The Development of a Computer Assisted Planning and Monitoring Framework for the Milling Process

by

Fu-Wing Raymond Yeung
B.A.Sc (Eng. Phys.), The University of British Columbia, 1996

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF APPLIED SCIENCE in

THE FACULTY OF GRADUATE STUDIES
MECHANICAL ENGINEERING

We accept this thesis as conforming to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA
September 1998

© F. W. Raymond Yeung, 1998
In presenting this thesis in partial fulfillment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Mechanical Engineering
The University of British Columbia
2324 Main Mall,
Vancouver, BC
V6T 1Z4, Canada

Date: ...OCTOBER 14TH, 1998
Abstract

The objective of this work is to examine new strategies to improve the quality of the machining operation during its preprocessing and real time stages. In particular, an integrated CAD/CAPP/CAM system and a distributed process monitoring system are developed. In the preprocessing stage, the integrated CAD/CAPP/CAM system attempts to apply the most economical cutting conditions to practical $2^{1/2}$-D milling operations; the approach retains all advantages of both APT NC programming and interactive, easy-to-use 3-D part modeling systems. In the real time stage, the distributed process monitoring system takes advantage of the existing networking and sensor technologies to provide real time machining process information. The information is made available to on-site machine operators and also to off-site manufacturing and production engineers who can use the information to "virtually" supervise the progress of the operations.

The work presented in this thesis represents another step towards the realization of the Computer Integrated Manufacturing concept in a production environment.
Table of Contents

Abstract ii

Table of Contents iii

List of Figures v

List of Tables vii

Acknowledgments viii

1. Introduction 1
   1.1. Objective 3
   1.2. Outline 5

2. Fundamentals of Computer Integrated Manufacturing 7
   2.1. Introduction 7
   2.2. Computer-Aided Process Planning 8
      2.2.1. Introduction 8
      2.2.2. The Variant Approach 11
      2.2.3. The Generative Approach 13
   2.3. Machining Economics 18
      2.3.1. Introduction 18
      2.3.2. Formulation of the Economic Model for Milling 20
      2.3.3. Tool Life in Milling 22
      2.3.4. Economics of Single Pass Milling 25
      2.3.5. Economics of Multi-Pass Milling 28
      2.3.6. Operation Sequencing 30
   2.4. Computer-Aided Manufacturing 34
      2.4.1. Introduction 34
      2.4.2. NC Part Programming 34
         2.4.2.1. APT-Like Languages 35
         2.4.2.2. Interactive CAD/CAM 38
      2.4.3. Computer Numerical Controls 40
   2.5. Process Monitoring and Control 42
   2.6. Computer Networking for Manufacturing 44
   2.7. Summary 50

3. Integration of CAD/CAPP/CAM 52
   3.1. Introduction 52
   3.2. Software Selections 53
   3.3. Choice of Data Exchange Format 57
   3.4. Approximation of Non-Orthogonal Part Geometry 61
   3.5. Software Development for CAD/CAPP/CAM Integration 71
## List of Figures

| Figure 2-1 | The variant approach to process planning | 12 |
| Figure 2-2 | End Milling Geometry | 20 |
| Figure 2-3 | The influence of torque and power constraints on the relationship between width and cost | 30 |
| Figure 2-4 | An example part | 32 |
| Figure 2-5 | A Behind-Tape-Reader DNC system | 46 |
| Figure 3-1 | LINE in DXF format | 57 |
| Figure 3-2 | CIRCLE in DXF format | 58 |
| Figure 3-3 | An example part | 62 |
| Figure 3-4 | Two versions of approximation of the example part | 63 |
| Figure 3-5 | Operation sequences for both versions of approximation of the example part | 65 |
| Figure 3-6 | Non-orthogonal partitioning of the example part | 66 |
| Figure 3-7 | Another example part with non-orthogonal entities | 68 |
| Figure 3-8 | An example part with a semi-circular entity | 69 |
| Figure 3-9 | An example part with a circular entity | 70 |
| Figure 3-10 | Overall system architecture | 74 |
| Figure 3-11 | The example of an approximated milling part | 76 |
| Figure 3-12 | Part geometry bounded at each depth | 77 |
| Figure 3-13 | Legends for Table 3 | 78 |
| Figure 3-14 | Elemental volumes at depth 1 in the example part | 80 |
| Figure 3-15 | Interlocking features | 82 |
| Figure 3-16 | Cutting sequence for the example part | 87 |
| Figure 3-17 | Tool sequence for the example part | 87 |
| Figure 3-18 | Overlapped top surfaces of each combined volume | 89 |
| Figure 3-19 | Combined volumes for the sample part | 91 |
| Figure 3-20 | The tool paths for (a) uni-directional, (b) staircase, and (c) spiral cutting | 95 |
| Figure 3-21 | The volume boundaries and partitioning lines for full slotting | 97 |
| Figure 3-22 | Cusps left during a staircase cutting operation | 97 |
| Figure 3-23 | Experimental part geometry | 100 |
| Figure 3-24 | AutoCAD pull-down menu - "Sectioning" | 100 |
| Figure 3-25 | Cross sectional planes for the experimental part | 101 |
| Figure 3-26 | Approximated cross sectional planes for the experimental part | 102 |
| Figure 3-27 | Part name defined in a dialogue box | 102 |
| Figure 3-28 | AutoCAD pull-down menu - "Feature Extraction" | 103 |
| Figure 3-29 | NLI-type statements for the experimental part W | 103 |
| Figure 3-30 | Cutting sequence and tool selections for the part W | 105 |
| Figure 3-31 | Top boundaries of the combined volume of the part W | 106 |
| Figure 3-32 | Tool paths for the combined volumes of the part W | 108 |
| Figure 4-1 | Object data structure | 120 |
| Figure 4-2 | Virtual Machining Environment (VME) | 122 |
| Figure 4-3 | Overall system network architecture | 125 |
| Figure 4-4 | Operator Console for "Emergency Stop" | 129 |
| Figure 4-5 | Experimental set-up | 131 |
| Figure 4-6 | Tool path for case study (I) | 135 |
| Figure 4-7 | Snapshots of the tool movement in case study (I) | 136 |
| Figure 4-8 | Linear contouring operation | 137 |
| Figure 4-9 | Tool path for case study (ii) | 138 |
| Figure 4-10 | Snapshots of the tool movement in case study (II) | 139 |
| Figure 4-11 | Multi-pass peripheral milling operation | 140 |
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Decision Hierarchy in Process Planning</td>
<td>9</td>
</tr>
<tr>
<td>Table 2</td>
<td>Compound volumes and their costs</td>
<td>32</td>
</tr>
<tr>
<td>Table 3</td>
<td>Decision logic table</td>
<td>79</td>
</tr>
<tr>
<td>Table 4</td>
<td>States of all the elemental volumes of the example part</td>
<td>81</td>
</tr>
<tr>
<td>Table 5</td>
<td>NLI-type commands for the example part</td>
<td>85</td>
</tr>
<tr>
<td>Table 6</td>
<td>States of the faces of the combined volumes for the sample part</td>
<td>91</td>
</tr>
<tr>
<td>Table 7</td>
<td>Openness testing for one sample combined volume</td>
<td>92</td>
</tr>
<tr>
<td>Table 8</td>
<td>Machining database</td>
<td>93</td>
</tr>
<tr>
<td>Table 9</td>
<td>X, Y, and Z boundary-related information of the experimental part</td>
<td>102</td>
</tr>
<tr>
<td>Table 10</td>
<td>Types of the combined volumes for the part W</td>
<td>104</td>
</tr>
<tr>
<td>Table 11</td>
<td>Summary of feed and speed selections for the part W</td>
<td>108</td>
</tr>
<tr>
<td>Table 12</td>
<td>Tool data</td>
<td>161</td>
</tr>
<tr>
<td>Table 13</td>
<td>Tool life data</td>
<td>161</td>
</tr>
<tr>
<td>Table 14</td>
<td>Machine data</td>
<td>161</td>
</tr>
<tr>
<td>Table 15</td>
<td>Material data</td>
<td>161</td>
</tr>
</tbody>
</table>
Acknowledgments

I would like to express my sincere gratitude and appreciation to my supervisor, Dr. Ian Yellowley for his support (both technically and financially), guidance, and encouragement throughout the course of this thesis.

I would like to recognize and thank Dr. LongXiang Yang and Dr. Rudi Seethaler for their assistance with laboratory equipment. I would also like to thank them, Henry Reiser, and Thomas Fung for making the work environment very pleasant.

Finally, I would like to thank my family for their support and encouragement throughout my life, and Lucinda, for her infinite patience and moral support especially during difficult times.
Chapter 1

Introduction

1. Introduction

In the metal cutting industry, the process of producing a useful component spans a wide spectrum of activities ranging from the design of the component, to the generation of the detailed process plan and the manufacture of the component. Both the design and manufacturing stages of the production process have been computerized for many years. The design stage has been revolutionized by the emergence of Computer-Aided Design (CAD) technology. The conceptual design of the component is entered interactively using devices such as a digitizer in an interactive computer graphics system. The design data is stored in the computer databases and is readily retrieved for other downstream applications. The applications of Computer-Aided Manufacturing (CAM) technology in the manufacturing stage include the automatic generation of machining instructions (G-codes) by computer programs, and the control of machine tool operations by means of Computer Numerical Control (CNC). The use of interactive graphics to aid NC part programming has been accomplished through the integration of CAD and CAM systems. The integrated CAD/CAM approach attempts to take advantage of the computer’s huge memory capacity, fast processing speed, and user-friendly interactive graphics capabilities to automate and tie together these separate production tasks, thus reducing the time and cost of product development. Computer-Aided Process Planning (CAPP) can be considered as a part of CAD/CAM, however the present
generation of integrated CAD/CAM systems support the process planning functions rather poorly; usually the process planner has to select cutting tools, machining conditions, and operation sequences according to his/her past experience and handbook data. Since much progress has been made in the derivation of sound physical and economic models of the machining processes, it becomes possible to perform process planning activities in a more cost-effective fashion. Part of the work presented in this thesis demonstrates the benefits of embedding machining knowledge into the process of CAD/CAM integration. Such integration will result in the automatic generation of optimized low level process plans and tool paths based on digitized design data. The successful implementation of such an integrated CAD/CAPP/CAM system will no doubt represent another step towards the realization of the Computer Integrated Manufacturing (CIM) concept in a production environment.

The successful realization of CIM should allow efficient sharing of data and resources and the flow of information and work throughout the manufacturing facility. With the availability of fast and reliable networking technologies, each computer no longer operates by itself in a production environment. In a typical manufacturing facility, many computers are connected together by cables to form a factory-wide, computer network in an attempt to provide various types of office-type services such as printer sharing, remote file retrieval, and so on. The existing office-type network can in fact be extended to allow other computer-controlled devices such as machine tools to become part
of the factory-wide network. Traditionally, only a limited amount of shop floor information is available to the machine operator who controls a single machining operation through the operator interface. The interface only conveys basic information such as tool position and feed rates, spindle speeds, and tool being used. With recent advances in sensor technology, different types of real time process information can be captured at the shop floor level by suitable sensors. This information can be displayed on the operator interface to provide the on-site machine operator with the various types of process information, and at the same time distributed to the off-site computers connected to the factory-wide network. These pieces of real time information can be then accessible to the factory personnel who do not work on the shop floor and who have different types of responsibilities. For instance, the manufacturing engineer can keep track of the progress of the on-going machining operations as if he/she were on the shop floor, thus enabling him/her to assign new jobs to the machines that are ready for more work. The tool engineer who has access to the information on tool wear can keep track of the tool condition, and can therefore plan the tool-replacement schedule or place new tool orders in advance, thus reducing the machine down-time.

