22)).

4. Choose the relative flow rate, corresponding to the maximum efficiency as q = 0.4 - 0.6 (Figure A3.8).

5. Calculate the operating flow rate, Q_1 as:

$$Q_1 = Q_0 \cdot q \quad \textbf{(A37)}$$

6. Calculate the required operating head (Equation (A18) when $Q = Q_1$).

7. Calculate the inlet flow rate as:

$$Q_\Sigma = Q_0 + Q_1 \tag{A38}$$

8. Calculate the hydraulic-resistance co-efficient S_{BC} from Equation (A23).

9. Calculate the ejector position (length l_0) from Equation (A21).

10. Verify, using Equation (A32).

11. Calculate the relative head, h, by using Equation (A11) for the independent ejector design or Equation (A15) for the dependent ejector design. This calculated value of h can then be compared to the maximum value of h under given conditions (Figure A3.9). The difference between the two values shows how far the chosen regime is from the optimal.

12. Calculate the head, H_2 (Equation A1).

13. Verify the condition:

$$H_2 > = \Sigma h \tag{A39}$$

where Σh is the total hydraulic head loss.

If inequality (A32) holds, the calculations are complete. Otherwise, one must reduce the relative flow rate in accordance with Figure A3.9 in order to achieve the required value of h and therefore, H_2.