1.1. Objective

The objective of this thesis is to investigate new strategies which can help improve the quality of the machining operation during its preprocessing and real time stages. In particular, an integrated CAD/CAPP/CAM system and a
distributed process monitoring system are to be developed. The development of the prototype integrated CAD/CAPP/CAM system serves to demonstrate the feasibility and benefits of embedding the intelligence of the machining process into a commercially available CAD/CAM system. This prototype system is only capable of dealing with \(2^{1/2}\)-D milling part, and should be able to automate most of the activities ranging from the extraction and interpretation of the part's design data, to the generation of the optimized low level process plan and NC part program which is ready to be used by the machine tool to manufacture the part.

The distributed process monitoring system developed in this thesis should allow the sharing of the real time process information collected at the factory floor level by both on-site machine operators and off-site production personnel in a computer network environment. In particular, the prototype system is to be implemented using 3-D virtual reality technology, and is expected to be able to demonstrate the feasibility of distributing both the graphical animation of the machining processes and the cutting force measurements across the network in a real time fashion. The prototype system should also provide some kind of remote control mechanism such as remote process shutdown so that the staff can respond to any abnormality detected during the machining operations while they are "watching" the processes remotely via the network.
1.2. Outline

Since the work presented in this thesis mainly deals with the development of the applications which contribute in different degrees to the goal of realizing a truly computer integrated manufacturing (CIM) environment, an entire chapter (chapter two) is devoted to the review of the relevant literature discussing various aspects related to CIM. The discussion will be focused on the metal cutting process only. The topics which will be covered in detail include computer-aided process planning, computer-aided manufacturing, process control and monitoring, and computer networking for manufacturing. The economics of the machining process which forms the basis for the formulation of a good solution to the low level planning problem will also be presented in the same chapter.

Chapter three will present the development of an integrated CAD/CAPP/CAM system. The system was designed for $2\frac{1}{2}$-D milling operations only. It consists of commercially available CAD and NC part programming systems and an in-house developed low level process planning system. It accepts the part design data represented by a 3-D CAD model and generates NC part programs which are in turn transferred to the machine tools for manufacturing the part. Most of the activities involving the extraction of the design data and the generation of both the low level process plans and part programs are performed automatically. This system extends the functionality of the previous system presented by Seethaler [1] by allowing non-orthogonal part
geometry to be dealt with by the same process planning package which accepts prismatic geometry only. The chapter then concludes with some remarks about the contribution of this system to the realization of dynamic process planning.

Chapter four will present the development of a network-based real time machining process monitoring system. The system was implemented using 3-D virtual reality technology and was originally developed to run in a local area network (LAN), although it is potentially capable of operating in a wide area network (WAN) or an inter-networking environment. It allows the distribution of both the real time graphical animation of the machining process(es) and the maximum cutting force measurements collected by a force sensor during the machining operation(s) across the local area network. It also enables remote shutdown of the process by authorized personnel via the network connection. The majority of this section concentrates on the explanation of the internal structure of the monitoring system. Two examples are presented to demonstrate the system's capabilities. The chapter then concludes with some remarks about the possibility of allowing near-real-time process monitoring via internet and the security issues involved with such the system and the measures taken by the prototype system in an effort to prevent unauthorized access to the system when it is operating.
2. Fundamentals of Computer Integrated Manufacturing

2.1. Introduction

Computer integrated manufacturing (CIM) attempts to integrate all aspects of manufacturing from the initial engineering design and process planning to machine operation and utilization, process and quality control, and tracking and maintenance [2]. Within a CIM-equipped manufacturing facility information such as process plans, machine control codes and process feedback are readily accessible to manufacturing engineers. Also, the system generally extends to materials management, the availability of detailed part scheduling information allowing production personnel to make more intelligent decisions on inventory management. In order to maximize the economic benefits of a CIM system, the most advanced technology should be readily adopted for each of its components. The areas which have attracted most research effort include computer-aided process planning and manufacturing, process control and monitoring, and computer networking for manufacturing. A brief discussion of recent research efforts in these areas is presented in the following sections. For the purposes of this thesis, the discussion is primarily focused on the metal cutting process.
2.2. Computer-Aided Process Planning

2.2.1. Introduction

In the metal cutting industry, the goal of process planning is to determine a set of detailed work instructions required to manufacture a component. The input to the process planning system would typically be an engineering drawing or the CAD (Computer-Aided Design) representation of the component. The drawing or CAD database would normally provide information about the geometry of the component, material of the workpiece, tolerance and surface finish constraints, and other special instructions which may be necessary in creating the process plan. The output should include the sequence of processing, selection of machine tools, fixture and tooling arrangement, tool path division, selection of cutting conditions, etc. It should become apparent that the tasks involved in creating a process plan are hierarchical in nature. The typical tasks which have to be covered in most cases are shown in Table 1. The decision making activities involved in process planning should take into account the part design (e.g. the complexity of the design, etc.), manufacturing constraints (e.g. the size of the workpiece, the achievable tolerances, finish, and material removal rate, etc.), and economic considerations (e.g. the cost of the machines, tool cost, etc.). It should be remembered that, because of the hierarchical nature of the planning problem, it is possible that infeasibility be encountered during the decision making process as it proceeds from its highest to lowest decision levels; thus, backtracking is usually necessary.
Traditionally, the process plan has been prepared manually by an engineer or a process planner based on his/her own discretion and handbook data. The process plans prepared by different engineers would inevitably be different even for the same component. To minimize the manufacturing cost, manufacturing companies strive to create the most optimized process plan such that the machining process can be performed in the most cost-effective way. However, it would indeed be very difficult and inefficient for them to allocate a vast amount of human resource to deal with the process optimization problem which usually involves a tremendous amount of complex computations, not to mention the fact that the lack of accurate process data that are needed during process optimization leads to the generation of sub-optimal process plans.
Furthermore, as the engineers who are familiar with the details of the machining process are gradually retiring, fewer qualified personnel will be available in the future to do process planning. Undoubtedly, a more efficient way of conducting process planning is necessary, and the Computer-Aided Process Planning (CAPP) approach seems to provide the answer.

A computer-aided process planning (CAPP) system implements the process planning functions by computer. It has been widely recognized that CAPP represents the missing link between the design and manufacturing stages of product development within an integrated CAD/CAM system. Besides performing typical process planning activities, the successful implementation of a CAPP system should provide interfacing capabilities with both CAD and CAM systems. The provision of interfaces to other production-related software packages such as the Material Requirements Planning (MRP) system should also be considered. Some of the benefits derived from computer-aided process planning include:

- reduction of overall process planning time and cost
- reduction of manufacturing cost through optimization
- more standardized, accurate process plans
- less skilled process planners required

Over the last two decades or so, many attempts have been made towards the development of so called automated process planning systems. In general,
automation of process planning can be carried out using two different approaches: variant and generative. In the following sections, the characteristics and merits of each approach of creating process plans are discussed.

2.2.2. The Variant Approach

A variant or retrieval approach to computer-aided process planning can be seen as a tool to assist in manual process planning activities. In this approach, a process plan for a new part is created by identifying and retrieving an existing plan for a similar part from an established database, and making necessary modifications for the new part. In most variant CAPP systems, the development of such a database is based on the principles of group technology (GT) and parts classification and coding. In the context of GT, parts are grouped into a number of part families according to their attributes which may include basic geometry, part function, manufacturing methods, machine tools, fixture types, and so forth [3]. Each part group or family has to be encoded into some type of the code structure. Typically, the coding scheme for parts classification involves using a sequence of numerical digits to identify the part's attributes. The coding scheme is sometimes enterprise-dependent, which means a custom designed coding scheme may be necessary to suit the specific needs of a company.

Figure 2-1 shows a flow diagram which illustrates how a variant CAPP system works. To create a process plan for a new part, the system generates a
GT part code for the new part based on the adopted coding scheme. With this part code, a search of the part family database determines if a generic process plan exists for the part family to which the new part belongs. If the process plan for the new part exists, it is retrieved and edited by the user as necessary. If the database does not contain a process plan for the given part code, the user may either search the database for the similar part codes for which a generic process plan exists, or create a new plan from ground up. In either case, the newly generated process plan is saved in the database and becomes the standard plan for the new part family.

Figure 2-1 The variant approach to process planning

In comparison with manually performed process planning, the variant approach increases the manufacturing information management capabilities by providing an efficient way for organizing and storing complete plans. It offers tremendous savings in planning times by allowing the editing of an existing process plan retrieved from the GT-oriented database. Since one only has to
modify an existing plan for the part with similar attributes, the majority of the
effort involved in the decision making process is saved. There have been a
number of successful implementations of this technology, and among them are
the CAPP from CAM-I [4, 6] and MIPLAN from OIR [5, 6].

The most noticeable disadvantage of adopting the variant approach to
process planning is that there is a danger of repeating mistakes from earlier
plans that were used to build up the original database. Also, the optimality of the
process plan which is created or modified for each new part still relies on the
human process planner's expertise and experience. Furthermore, the variant
approach seems to become impractical in cases where small batches of widely
varying parts are being produced. In these circumstances, the substantial task
involved in the initial development of the database may not be justified.

2.2.3. The Generative Approach

Generative CAPP systems represent an alternative approach to
automated process planning. Instead of retrieving and editing an existing
process plan stored in a GT-based database, a generative CAPP system
attempts to create an optimized process plan for a new component based on
mathematical models of manufacturing processes without human intervention
and reference to old standard plans. The input to this type of the process
planning system would typically be a CAD database which contains information
about the geometry of the component and the product quality requirements in
terms of tolerances, surface finish, etc. Other component-independent information such as machine tools' capabilities and cutting tool data is stored in the database which is ready to be referenced. Once the CAD representation of the part has been received, the process planning system is ready to apply the manufacturing knowledge which has been "implanted" into the system and make decisions on a number of process planning functions such as those listed in Table 1.

In comparison with the variant approach, the biggest advantage of the generative approach is that the scientific approach to process planning eliminates any inconsistency and inaccuracy which sometimes characterize the plans created by the variant systems. The availability of powerful, inexpensive computer systems allows more efficient and economical implementation of the generative CAPP system which usually demands a tremendous amount of computer resource to perform many complex computations. Also, the generative approach to process planning fits perfectly in situations where small batches of widely varying parts are being produced. Over the last two decades, there have been a number of implementations which use the generative approach to process planning and which enjoy various levels of success. Among them are TIPPS [6, 7], GARI [6, 8], and the work by Fisher [9].

Both TIPPS and GARI adopt an "expert system" approach to process planning. These systems attempt to translate human-like expertise of process
planning into a number of "IF...THEN..." type production rules. The conditions represented by the IF statement of each rule may bear on the part to be made, the available machine tools, and/or the plan to be produced. The THEN statement gives the advice which should be applied to the current plan when the conditions in the IF statement are satisfied. Process capability (e.g. the properties of the available machines/cutting tools, the allowable sizes of the workpiece, etc.) varies among manufacturing facilities, and therefore these production rules should be modified to adapt to the different manufacturing environments or cultures. By comparing the conditions represented by the IF statements with the design specification of the part, the selection of machining processes, machine tools, and the sequence are made. However, both systems lack the ability to provide optimized solutions to the low level process planning problem. The selection of machining parameters such as feeds and speeds, for instance, are chosen based on handbook data.

While the so-called "expert systems" mentioned in the previous paragraph have shown some promise in the area of higher level decision making, Fisher's work in his thesis mainly concentrated on finding the optimal solution to the low level process planning problem. His system was primarily designed for peripheral milling operations. It attempts to generate optimal cutting parameters (e.g. feeds and speeds) and path division for multi-pass operations based on complex mathematical models of the milling process. The
methodologies on which the implementation of his system was based are presented in Chapter 2.3.

Generative process planning systems are considered as being in their early stages of development and use, and there are still many challenges which have yet to be overcome. Because the methodology required to develop a generative CAPP system is far beyond that required to develop a variant system, generative process planning systems are not yet very common. Nowadays, there are very few generative CAPP systems which are commercially available (e.g. PART by Tecnomatix Machining Automation) and they are usually very expensive. Both long development time and huge capital investment have prevented small to mid-size companies from developing their own generative CAPP systems. The problem of interfacing CAPP with CAD still requires a lot of work. Extracting necessary geometrical and technological information of the part from the data structure inherent in the traditional CAD systems for process planning purposes presents a great challenge. In fact, there are very few commercially available CAD systems which can be used for manufacturing purposes, and many of them do not even support data other than those of geometrical nature. The methodology called feature recognition has been explored by many researchers in an attempt to extract and interpret the geometry and topology associated with the part from its solid model representation and some degree of success has been reported [10, 11]. The optimality of process plans generated by generative process planning systems
depends on the accuracy of process parameters which characterize the models that describe the manufacturing processes; in most practical situations, however, only an approximation of these parameters is available during the preprocessing stage. As a result, most CAPP systems tend to be conservative in the selection of cutting parameters, which lead to the generation of more safe, but less optimal process plans.
2.3. Machining Economics

2.3.1. Introduction

The generation of an optimal low level process plan for machining operations requires optimization at various levels. At the lowest level, cutting parameters such as feed and speed are selected. Also, path subdivision strategies which involve the selection of radial depth in turning and the selection of radial width and axial depth in milling are formulated. Constraints such as tool wear, machine limits, chatter and tool breakage limits must be observed in the selection processes. At the highest level the operation sequencing strategy which aims to obtain an optimal cut volume removal sequence is determined. The outcome of the analysis of this hierarchy is a low level process plan which should aim to minimize the overall machining cost.

At the lowest level of the hierarchy decisions may be made based on a sound economic model of the machining process. A lot of research effort has been expended in an attempt to derive models on which the selection of optimal machining parameters may be based. In general, there are two approaches which are proposed to accomplish this goal. In the first approach, researchers use a mathematical model to represent the process and use non-linear numerical optimization techniques to calculate economically optimal machining parameters [12]. This approach has been applied in both single pass and multi-pass machining operations [13, 14, 15]. The success of such technique relies
heavily on the accuracy of the process parameters required by the optimization technique. The second approach to the selection of machining parameters tries to avoid the need to create an accurate model of the process and to bypass the vigorous computation involved in the optimization procedure. In this approach, a set of heuristic rules is created to find the required machining parameters to achieve a good solution to the optimization problem [16]. This approach, though not guaranteeing a truly optimal solution, is robust enough to be applied in a real machining environment where an accurate process model is not usually available [17].

At the highest level of the hierarchy, the operation sequence is selected to minimize the overall machining cost. A sequencing problem arises when a number of removal volumes or features intersect with each other and more than one machining operation has to be performed. One should appreciate that the computational cost involved in solving this problem increases rapidly as the number of removal volumes increase. A number of researchers have attempted to solve the problem using different approaches. Kishinimani and Saito [18] suggested a dynamic programming approach to finding an optimal solution. This approach examines the cost of all possible combinations of removal volumes. Despite the savings which occur due to the nature of this technique it is still inefficient as a result of the need to calculate costs for all feasible volumes. A heuristic approach to this problem has been proposed by Yellowley and Kusiak [19] and was further improved by Yellowley and Fisher [20]. The advantage of
the latest version of this approach is that it does not require one to examine the cost of all possible volume combinations alternatives. Further detail of this approach is presented in a later section.

In a realistic machining environment where process parameters can never be identified accurately, it is believed that a more computationally efficient solution to the optimization problem should perform at least equally well if not better compared with a more traditional approach which seeks a "true" optimal. The following sections outline the approach taken in this thesis towards the generation of an optimal low level process plan; the approach is directed towards end milling operations.

2.3.2. Formulation of the Economic Model for Milling

In this section an economic model for optimizing feed and speed in an end milling operation (see Figure 2-2) is formulated.

![End Milling Geometry](image)
Chapter 2  
*Fundamentals of Computer Integrated Manufacturing*

The symbols shown in Figure 2-2 are described here:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>depth of cut</td>
</tr>
<tr>
<td>R</td>
<td>tool radius</td>
</tr>
<tr>
<td>S</td>
<td>feed rate</td>
</tr>
<tr>
<td>d</td>
<td>width of cut</td>
</tr>
<tr>
<td>(\phi_s)</td>
<td>swept angle of cut</td>
</tr>
<tr>
<td>l</td>
<td>tool length</td>
</tr>
</tbody>
</table>

In general, when considering the economics of the process, one of the following two objectives may be pursued: minimization of either machining cost or machining time. Each objective will usually lead to different solutions to the optimization problem. In both cases, an economic model on which the optimization is based is required. In a milling operation, the total machining cost consists of fixed and variable components. The fixed component includes the time spent in tool manipulation, machine set-up, and workpiece load and unload. The variable component includes the amount of time spent on both the actual machining operations and tool changes. Since the fixed part of the economic model is not directly related to the parameters which are to be optimized, it is not included in the optimization procedure. The variable part of the model is directly influenced by the choice of machining parameters and is formulated as:

\[
t_v = \frac{1}{\text{RPM} \times S} + \frac{t_s}{\text{RPM} \times S \times T}
\]  

(2.1)

where

- \(t_v\)  variable time of machining a unit length of material [min/in]
- \(t_s\)  time to replace a worn tool [min]
- \(S\) feed per revolution [in/rev]
- \(\text{RPM}\) spindle speed [rev/min]
- \(T\) tool life [min]
By incorporating the machine overhead rate and the tool cost, the variable machining cost is then expressed as:

\[ C_v = \frac{x}{\text{RPM} \times S} + \frac{xt_s + C_t}{\text{RPM} \times S \times T} \]  

(2.2)

where

- \( C_v \) variable cost of machining a unit length of material [$/in]
- \( x \) machine cost rate [$/min]
- \( C_t \) tool cost [$]

It is more usual to consider the minimum cost criterion (eq. 2.2) because of the high overhead rate of modern CNC machines. It is also worth noting that the tool life plays an important role in both optimization strategies. Further detail of its implications on the selection of optimal machining conditions is presented in the following section.

### 2.3.3. Tool Life in Milling

The life of a cutting tool depends on various machining conditions, e.g. work/tool materials, feed, speed, width of cut, and depth of cut. The tool life is measured by the period of time after which the amount of flank and crater wear exceeds an allowable limit. The earliest equation describing the relationship between the tool life and the machining conditions was proposed by F. W. Taylor in 1907, in simplest form of the Taylor equation only considers velocity and is of the form:

\[ VT^\alpha = \text{Const.} \]  

(2.3)
where \( \alpha \) is a constant. Many researchers have developed their own tool life equations to include the effect of other machining conditions but many of these are similar to Taylor's equation. Yellowley and Barrow [21] proposed the following tool life equation which includes the effects of depth of cut, time in/out of cut, feed rate, cutter immersion, and thermal fatigue parameter:

\[
T = \frac{360}{\phi_s} \times \frac{\text{Const}}{X^n S_{eq}^{1/m} V^p a^q}
\]  

(2.4)

where
- \( X \) thermal fatigue parameter
- \( S_{eq} \) equivalent chip thickness
- \( n, m, p, q \) constants
- \( V \) peripheral cutting velocity

This equation is valid when there is no chip sticking occurring in the end milling operation and when the lag angle between the leading and trailing edges of the cutter is small. Furthermore, this equation does not take into account the effects of tool entry and exit conditions.

The term related to the feed rate in the tool life equation is \( S_{eq} \). \( S_{eq} \) is called the equivalent feed and is defined as the constant value of feed which will result in the same wear rate as the varying feed in milling. It is used in equation (2.4) to account for the discontinuous nature of the chip and the sinusoidal variation of the chip thickness in the actual milling process. Assuming a simple Taylor-like relationship between feed and tool life, the equivalent feed per tooth or equivalent chip thickness is expressed analytically as follows:
\[ S_{eq} = \left\{ \frac{1}{\phi_s} \times \int_{0}^{\phi_s} (S_{t} \sin \theta)^{1/m} \, d\theta \right\}^m \]  \hspace{1cm} (2.5)

\[ \phi_s = \arccos \left( 1 - \frac{d}{R} \right) \]  \hspace{1cm} (2.6)

where

- \( S_{eq} \): equivalent chip thickness or equivalent feed/tooth
- \( S_{t} \): feed/tooth
- \( d \): radial width of cut
- \( R \): cutter radius
- \( \phi_s \): swept angle of cut

\( X \) is called the thermal cycling parameter and is used to characterize the influences of both the range of thermal strain and the number of cycles of thermal, plastic strain on the tool life in intermittent cutting. This parameter is added to the tool life equation in order to account for the fact that the milling cutter deteriorates as it continuously alternates between cutting the material and moving through the air. Obviously, the parameter is closely related to the factors such as the spindle speed and the time in/out of cut. Yellowley and Barrow [22] formulated this relationship as follows:

\[ X = E_r \ast (N \ast y)^{1/2} \]  \hspace{1cm} (2.7)

\[ E_r = 39 \log t_c - 23 \log t_h + 37.5 \]  \hspace{1cm} (2.8)

where

- \( E_r \): range of thermal strain parameter
- \( N \): spindle speed [rpm]
- \( t_c \): cooling time [msec]
- \( t_h \): heating time [msec]
- \( y \): ratio of total cycling time to time in cut
By combining equations (2.4) - (2.8) the tool life can be calculated once the feed, speed, and the geometry of cut are known.

2.3.4. Economics of Single Pass Milling

In a single pass end milling operation the machining parameters to be optimized are feed and speed. To minimize the variable machining time, eq. (2.1) is differentiated with respect to the velocity \( V \), and the result is set to zero to find the required tool life which is expressed as:

\[
T_{\text{min time}} = \left( \left( \frac{p + \frac{n}{2}}{2} \right) - 1 \right) \times t_s
\]  

(2.9)

where \( p \) and \( n \) are tool life equation exponents.

The optimal velocity \( V \) is then found by substituting eq. (2.9) into eq. (2.4). Since \( q << 1 \), the influence of the depth of cut on tool life is deemed negligible and therefore the term "a" is assumed to be a constant. Then the result is expressed as:

\[
V_{\text{min time}} = \frac{C_1}{\frac{1}{p} T_{\text{min time}}^{\frac{1}{p}} S_{eq}^{\frac{1}{pm}}}
\]  

(2.10)

where \( C_1 \) is a constant. By combining eq. (2.1), (2.5)-(2.8), the variable machining time becomes:

\[
t_v = \frac{C_2}{\frac{1}{pm-2} S^{pm-1}}
\]  

(2.11)
where $C_2$ is a constant. Obviously, since $(p \times m) > 2$, $\frac{pm-2}{pm-1} > 0$. Therefore, the minimum solution to the variable machining time equation requires that the feed rate be maximized, and the optimal velocity be selected according to eq. (2.10). Similarly, to minimize the variable machining cost, eq. (2.2) is differentiated with respect to the velocity $V$, and the result is set to zero to find the required tool life which is expressed as:

$$T_{\text{min cost}} = \left( \left( p + \frac{n}{2} \right) - 1 \right) \times \left( t_s + \frac{C_1}{X} \right) \quad (2.12)$$

Following the same procedure, the variable machining cost becomes:

$$t_v = \frac{C_3}{\frac{pm-2}{pm-1}} \quad (2.13)$$

where $C_3$ is a constant. Hence, the minimum solution to the variable machining cost equation requires that the feed rate be maximized, and the optimal velocity be selected in order to yield the tool life in eq. (2.12).

There are, however, several physical constraints which limit the selection of the feed and speed. For milling, the tooth edge breakage constraint requires that the maximum equivalent chip thickness remain below a critical value throughout the operation. Yellowley and Gunn [16] defined this constraint as:
In order to avoid the catastrophic failure of the shank of the tool, the maximum tensile stress on the shank has to be kept below a certain critical value which is material dependent. A set of equations is used to calculate the maximum tensile stress on the shank [9]:

\[
\sigma_{\text{max}} = \frac{2}{\pi R^3} \left( M + (M^2 + T^2)^{1/2} \right) 
\]  
\begin{align*}
M &= \left( 1 - \frac{a}{2} \right) \sqrt{F_x^2 + F_y^2} \\
F_x &= \frac{kaR}{4} \times \frac{v}{V} \times \left( -\left( 1 - \cos(2\phi_s) \right) + r(2\phi_s - \sin(2\phi_s)) \right) \\
F_y &= \frac{kaR}{4} \times \frac{v}{V} \times \left( r\left( 1 - \cos(2\phi_s) \right) + (2\phi_s - \sin(2\phi_s)) \right) \\
T &= \frac{ka v R d}{V}
\end{align*}

where
- \(a\) depth of cut [in]
- \(k\) specific cutting pressure [psi]
- \(v\) cutter traverse rate [in/min]
- \(V\) peripheral cutting velocity [in/min]
- \(M\) bending moment [in.lbs]
- \(F_x\) x component of in-plane resultant cutting force [lbs]
- \(F_y\) y component of in-plane resultant cutting force [lbs]
- \(\sigma_{\text{max}}\) maximum tensile stress in shank [psi]
- \(T\) torque [in.lbs]
- \(l\) tool length [in]
The "=\" signs in the above equations hold true when the average force values are used. The expressions for $F_x$ and $F_y$ are for down-milling. For a typical HSS end mill, the maximum tensile stress is about 180,000 psi. If the stress on the shank exceeds the critical value, the feed rate should be reduced accordingly.

The power required in the end milling operation can be calculated as follows:

$$P = kavd$$

(2.21)

and the torque required is expressed in eq. (2.20). If the power and torque exceed their respective allowable limits (which are machine-specific), the spindle speed and the feed should be reduced respectively. The affects of other constraints such as surface finish and chatter on the selection of machining parameters will not be quantified in this thesis.

2.3.5. Economics of Multi-Pass Milling

In a single pass machining operation only feed and speed are optimized. Often, however, the amount of material to be removed in a machining operation exceeds the allowable width or depth; In such cases more than one cut is required. Many investigations into the economics of multi-pass machining operations lean towards total optimization of width/depth of cut, feed and speed in each pass [14, 15]. Their strategies assume a complete knowledge of tool life and constraint equations which are rarely available in a real machining.
environment. The implementation of such strategies is challenging especially when in practice only feed and speed can be easily varied while the width/depth of cut are rarely altered in any practical shop floor operation. Therefore, it seems more appropriate to consider the optimization of width/depth of cut separately. Yellowley and Gunn [16] provided a general strategy to the selection of width or depth of cut for turning and milling with the presence of tool breakage, power, and torque constraints. If the optimal machining conditions are chosen, the variable machining cost per unit length in a single pass operation is denoted as:

$$C_v = \frac{xt_c}{n} \left( p + \frac{n}{2} - 1 \right)$$

(2.22)

where $t_c$ is the optimal cutting time per unit length and $p$, $n$ are tool life equation exponents. Therefore, in order to minimize the cost of a $k$-pass milling operation, the following objective function, with the substitution of eq. (2.4) and the appropriate constraints equations, has to be satisfied:

$$\min \left( \sum_{i=1}^{k} t_{co,i} \right) = \min \left( \sum_{i=1}^{k} \frac{1}{RPM_i S_i} \right)$$

(2.23)

The strategy is summarized in Figure 2-3. In the case when only the tool breakage constraint is present, the minimum solution requires that all passes except one are taken at the maximum allowable width or depth $d_{max}$, with the remaining pass to remove the rest (see Figure 2-3a).
In the presence of both the power and torque constraints, the variable machining cost is determined by the tool breakage constraint up to a critical value of width or depth beyond which the power and torque constraints become active. In the case when the critical width or depth is smaller than the even width or depth, the minimum solution requires that one pass is taken at \( d_{\text{crit}} \) and the rest at even values, or all passes but one are taken at \( d_{\text{max}} \) and the remaining pass takes the rest (see Figure 2-3b). In the case when the critical width or depth is larger than the even width or depth, the optimal solution requires that all but one pass are taken at \( d_{\text{crit}} \) and the remaining pass takes the rest, or all passes but one are taken at \( d_{\text{max}} \), with the remaining pass taking the rest (see Figure 2-3c).

![Figure 2-3 The influence of torque and power constraints on the relationship between width and cost](image)

2.3.6. Operation Sequencing

In this section a heuristic strategy for the selection and sequencing of removal volumes proposed by Yellowley and Fisher [20] is presented. The
solution suggested here, however, is largely based on the previous effort by Yellowley and Kusiak [19] and is promised to provide a good, if not necessarily the optimal in some unusual cases, solution to the sequencing problem. A simple rectangular part is given as an example to help illustrate the sequencing algorithm (see Figure 2-4).

The material to be removed from the stock is subdivided into a number of non-overlapping elemental volumes. Any elemental volume which has at least one machined surface as a bounding face is called a seed volume. In the example, V₁, V₃ and V₄ are designated as seed volumes. Elemental volumes that can be machined together are combined to form compound volumes. In order to form compound volumes, all seed volumes are extended until no further elemental volumes that can be removed with the originating seed volumes. In this example, two compound volumes are identified: S₁ (V₁, V₂, V₄) and S₂ (V₂, V₃). They are now called feasible compound volumes. The cost per unit volume of each of the feasible compound volumes is calculated and the volume with the lowest cost per unit volume is removed first. All elemental volumes which are contained in the selected feasible compound volume are also removed. The procedure is repeated until no volume is left. The fictitious cost of each volume of the part is listed in table 2.
In this example, $S_1$ and its associated elemental volumes are removed first and $V_3$ is the last volume to be removed. This algorithm is based on the proposition that for single pass operations all elemental volumes have the characteristic that their cost per unit volume decrease as their volumes increase. It is, however, unlikely to perform in an optimal fashion if the majority of the volumes are to be removed with multiple passes.

As the part becomes more complicated the number of elemental volumes can increase very rapidly. Fisher [9] has suggested a heuristic rule that allows
precedence restrictions to be added to the original algorithm, thus reducing the number of feasible compound volumes needed to be considered. The restrictions apply when some certain feasible volumes must be machined before it is possible to machine a given feasible volume. He also proposed an algorithm which dealt with the cost of tool changes. This algorithm, however, tends to cost a significant amount of computational time since it involves calculations of optimal costs for each volume for each tool available. The overall solution discussed here to the sequencing problem has been demonstrated with rectangular part geometry only, while in most practical cases parts with more complicated shapes will be encountered.
2.4. Computer-Aided Manufacturing

2.4.1. Introduction

Computer-aided manufacturing (CAM) involves the use of computer technology to convert planning information into control instructions to be used by a computer numerical control (CNC) machine tools which are used to make the part. Typically, a CAM system, upon receiving information such as part geometry from a CAD database and a sequence of operations from the process planning system, generates a set of machine instructions which will drive the axes and auxiliary elements of the machine tool. In this section, brief reviews of the recent development in NC part programming and CNC technologies are presented.

2.4.2. NC Part Programming

As mentioned in the introduction, Numerical Control (NC) part programming converts information on how a part should be machined, either manually or with the assistance of a computer software, into G-code programs which are then interpreted by the CNC machine's control unit. A G-code program is a standardized codification of motion and control commands needed to produce a part. Traditionally, the part programmer, upon receiving design information, has to determine machining conditions such as the feeds, speeds, and the depths to be used to machine a part considering the work-tool materials, finish requirement, the limitations of the machine, and so on. The programmer also has to program auxiliary functions such as coolant on/off sequences and
tool/fixture offsets. However, the biggest task by far is to program the tool path which the cutter must travel. This task demands a tremendous amount of sophisticated computations and if done manually, requires the programmer to have good knowledge of trigonometry and algebra. A large amount of manual computations means that the code is error-prone. Therefore, as the geometry becomes more complicated, the assistance of a computer software becomes necessary. In this section, two types of systems which assist in NC part programming are examined.

2.4.2.1. APT-Like Languages

APT stands for Automatically Programmed Tools. The initial concept and basic structure of the APT system were developed in late 50's by the Electronic Systems Laboratory of the Massachusetts Institute of Technology and since then it has evolved into an extremely powerful NC programming system. APT is still being used in a number of companies, primarily in the aerospace industry. Other notable APT-like systems such as EXAPT, which was developed in Germany, and Compact II, developed by Manufacturing Data Systems in Ann Arbor, Michigan, are still very much in use. Their structure resemble the structure of the original APT with additions of routines to perform specialized operations.

APT is a 3-D NC part programming language which, like other conventional programming languages such as Pascal and C, can perform
looping, conditional branching and sub-programming functions. It describes, in English-like format, the sequence of operations to be performed by CNC machines. A typical APT program consists of four types of statements. The geometry statements define the geometric elements which comprise the part. These elements include points, lines, circles, and other entities. These entities are unbound which means lines have infinite length and circles are closed. The motion statements define the actual path followed by the cutting tool relative to the described geometry. The postprocessor statements specify control features such as feed, speed, and coolant injection. The auxiliary statements are there for the completion of an APT program. They define cutter sizes, tolerances, part identification, and so on. Macros are frequently used to program repetitive, specialized tasks such as the calculation of the tool path and the associate cutting conditions for a general-sized pocket removal. Once an APT program is complete, the computer program interprets the APT statements and generates a neutral format cutter-center path location file (CLFILE) which is then compiled by a postprocessor. In most APT systems, one can configure the postprocessor such that the G-codes produced by the postprocessor are recognized by specific machine tools. The G-code program is then either saved entirely or drip-fed into the memory of the machine tool's control unit. More advanced features of APT systems include the addition of functions such as graphical NC path verification and graphical APT programming. A graphical NC path verification system takes a CL data file and animates the tool motions for visual error checking. A
APT is versatile and powerful, in terms of its ability to give the programmer a great amount of flexibility in the way in which he/she can describe a part and program the tool path. The application of macros is not only able to reduce precious time for programming repetitive operations, but also capable of calculating the most economical machining conditions. However, APT has its weaknesses. A large time investment has to be made in part geometry description, tool path programming, and initial development of special purpose APT macros. Like other conventional programming languages, errors of syntactical and logical types may exist in APT programs and time is lost in debugging the programs.
2.4.2.2. Interactive CAD/CAM

A CAD/CAM system is a computer system equipped with a variety of software which is capable of accomplishing functions of both design and manufacturing in a highly interactive and integrated environment. The integrated system has two basic features which make it a more attractive alternative to regular APT part programming systems: the ability to input geometry definitions interactively and graphically, and the ability to automatically generate tool paths directly from CAD models.

In a typical CAD/CAM system, the part geometry is either developed interactively by the designer via a computer graphics terminal, or imported from other CAD systems. The most noticeable advantage of having the part geometry represented graphically instead of using a text-based method such as the one employed by APT is the possibility of immediate visual verification of the design. Interactive modification of the existing design can easily be accomplished. The part model created by the CAD portion of the system is then stored in a database which is used later for tool path generation. In the CAM portion of the system, information is entered entirely via a graphical user interface. The user highlights the part geometry to be processed using pointing devices such as a digitizer or mouse. The selection of machining parameters such as feeds, speeds, tool lead in/out, depth of cuts for roughing and finishing, tool types, cutter immersion, tool offsets, and tolerance is accomplished by entering suitable values with the assistance of dialog-type menus. Predefined routines of
common machining cycles such as pocket milling, surface contouring, engraving, and drilling are also provided. If the operation consists of more than one type of cutting cycle, the sequencing strategy employed to machine the part are usually determined based on the operator's knowledge and experience. Some CAD/CAM systems even have tool motion animation capability to "dry-run" the entire process in order to check for such things as tool collisions. Finally, once all of the machining information has been defined, a single command creates the G-code program which is then ready to be transferred to the CNC controller on the factory floor.

The CAD/CAM approach to NC part programming provides much more convenience in terms of part description, tool path generation, and process verification than a regular APT system can provide. However, it is still not able to apply the most economical machining conditions, nor to solve the problem of sequencing operations. Therefore, it becomes obvious that embedding a knowledge of process economics into a CAD/CAM system is the only way to guarantee that parts are produced in the most optimal fashion. Hart [24] demonstrated the feasibility of applying optimal machining conditions to multi-pass turning operations while retaining all the advantages of both CAD/CAM and APT systems.
2.4.3. Computer Numerical Controls

Computer numerical control (CNC) technology has been used to automate machine tools for many years. Applications, ranging from automotive and aerospace, to die and mold making, make use of advanced CNC technologies to achieve enormous increases in productivity. To meet the demand for more efficient production, industry and related academic groups have been actively involved in conducting research on making existing CNC technologies more advanced and most importantly, more affordable. Traditionally, machine tool controls have relied upon specialized controllers with vendor-specific architectures. Integrating innovative functions, such as process data extraction and sensing, with the existing control entails custom design of hardware and software interfaces, thus leads to huge development and maintenance costs. Therefore, as recent development in embedding real time intelligence within the control has already shown tremendous promise on more safe and cost-effective operations, a flexible control architecture to smoothly accommodate such technology needs to be developed. In recent years, there have been some research projects which have tried to standardize the interfaces among a wide variety of hardware and software components in order to achieve system openness. NIST\textsuperscript{a} [25] proposed the development of the Enhanced Machine Controller (EMC) architecture in an attempt to specify standard tasks for interfaces of individual components such as sensor integration, axis controls

\textsuperscript{a} NIST stands for National Institute of Standards and Technology
and trajectory generation. An open architecture control system has also been developed under a joint European research project called OSACA [26]. The system consists of a set of well-defined controller modules and associated APIs in an attempt to achieve openness and modularity. Instead of trying to specify interface standards for various software, processors or communication systems. Yellowley et al. [27, 28] proposed an open architecture CNC system in an effort to implement both traditional interpolation and process control activities within a highly parallel control arrangement. The basic operating software used to access system resources is written in FORTH for its accessibility, extensibility and compactness. They demonstrated successfully the extensibility and flexibility of the system in accommodating applications such as CAD/CAM, axis control, process identification, control and monitoring in a tightly integrated environment [29, 30, 31]. More details on how open architecture controls contribute to the research objectives of this thesis are given in the later section.

Despite recent success in the development of open architecture control systems, considerable effort has still been made to extend functionalities of some existing proprietary controllers. Memex Electronics Ltd., for instance, has been specializing in developing memory and networking technologies enabling Fanuc controls to "open up" to external systems [32].
2.5. Process Monitoring and Control

Process monitoring and control involve real time measurement and adjustment of process parameters to achieve optimal machining economics, while at the same time ensuring machining safety and product quality. As mentioned in the previous sections, machining parameters such as cutting speeds and feed rates are selected in the process planning stage to satisfy process constraints and to arrive at a minimum cost machining solution. In reality, however, such a solution is rarely obtained because exact knowledge of geometry, work material and tool properties are usually unavailable in the planning stage. Thus, on-line acquisition of process knowledge and real time optimization of the machining parameters would be helpful especially since the dynamic nature of a machining process usually leads to unmodeled variations in workpiece hardness and tool wear. To address these problems, research efforts have been made to develop adaptive control strategies for machine tools to take advantages of recent advances in process modeling and sensor and control architecture development.

From a machine tool perspective, the term adaptive control describes a control system that is able to measure certain process outputs and use these to adjust feed rate and spindle speed to achieve some prescribed objectives. In particular, there are two approaches to adaptive control in machining. The first approach is called Adaptive Control Constraint (ACC). In these systems, metal
removal rate is maximized within the prescribed limits of the constraints such as force and torque. Most adaptive systems are of this type and most systems try to maximize the feed rate within the cutting force limit [33, 34]. These systems ensure maximum metal removal rate, but not minimum cost. The second approach is called Adaptive Control Optimization (ACO). In these systems, an index of performance based on a process economic model is optimized subject to process constraints such as force and torque. Although this type of system is thought to be a more effective optimization strategy, it has not been widely used in practice because of the difficulties in identifying realistic economic indices and finding reliable tool wear sensors, which are instrumental in estimating tool life parameters. Recent developments in real time ACO systems have been reported for turning [29] and milling [1]. Both of these systems circumvent the necessity of on-line tool wear tracking by maintaining a constant relationship between equivalent chip thickness and cutting velocity with an attempt to yield a uniform tool wear rate or tool life. To avoid tool edge and shank breakage, the feed rate is always reduced to prevent the identified equivalent chip thickness and maximum forces from exceeding their respective maximum allowable limits.
2.6. Computer Networking for Manufacturing

The implementation of a truly computer integrated manufacturing system requires various computer devices be able to communicate efficiently with each other. In order to transport information between these computer-related devices, they must be physically connected, either by internal buses or cables. The selection of networking solutions is application-specific. Depending on the nature of applications, which can range from real-time process control requiring program execution at precise times, to file transfer and sharing which is not subject to the constraints of real-time computing, different types of networking solutions may be employed. In either scenario, however, proper selections of networking hardware and software are critical.

In a typical production facility, real time and non-real time activities often require the services of different types of networks. The so-called low level control network is often present in for instance a manufacturing cell which is composed of a small group of automated machines (predominately CNC machines) which are designated to perform specific tasks. This type of network is used to synchronize, in real time, all the operations which may include tool changes, stock loading, and machining within a manufacturing cell. Because various programs have to be run at a precise time, the choices of networking interfaces are quite limited. Merton [31] developed a simple distributed control architecture which was intended to synchronize multiple CNC machining
processes. The low level control network of the experimental system connects the controller of a CNC lathe to a pair of open connector lines in order to provide synchronized real-time feed control. The interface between the low level control network and the CNC controller is done via direct port manipulation. One of the advantages of using a direct port interface instead of a standard software protocol in this real time application is that it provides faster and rather deterministic communication. Also, computer resources are not frozen up due to the absence of software overhead. However, the inflexibility of interface design means longer development time and more substantial maintenance cost.

On the other hand, the so-called high level network is mainly responsible for non-real time applications such as file transfer, sharing, and printing. These are applications more often found in an office environment. In a manufacturing enterprise, CAD data files that are stored electronically in a central database are readily transferred from the design department to the production planning department via the network. On the shop floor, the most common application of high level networks is the retrieval of NC part programs and tooling data from the CAD/CAM database. Also, the network offers an opportunity to monitor the machining state on-line, not just by the operator sitting beside the machine, but also by the factory manager who may use up-to-the-minute process information to improve the existing production plan.
One of the earliest networking technologies to be employed on the shop floor is called direct numerical control (DNC). The term DNC refers to a group of machine tools which are directly controlled by a central computer. The first generation of DNC systems was introduced in the late 1960s when both digital computers and NC controllers were still very expensive. These DNC systems were designed with the idea of replacing hard-wired logic in the individual NC controller of each machine tool by software in the central computer. They operated in a time-shared mode, with a supervisory software program running in the central computer linking the machine tools' controllers and establishing any necessary priority [35].

![Figure 2-5 A Behind-Tape-Reader DNC system](image)

The most noticeable disadvantage of the early DNC system design is that the design of the supervisory program is highly machine-specific; adding or removing a machine tool to or from the DNC system may involve substantial modification of the software logic and timing schemes that have previously been
defined within the supervisory program. Because special-purpose hardware and software were used, the original DNC systems were very expensive. It was difficult to retrofit existing NC controllers into such a system without scrapping the NC controllers.

In the mid 1970s, a new concept in DNC system design was introduced. The new DNC configuration used a Behind-Tape-Reader (BTR) interface to facilitate communication between the central computer and the existing NC or CNC controller. In a BTR DNC system the part program input device, typically the tape reader, of an existing NC or CNC system was replaced by a data-transmission channel and BTR unit. The BTR unit provides the tape-reading operation of the machine tool's NC or CNC controller unit. Typically, each BTR unit used two storage buffers to allow smooth data transmission between the DNC computer and the machine tool controller. While one buffer is receiving a block of instructions from the DNC computer, the other provides machine instructions to the NC or CNC controller (see Figure 2-5). Many implementations of this technology have been reported [3, 35, 36, 37, 38] and some of the benefits derived from this technology include:

- the BTR interface can be easily adapted to any type of machine tool for ease of system modification.
- existing NC control equipment does not need to be scrapped.
- punched tapes and tape readers are eliminated for improved reliability.
• the centralized part program storage allows the latest versions of part programs to be available to the operators.
• the operator no longer needs to leave his machine and walk to the storage room to retrieve the part programs which would be stored in disks or tapes.
• the expensive central computers are not located in the hazardous machine-shop environment.

The availability of less expensive magnetic storage devices in the early 1980's, together with the increasing popularity of CNC technology, redefined the role of a direct numerical control system. A transition took place where direct numerical control re-emerged as distributed numerical control, whose acronym is also DNC [39]. Instead of being used to control a group of stand-alone NC and CNC machines in an "on-line" fashion, the central computer of a modern DNC system is more often used to store and distribute complete part programs, tool offset files, and other machine set-up files to the CNC machines. Since most modern CNC controllers are equipped with RS-232 interfaces (with data transmission rate as fast as 115,000 bits per second) as standard I/O ports, the communication link between the central computer and the individual CNC controllers is usually a network of serial cables. The central computer of a modern DNC system is mainly responsible for storing NC part programs in its memory storage unit and transmitting complete part programs into the requesting CNC machines via the serial network. The central computer can also
"drip-feed" the programs in blocks to the memory units of individual CNC controllers in the case when the size of a part program exceeds the capacity of the CNC controller.

In addition to the basic functions such as part program storage and transfer, DNC has assumed the role as an information backbone of the entire manufacturing enterprise. The DNC installations at General Electric and Westinghouse Electric Corporation demonstrate how the basic DNC can be extended by including such functions as CAD/CAM and planning, inventory control, process control and monitoring, material handling, assembly line control, and performance evaluation [39, 40]. With the rapid emergence of local area network (LAN) technologies which have become the most widely used solutions to office-like networking problems, traditional serial networks on the shop floor are gradually being replaced by networking media such as ethernet to allow not only faster flow of information, but also tighter integration between machine tools on the shop floor and CAD/CAM workstations in the engineering department [41].
2.7. Summary

CIM touches areas as diverse as drafting, design, analysis, and planning in the engineering department; machine control, inventory control, and quality assurance on the shop floor, and report generation, forecasting, and plant management in the front office. In this chapter a brief review of several components of a CIM system was presented. The majority of the discussion has been focused on the following areas: computer-aided process planning (CAPP), computer-aided manufacturing (CAM), process monitoring and control, and networking for manufacturing. The manufacturing industry is progressing rapidly towards the use of CIM. Since there are no off-the-shelf solutions to such system, users must build their own CIM systems by buying and tying together different pieces of software and hardware from different vendors. Traditional, but limited strategies of integrating design and manufacturing have been adopted to allow generation of reliable control information directly from design information. Such integrated systems, however, seldom produce a good low level process plan which becomes the key to minimal cost machining. In the next chapter, the development of an integrated system which attempts to embed low level process planning intelligence into commercially available CAD and CAM systems will be presented.

There has also been significant effort to develop reliable networking solutions to aid activities such as low level process control/monitoring and high-level enterprise-wide resource sharing. As low cost, high bandwidth network
connections are more widely used within a manufacturing environment, real time process information may be made accessible to all levels of management. The capability of modern computer software systems to represent real time process information graphically provides an opportunity for production personnel who do not work on the shop floor to "visually" monitor a group of machining operations. In chapter four, the development of a network-based, real time machining process animation system based on virtual reality technology will be presented.
3. Integration of CAD/CAPP/CAM

3.1. Introduction

This chapter presents an overview of the integrated CAD/CAPP/CAM system developed in this thesis. The system was designed primarily for 2\(\frac{1}{2}\)-D milling operations. It accepts a CAD part model as an input, and produces a set of optimized control instructions, which can be readily used by a general machine tool controller. Detailed discussion on both the system's structure and software implementation of its components is given. A few examples are also presented to demonstrate the system's capability. The chapter is then concluded with a brief glance at the system's contribution to the realization of dynamic process planning.
3.2. **Software Selections**

The integrated system consists of three major components: a geometry input or CAD module, an automated process planning module, and a CAM module. Both the CAD and CAM modules were implemented using commercially available solutions. An automated process planning package, which has been under development in the Manufacturing Engineering Laboratory at UBC for the last few years, was used to derive a low level, optimized process plan. The original planning module was developed by Seethaler [1], and only minor modifications were necessary for this part of the work.

The automated process planning system used in this project was originally developed for optimizing 2½-D milling operations. It was written in FORTH and designed to run as a 32 bit DOS protected mode application. The solid modeling kernel used employs a combined cell decomposition and boundary representation to describe part geometry. Solids are modeled as rectangular volumes, and the data structure for each volume contains technological information such as the nature of the volume (e.g. full, empty, to be machined) and the nature of its faces (free, seed, or interface). However, a provision for users to enter tolerance and surface finish constraints is not available. The system is equipped with a text-based CAD interface that allows part geometry to be entered using a proprietary, easy-to-use part description language [1]. The language itself is feature-driven, and is able to specify
dimensions and locations of common machined features (e.g. slots, shoulders, surfaces, etc.). The system also consists of a set of libraries where the properties of different types of tools, machines and workpiece materials are specified. The system possesses both preprocessing and real time capabilities. In the preprocessing stage, the system performs operation sequencing, width/depth subdivision, and single pass optimization of feed and speed. The operation sequencing algorithm attempts to derive the optimal machining sequence using the analysis given in chapter 2.3.6. In the width/depth subdivision algorithm, preference is given to the smallest cutter which can remove a volume in a single pass. If a large enough cutter is not available, or if no cutter fits the geometry to be removed, a multi-pass operation is needed. The feed and speed for each single pass operation are selected to minimize machining cost. When the machine starts cutting, the system continuously receives updated process data and adjusts relevant machining parameters to achieve minimum cost and guarantee safe operation. The software has been integrated with the UBC open architecture CNC controller to demonstrate its real time capability [1,17].

The CAD system selected for this work must provide both the "approximate" part geometry for the automated process planning system and the "exact" geometry for the NC part programming system (more detail on these topics is presented in Chapter 3.4). The actual software selected for the work is Autodesk's AutoCAD release 13 which is a Windows-based CAD system which
runs on a PC platform. The system allows the designer to use AutoCAD's graphical front end and command line text dialogue box to enter the part description using devices such as a mouse and keyboard. Parts may be represented by either wire-frame, surface, or solid models. Part drawings in AutoCAD are represented by a proprietary drawing file format called DWG which is capable of representing geometry, layer, dimension, and text information within AutoCAD. In order to allow data exchange with other CAD applications, data files of common formats, such as DXF and SAT (a CAD data representation standard used by ACIS-based solid modellers), can be created from or converted into DWG files by the system's built-in file converters. However, solutions to other common file conversion problems such as IGES\(^b\)-DWG and STEP\(^c\)-DWG data translation are not provided by the software. AutoCAD's application programming interface (API) allows third-party routines, which are written in either C, C++, ATLAST\(^d\), or AutoLISP, to run within the system. In fact, the API plays a key role in the development of software interfaces that allow smooth data transfer among AutoCAD, the automated process planning system, and the NC part programming system.

The CAM software used in this work was Personal APT version 2.5. It is a DOS-based 2\(^{1/2}\) axis NC/CNC programming system which utilizes a subset of the

\(^c\) STEP stands for STandard for the Exchange of Product model data, developed by the IGES/PDES Organization (IPO) in the mid-1980s.  
\(^d\) ATLAST stands for Autodesk Threaded Language Application System Toolkit.
standard APT part programming language. The reason that an APT system was chosen in this work is largely due to the ease of integrating the most economic machining conditions derived by the process planning system. Personal APT features an integrated Program Development Environment (PDE) which consists of a program editor, an APT processor, a graphic tool path and geometry verification module, and a user-configurable general purpose post-processor. It provides most of the standard APT words plus some non-APT functions such as branching that uses Basic-like GOSUB and RETURN statements, and DXF geometry import. The DXF import function is particularly useful when an irregularly curved part profile or tool path is to be defined. The standard APT representation of such complicated geometry usually results in numerous definitions of points, which are located along the profile, to be connected by line segments. In this work, since only simple entities such as straight lines and circles/semi-circles were considered, the DXF import function (and other non-standard APT functions) was not used in an effort to maintain both compatibility with the standard APT language and flexibility in commercial APT software selection.
3.3. Choice of Data Exchange Format

Since only the geometric information of a part needs to be shared within the integrated CAD/CAPP/CAM system, it was concluded that a rather simple data exchange format would suffice. The DXF format, which was developed by Autodesk, was chosen as the standard data exchange format used for translating geometry data among different components of the integrated system. A standard DXF file is a plain text file with a file extension of .dxf. It can be read or modified using standard word processors or text editors. The contents of a DXF file are arranged into the following six sections: header, classes, tables, blocks, entities, and objects \[42, 43, 44\]. A full DXF file includes all the components of a .dwg file including graphical entity definitions, block definitions, linetype definitions, layer information, text styles, dimension styles, named views, named coordinate systems, and so on. In this section, entries in a DXF file which store geometric information of a graphical entity are discussed.

<table>
<thead>
<tr>
<th>0</th>
<th>SECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>ENTITIES</td>
</tr>
<tr>
<td>0</td>
<td>LINE</td>
</tr>
<tr>
<td>5</td>
<td>21D</td>
</tr>
<tr>
<td>100</td>
<td>AcDbEntity</td>
</tr>
<tr>
<td>8</td>
<td>LAY1</td>
</tr>
<tr>
<td>100</td>
<td>AcDbLine</td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3-1 LINE in DXF format
A DXF file is composed of many two-lined code/value groups. Figure 3-1 shows an example of a DXF file for a simple AutoCAD entity "line". In the group 0/LINE, 0 is the group code which denotes a text string that indicates the entity type. The value of the code 0 is LINE. Within the "line" definition, the groups 10/x.x, 20/y.y, and 30/z.z represent the x, y, and z coordinates of the starting point of a line in a 3-D coordinate system. The groups 11/x.x, 21/y.y, and 31/z.z represent the x, y, and z coordinates of the end point of a line. The length of the line can also be calculated. In this example, the extremities of the line are located at (3.0, 0.0, 0.0) and (4.0, 2.0, 0.0). The length of the line is approximately 2.24. Figure 3-2 shows an example of a DXF file for a simple AutoCAD entity "circle". In the group 0/CIRCLE, the value of the code 0 is CIRCLE. Within the "circle" entity, the groups 10/x.x, 20/y.y, and 30/z.z represent the x, y, and z coordinates of the center of a circle. The group 40/r.r represents the radius of a circle. In this example, the center of the circle is located at (3.5, 1.0, 0.0) with the radius of 1.0.

```
0
SECTION
2
ENTITIES
0
CIRCLE
5
21E
100
AcDbEntity
8
LAY1
100

AcDbCircle
10
3.5
20
1.0
30
0.0
40
1.0
0
ENDSEC
0
EOF
```

Figure 3-2 CIRCLE in DXF format
AutoCAD allows interactive tolerance input. The DXF format is able to store both dimensional and geometric tolerance information. However, the way in which DXF stores tolerance information makes it very difficult to associate the tolerance information with the intended geometrical entities. In this thesis, therefore, no attempt has been made to store tolerance information into the DXF file. The tolerance information associated with a particular component of a part geometry is not carried over to the automated process planning system or the NC part programming system.

As AutoCAD release 13 uses the ACIS® modeling kernel to describe 3-D solid objects, these objects appear in the drawing database and consequently in the DXF file. However, Autodesk, as an ACIS licensee, is not allowed to document the data structure used to describe these objects, which means that the DXF file will contain binary ACIS data, associated with 3-D solid objects, that only AutoCAD is able to understand. Two alternatives are available for extracting the content of an ACIS solid object within AutoCAD. The first alternative involves representing ACIS 3-D solid objects as SAT entities. SAT, developed by Spatial Technology, is a proprietary file exchange format which was designed and implemented to facilitate sharing of 3-D data between ACIS-based applications. Like DXF, SAT files are text files which can be read or edited in most word processors. However, the rather cumbersome structure of the SAT representation makes extracting the detail of an ACIS object a daunting task.

\*ACIS® was developed and marketed by Spatial Technology Inc.*
The other alternative is to preprocess 3-D ACIS solid objects, which can be created in or imported into the AutoCAD drawing editor, such that only simple entities such as lines, arcs, and circles are used for part geometry description. These simple entities are then stored in a DXF file for downstream applications. In this work, the latter alternative was adopted. Some of the advantages of the latter alternative are that each entity type is clearly labeled and important parameters such as locations about the edges of a 3-D ACIS solid object are listed in the DXF file in an orderly fashion. The extraction of relevant geometry information of the part then becomes a relatively simple task as only a few file parsers need to be constructed. Also, transforming an ACIS solid object into a group of lines and arcs is as simple as issuing a single command (e.g. EXPLODE) at the AutoCAD command prompt.
3.4. Approximation of Non-Orthogonal Part Geometry

The low level process planning package used in this work is capable of accepting part geometry with rectangular removal volumes as an input. In reality, however, most workpieces exhibit a more complex geometry. While the direct integration of complex part geometry into the process planning system seems to be a logical thing to do, the task itself is not a simple one. It would seem, in most cases, that an approximated set of volumes will result in a good sequence of volume removal for the more complex actual part geometry [20]. While this may represent a good solution to the sequencing problem associated with complex geometry, the difficulty in realizing such proposition becomes evident when one has to prescribe a general set of rules on which the geometry transformation is based. In this section, a brief discussion of issues regarding the geometry transformation is presented. The purpose of the discussion, however, is to demonstrate that the difficulty involved in this kind of problem should not be underestimated. Little attempt has been made in this work to formulate a formal set of rules which the procedure of approximating non-orthogonal geometry should follow. Instead, some practical guidelines were suggested for deriving a reasonable set of approximate geometry in an effort to find a good volume removal sequence for the more complex actual component. A few examples are given to illustrate the working principle of the guidelines. Finally, a simple
heuristic rule for transforming volumes with circular/semi-circular edges into straight-edged, rectangular volumes was implemented in the integrated system.

In this thesis, a non-orthogonal part geometry is defined as a geometry which possesses at least one of its boundary edges which is not parallel to any of the stock edges. The stock is assumed to be rectangular in shape. Most of the non-orthogonal geometry described in this work is comprised of triangles, polygons (except squares and rectangles), circles, and semi-circles. Figure 3-3 shows a simple part geometry where the material in the closed-end slot and surrounding the cubic boss need to be removed. The cutting tool used in this example has a diameter of 0.5 inch. This example serves to demonstrate not only the strategy used in this thesis to find the optimal cutting sequence for non-orthogonal part geometry but also the underlying difficulty encountered in the sequencing process.

Figure 3-3 An example part
There are a number of ways to machine this part but only the seed volume approach is demonstrated here. The approach requires that the entire removal volume is partitioned. Usually, the decision on whether the geometry is approximated should have been made before partitioning is performed. It will soon become obvious that the decision indeed depends on both the resulting machining cost and the complexity of the part geometry itself. In the case where the non-orthogonal geometry is to be approximated, all the non-horizontal and non-vertical edges of the original part geometry are replaced by horizontal and vertical edges respectively. The "rule of thumb" employed here is that the amount of material removed to create the transformed geometry should be roughly the same as the amount removed to create the original geometry. Figure 3-4 shows two POSSIBLE versions of the approximated part geometry and their resulting partitioning schemes.

![Figure 3-4 Two versions of approximation of the example part](image)

In either version, the cubic boss is rotated by 45 degrees so that the edges align with the edges of the stock. The closed-end, diagonal slot is
replaced by a closed-back-end shoulder which roughly spans the same amount of area. Once the tool information becomes available, the cost of removing each approximated feasible combined volume can be calculated. The most optimal cutting sequence is then derived according to the procedure outlined in chapter 2.3.6. Finally, the tool paths are generated after the real geometry is retrieved. Figure 3-5 shows the actual cutting sequence of the original part geometry derived by each version. Both versions of the approximation strategy arrive at the same conclusion that the material above the diagonal slot should be removed first. Each version, however, allow some degree of flexibility in the way the edges of the real geometry are selected for the subsequent feasible volumes. Both versions call for the shoulder at the back of the part to be the second combined volume to be removed. In version (i), the volume a-b-c-d-e-f-g-h-i-j-k is selected. The volume e-f-g-h is added to the combined volume since it does not exist in the approximated geometry. In version (ii), the volume a-b-c-d-e-h-i-j-k is selected. Obviously, the amount of material removed in version (i) in this case is different from the amount removed in version (ii). The same scenario occurs in both versions during the selection of subsequent feasible volumes. The only other characteristic which is shared by both versions of geometry approximation in this example is that each subsequent feasible combined volume occupies a smaller amount of volume than its predecessor does. No further investigation of the matter is pursued here. In general, it should become obvious that both the resulting machining cost and the machinability of the
specified feasible volumes are among the major factors which must be considered when one has to decide which versions of geometry approximation should be employed.

Figure 3-5  Operation sequences for both versions of approximation of the example part
Figure 3-6  Non-orthogonal partitioning of the example part

The partitioning of the overall removal volume can also be achieved without having to preprocess the non-orthogonal geometry. Figure 3-6 shows the original part geometry and the resulting partitioning scheme. One should notice that the elemental volumes as well as the combined feasible volumes are no longer rectangular in shape. In order to derive the most optimal cutting sequence, the cost of machining each polygon-shaped volume need to be calculated. Some research efforts have been devoted to finding optimal machining conditions for milling polygonal surfaces. Wang et al. [45] presented a mathematical model for finding the optimal tool path for face milling of polygonal surfaces. In their work, both window frame and stair case milling were considered. Later, Prabhu et al. [46] developed yet another method to calculate the optimal tool path on an N-sided convex polygon surface. In their work, only
stair case milling was considered. One of the conclusions which could be made from both studies was that for stair case milling of polygonal surfaces, the optimal cutting orientation is normally parallel to the longest edge of a given polygon. One should notice, however, that both tool path planning strategies mentioned above aim to minimize the total length of cut. In most practical situations, however, minimizing machining cost is of the uttermost importance. It has been proven in numerous literature that machining parameters and constraints such as feed, speed, width of cut, tool breakage, and power limits are among the most crucial factors which affect the overall machining cost. Singh [47] investigated the optimal tool path planning strategy, which included considerations of the workpiece geometry, tool sizes, cutting orientation, machining parameters and constraints, for N-sided polygonal surfaces with an attempt to achieve minimum cost machining. Once the result of this research is integrated with the operation sequencing algorithm described earlier, the solution to the direct sequencing problem of the non-orthogonal part geometry would be obtained.

Although no attempt has been made to discuss in detail the advantage of choosing either one of the approaches of planning the cutting sequence over the other, it is reasonable to say that the resulting machining cost is one of the most important factors which may affect one's decision on the selection of the planning approach. There are some cases, however, where the application of geometry transformation may significantly reduce the computational effort of
deriving a good cutting sequence. Figure 3-7 shows a non-orthogonal geometry which is made up of a number of vertical/horizontal and tilted edges. In this example, partitioning the original geometry would result in a lot of elemental volumes with small and irregular shapes (see Figure 3-7a). However, after the original geometry is transformed into the geometry which has only rectangular volumes, a rather simple sequencing algorithm such as the one being used in this work should be more than capable of deriving a good volume removal sequence for the actual complex part (see Figure 3-7b).

Figure 3-7 Another example part with non-orthogonal entities
The idea of transforming non-orthogonal geometry was implemented within the integrated CAD/CAPP/CAM system developed in this thesis. In order to ease the software development effort, the system was designed so that only the part geometry with straight, circular and/or semi-circular edges was dealt with. The automated process planning system used in this work requires that all circular and semi-circular edges are transformed into straight edges. The heuristic rules used to accomplish these types of transformation are demonstrated by the following examples. Figure 3-8 shows a simple part which has a step with a semi-circular end. In order to partition the entire removal volume, the semi-circular edge is transformed into a straight edge. The location of the straight edge is determined such that the amount of material removed from the original slot remains roughly the same as the amount removed from the new
slot. The same rule can be applied to the part geometry shown in Figure 3-9 where the cylinder is replaced by a cubic boss.

![Figure 3-9 An example part with a circular entity](image)

Original Geometry  Approximated Geometry
3.5. Software Development for CAD/CAPP/CAM Integration

An overview of the implementation of an integrated CAD/CAPP/CAM system is presented. The system runs on a low cost IBM compatible personal computer equipped with an Intel 100MHz Pentium processor. The computer is connected to a local area network so that finished part programs are readily downloaded to the specified CNC machine(s). Both the automated process planning and APT systems run in DOS while AutoCAD is a Windows-based application. In order to meet the requirement for a multi-OS environment, Microsoft's Windows 95 operating system (which has a built-in DOS environment) was chosen as the platform for system development. Details on the system's structure and software implementation of its components are presented in the following sections.

3.5.1. An Overview of System Architecture

The integrated CAD/CAPP/CAM system consists of five major components: AutoCAD, the automated process planning system, the APT NC programming system, the database which stores both geometric and manufacturing related information, and a set of software interfaces which ensure smooth data transfer between the database and other components of the system. Figure 3-10 shows a block diagram which summarizes the approach taken in this work towards CAD/CAPP/CAD integration.
AutoCAD's graphical user interface (GUI) acts as the main control console which oversees the entire operation of the integrated system. Major functions that include part geometry input and transformation, operation sequencing, APT NC part programming, and G-code generation have been made accessible from the GUI's pull-down menus. Auxiliary commands were provided for viewing and modifying entries in the central database and uploading part programs to CNC machine(s). The process starts with the part geometry input. A 3-D part model is either drafted directly in AutoCAD's drawing editor or imported from other CAD packages using a variety of built-in file exchange mechanisms such as DXF to DWG and SAT to DWG. The drawing may contain information other than the geometry such as dimensions and tolerance constraints. However, only the geometric information about the part model is processed. In the cases where the original part geometry contains non-straight edges, they are transformed into straight edges according to the analysis presented in chapter 3.4. The transformed part geometry is represented in DXF format and stored temporarily in the central database for subsequent processing. The DXF files are processed by a set of interface programs so that the geometric information about the approximate removal volumes is represented in the form which is recognized by the automated process planning system. The automated process planning system possesses a set of libraries where manufacturing related information such as tool properties, workpiece material properties, and machine capabilities are predefined and stored. By issuing text-
like commands, both the optimal cutting sequence and the associated tool selection are derived. The results are then stored in the central database and retrieved later for NC part programming. To generate the G-code for machining the part, the following tasks have to be performed:

- identify the real part geometry,
- retrieve the optimal cutting sequence and tool selection from the database,
- calculate the optimal machining parameters,
- generate tool paths, and
- generate all the necessary APT statements.

Machining parameters including widths/depths of cut, feeds, and speeds are derived using the optimization algorithms presented in chapter 2.3.2 to 2.3.5. A set of software interface programs have been written to handle all these tasks. Finally, the APT part program is processed by a general purpose milling postprocessor which generates a G-code data file. Figure 3-10 shows the self-explanatory schematic of the overall system architecture.
3.5.2. Choice of Programming Language

The programming of the software interface development was accomplished using AutoLISP; AutoLISP is a subset of the standard LISP programming language. AutoLISP is an interpretive language which has been enhanced by Autodesk to allow development of custom AutoCAD applications which require computation and manipulation of the drawing database. The
selection of AutoLISP as the development language was made because AutoCAD provides seamless integration between its kernel and AutoLISP applications. Like other conventional programming languages such as Pascal and C, AutoLISP provides basic functions such as arithmetic, string handling, looping, conditional branching, error handling, and file reading and writing. Utility functions are provided for CAD-specific activities. The geometric and user input utilities, for instance, allow AutoLISP applications to obtain pure geometric information from the drawing and prompt the user for input of data in the middle of a process respectively.

3.5.3. Software Implementation

This section presents an overview of the implementation of the interface programs denoted by (a), (b), and (c) in Figure 3-10. The functions of each interface will be explained with the aid of a simple example. The actual content of the code, however, will not be given.

3.5.3.1. Interface between AutoCAD and the Database

The software interface (a) is responsible for converting the approximated 2\(\frac{1}{2}\)-D milling part model into a geometry representation which is recognized by the automated process planning system. The planning system employs a feature-driven, natural language interface (NLI) which was particularly designed for describing simple 2\(\frac{1}{2}\)-D milling parts [1]. The syntax of the interface is discussed later in this section. The input to the interface is a transformed part
model displayed in AutoCAD's drawing editor. The interface first prompts the user to define the cross section planes through the part model at the heights where the width or length of the part changes. The top and bottom surfaces are also included. Figure 3-11 shows a simple $2^{1/2}$-D milling part and the cross section planes. The part features right, left, front, and back shoulders.

![Cross section planes](image)

**Figure 3-11  The example of an approximated milling part**

The example part has five cross section planes. The stock size is denoted by the dash-dot lines. By examining the corner coordinates of each plane, the following information can be obtained:

- the number of elemental volumes that the stock is divided into. The example part has $(3 \times 3 \times 5 = 45)$ elemental volumes
- the locations of the elemental volumes' boundaries in $x$, $y$, or $z$ direction
- the dimension of each elemental volume
The information above is extracted using the DXF file parsers and stored in the DXF files automatically. The next step is the identification of the part features. The approach taken here is to remove unwanted material from the stock one "depth" at a time (see Figure 3-11), starting from the material bounded by the top two cross section planes (e.g. depth 1 in Figure 3-12).

Figure 3-12 Part geometry bounded at each depth

Figure 3-12 shows the part geometry bounded at each depth. The unshaded area denotes the material that has to be removed from the stock. The shaded area at each depth, therefore, represents the area occupied by the corresponding cross section plane. The corner coordinates of each cross section
plane are recorded by the interface software in a DXF file automatically. The interface software then examines each DXF file and assigns "1" to the corners of the overall shaded area and "0" to other intersections of the boundaries at each depth. The 0's and 1's in each row are grouped together to form a row array. All the row arrays at each depth are then grouped together to form a row matrix. The goal is to determine, by examining the row matrix, which elemental volumes bounded by any pair of horizontal boundaries need to be removed from the stock at each depth. The decisions are made based on both the arrangement of 0's and 1's in the current row array and the result obtained for the previous row array. The decision making process starts by examining the left-most element of the row array at the bottom edge of the stock. Table 3 summarizes the heuristic rules applied in the decision making process by the interface software.

![Figure 3-13 Legends for Table 3](image)

<table>
<thead>
<tr>
<th>A</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

- **A**: State of previous elemental volume (current row array, previous column), represented by column #4 in Table 3.
- **B**: State of previous elemental volume (previous row array, previous column), represented by column #5 in Table 3.
- **C**: State of previous elemental volume (previous row array, same column), represented by column #3 in Table 3.
- **D**: State of current elemental volume, represented by column #6 in Table 3. (to be determined)
- **x**: An element of the current row array, represented by column #2 in Table 3.
Table 3  Decision logic table

The state of the current elemental volume for rules 12 and 17 cannot be determined because the predefined conditions (columns 2 to 5 in Table 3) for both rules are incompatible with each other, i.e. the predefined conditions for both rules will never be met simultaneously. This table is readily applied to the example shown in Figure 3-12.
The results for all depths are summarized in Table 4. The application of the rules to some of the elemental volumes at depth 1 is presented here in detail. The process starts with element (row0,col.0) at depth 1. Since the element is at the low left-hand corner, no condition can be defined in column 3, 4, and 5 of Table 3. The current row array element is coded "0". Therefore, rule #1 applies. The rule concludes that elemental volume (a) in Figure 3-14 should be empty (the result is listed in column 6 of Table 3), i.e. volume (a) should be removed. Rule #19 is used to determine the state of the elemental volume (e) since the elemental volumes (d), (a), and (b) are all empty and the current row array element (row1,col.1) is coded "1". The rule concludes that the elemental volume (e) is full, i.e. it should be left uncut.
### Table 4  State of all the elemental volumes of the example part

The predefined conditions for rule *20 and *25 are met simultaneously regardless of the state of the current elemental volume. In both cases, an extra condition has been imposed in order to provide a unique state for the current elemental volume. The condition states that no interlocking features should be created regardless of the state of the current elemental volume. An interlocking feature is defined as a feature which has one of its edges touching the edges of other features. A geometry which has two interlocking, rectangular features is shown in Figure 3.5.7. With the introduction of the new condition, rule 20 and 25 are able to conclude that the state of the current elemental volume is always...
empty and full respectively. Because of the newly added condition, the algorithm presented in this section is not able to recognize any "interlocking" blocks which may exist in an approximated geometry.

![Interlocking feature 1](image1.png)

**Figure 3-15 Interlocking features**

After the state of all the elemental volumes has been determined by the interface software, the commands which describe the removal volumes of an approximated part geometry are created according to the syntax specified by the Natural Language Interface (NLI). An interface program was designed to read the state of each elemental volume and write the commands to a text file. The program starts by specifying both the number of elemental volumes in the three orthogonal directions and the actual physical dimensions of the elemental volumes. This information has already been obtained and stored in the DXF files. The syntax of the commands which define the stock geometry is described as follows:
Chapter 3 Integration of CAD/CAPP/CAM

\[
\begin{align*}
x & \ y & \ z \ \text{ELEMENTS\_FOR\_PART \ Partname} \\
d1 & \ d2 & \ d3 & \ldots \ dz \ \text{Z\_DIM\_FOR \ Partname} \\
d1 & \ d2 & \ d3 & \ldots \ dx \ \text{X\_DIM\_FOR \ Partname} \\
d1 & \ d2 & \ d3 & \ldots \ dy \ \text{Y\_DIM\_FOR \ Partname}
\end{align*}
\]

\[x, y, \text{and } z \text{ are the number of elemental volumes in } x, y, \text{ and } z \text{ directions respectively. } d1, \ d2, \ldots, \ d[x,y,z] \text{ specify the dimensions of the elemental volumes.}
\]

\[\text{Partname} \text{ is the name of the part defined by the user. The stock geometry of the sample part is defined as follows:}
\]

\[
\begin{align*}
3 & \ 3 & \ 4 \ \text{VOLUMES\_FOR \ Partname} \\
0.5E0 & \ 0.5E0 & \ 0.5E0 & \ 0.5E0 \ \text{Z\_DIM\_FOR \ Partname} \\
1.5E0 & \ 3E0 & \ 3E0 \ \text{X\_DIM\_FOR \ Partname} \\
1.25E0 & \ 1.5E0 & \ 1.5E0 \ \text{Y\_DIM\_FOR \ Partname}
\end{align*}
\]

\[\text{After the stock geometry has been defined, the program generates the commands which define the final geometry of the approximated part. The program starts by analyzing the elemental volumes bounded by the top two cross section planes (i.e. depth 1) and by rows 0 and 1. The syntax of the volume removing command is described as follows:}
\]

\[\text{LEFT\_RIGHT\_SLOT \ ON\_Z \ #Z \ ALONG \ #Y1 \ & \ #Y2 \ BETWEEN \ #X1 \ & \ #X2 \ Partname}
\]

\[#Z \text{ represents the depth index. } \#Y1 \text{ and } \#Y2 \text{ represent the lower and upper boundaries of the elemental volumes to be removed respectively. } \#X1 \text{ and } \#X2 \text{ represent the left and right boundaries of the elemental volumes to be removed respectively. Again, the example previously presented in this section is used to}
\]

demonstrate this command creating process. The example shows that the elemental volumes (a), (b), and (c) are empty, i.e. they have to be removed. Thus, the command which describes the removal of these volumes is defined as follows:

```
LEFT_RIGHT_SLOT ON_Z 3 ALONG 0 & 1 BETWEEN 0 & 3 Partname
```

The program now analyzes the elemental volumes bounded by rows 1 and 2 at depth 1. The example shows that the elemental volumes (d) and (f) are empty. Thus, the commands are defined as follows:

```
LEFT_RIGHT_SLOT ON_Z 3 ALONG 1 & 2 BETWEEN 0 & 1 Partname
LEFT_RIGHT_SLOT ON_Z 3 ALONG 1 & 2 BETWEEN 2 & 3 Partname
```

Similarly, the command which describes the removal of the elemental volumes (g), (h), and (i) is defined as follows:

```
LEFT_RIGHT_SLOT ON_Z 3 ALONG 2 & 3 BETWEEN 0 & 3 Partname
```

Table 5 lists all the commands which describe all the removal volumes of the sample part geometry.

The commands which define both the stock and the final geometry of the approximated part are written to a text file called the "NLI geometry file". It is stored temporarily in the database and will be retrieved later by the automated process planning system.
### Table 5  NLI-type commands for the example part

<table>
<thead>
<tr>
<th>Depth 1</th>
<th>Part name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LEFT_RIGHT SLOT ON Z 3 ALONG 0 &amp; 1 BETWEEN 0 &amp; 3 Partname</td>
</tr>
<tr>
<td>2</td>
<td>LEFT_RIGHT SLOT ON Z 3 ALONG 1 &amp; 2 BETWEEN 0 &amp; 1 Partname</td>
</tr>
<tr>
<td>3</td>
<td>LEFT_RIGHT SLOT ON Z 3 ALONG 1 &amp; 2 BETWEEN 2 &amp; 3 Partname</td>
</tr>
<tr>
<td>4</td>
<td>LEFT_RIGHT SLOT ON Z 3 ALONG 2 &amp; 3 BETWEEN 0 &amp; 3 Partname</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Depth 2</th>
<th>Part name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LEFT_RIGHT SLOT ON Z 2 ALONG 0 &amp; 1 BETWEEN 0 &amp; 3 Partname</td>
</tr>
<tr>
<td>2</td>
<td>LEFT_RIGHT SLOT ON Z 2 ALONG 1 &amp; 2 BETWEEN 0 &amp; 1 Partname</td>
</tr>
<tr>
<td>3</td>
<td>LEFT_RIGHT SLOT ON Z 2 ALONG 1 &amp; 2 BETWEEN 2 &amp; 3 Partname</td>
</tr>
<tr>
<td>4</td>
<td>LEFT_RIGHT SLOT ON Z 2 ALONG 2 &amp; 3 BETWEEN 0 &amp; 1 Partname</td>
</tr>
<tr>
<td>5</td>
<td>LEFT_RIGHT SLOT ON Z 2 ALONG 2 &amp; 3 BETWEEN 2 &amp; 3 Partname</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Depth 3</th>
<th>Part name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LEFT_RIGHT SLOT ON Z 1 ALONG 0 &amp; 1 BETWEEN 0 &amp; 3 Partname</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Depth 4</th>
<th>Part name</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td></td>
</tr>
</tbody>
</table>

### 3.5.3.2. Interface between CAPP and the Database

The software interface (b) is responsible for both transferring the NLI geometry file from the database to the automated process planning system and writing the optimal cutting and tool sequences generated by the planning system to the text files. The NLI geometry file generated by the CAD-database interface is written in a pure text format. To ensure that the automated process planning system can understand the content of the file, it is necessary that the file is written in the Forth-specific screen (with the file extension ".scr") format. The text-to-screen file conversion is accomplished by issuing a command called "T2S" which has been built into the Forth system. To initiate file conversion, the command "T2S" has to be typed manually at the command prompt in order to invoke the function. The system then prompts the user for the name of the NLI
geometry file to be converted (the default name of the NLI geometry file is STAT).

The automated process planning system was originally constructed by Seethaler [1]. A few sub-routines were added to the original system in an attempt to write the planning results to the text files. The results include:

- the optimal cutting sequence
- the indices of the boundaries of each combined volume
- the locations of the boundaries relative to the origin (0,0,0)
- the optimal tool sequence

The text files are stored in the database and will be retrieved later for NC part programming. Figure 3.5.8-9 shows the two text files which contain the planning results for the sample part. All dimensions shown in Figure 3.5.8 are expressed in inches. The points \((x_1,y_1,z_1)\) and \((x_2,y_2,z_2)\) are the lower left-handed and upper right-handed corner coordinates of the combined volume respectively. The file which contains the optimal tool sequence stores the indices of the tools defined in the tool library. Figure 3.5.9 shows that tools #3 and #2 are used to machine the first two combined volumes and the last two combined volumes respectively.
3.5.3.3. Interface between APT and the Database

The software interface (c) is responsible for retrieving the results obtained from the automated process planning system and then generating APT statements which describe tool paths and other machining parameters. The major tasks which the interface program performs include:

- retrieve the actual part geometry
- determine the tool's starting position for each combined volume
• determine optimal path subdivision for each combined volume
• determine optimal feeds and speeds
• generate tool paths

The retrieval of the actual part geometry is achieved interactively between the user and the drawing editor. Both the top surfaces of the approximated combined volumes defined by the process planning system and the cross sectional planes, which represent the top surfaces of the actual removal volumes, of the actual part geometry are redrawn automatically in AutoCAD's drawing editor by the interface software. The order in which the "approximated" top surfaces are displayed in the drawing editor follows the cutting sequence derived earlier by the process planning system. To help the user select the top surface of each actual removal volume, the top surfaces of both the approximated combined volume and the corresponding removal volume are positioned such that they appear to overlap with each other. The interface program then writes the corresponding APT geometry statements to the part programs. Figure 3-18 shows an example of the overlapped top surfaces. In this simple example, the top surface a-b-d-f-g should be picked.
Figure 3-18 Overlapped top surfaces of each combined volume

The tool's starting position is the point at which the tool enters the workpiece. Each combined volume has its own starting position. In a typical $2^{1/2}$-D milling operation the tool enters the workpiece either from the sides or from the top. The milling cutter which is capable of entering the workpiece from the top must have an edge which cuts to center. For simplicity, the system presented here does not allow the tool to enter the workpiece from the top. Instead, the tool enters the workpiece from the side where there is no material adjacent to that side except the material to be removed. To determine the side(s) from which the tool may enter the workpiece, the interface software employs a simple algorithm which uses the locations of the boundaries (see Figure 3-16) to determine the states (open or interfaced) of the faces of each combined volume automatically. In many cases, the states of the faces of a combined volume can be determined by simply comparing its boundary indices with the indices which represent the
extremities of the stock. A combined volume, for instance, is found to have an open front face if the index of one of its y boundaries equals zero. The faces which are not located on one of the stock's edges are considered closed. In some cases, however, the removal of a combined volume(s) may "open" some of the faces of the combined volumes which have yet to be removed. To determine the state of a face at any stage, the algorithm first records the boundary indices of the combined volumes which have already been removed in a set of arrays, and then determines if the new volume intersects with its predecessors. The two volumes are said to be intersecting if, for instance, the right (front) boundary of one volume and the left (rear) boundary of another volume share the same index. If the new volume intersects with its predecessors, the boundary indices of the new volume are compared with the boundary indices previously recorded in the arrays in order to determine the openness of its faces. If the face of a new combined volume is contained inside the area spanned by the boundaries of the previous combined volumes on the intersecting plane, then the face is said to be empty. On the other hand, if the new volume doesn't intersect with any one of its predecessors, no precedence relationship is found. In either case, the record is updated by including the boundary indices of the new combined volume in the arrays. Figure 3-19 shows the top surface of each approximated combined volume of the sample part. Table 6 shows the "openness" of the faces of each combined volume.
The strategies employed by the interface software to determine the states of the faces of both the combined volumes 1 and 4 are explained in detail. The arrays are empty before the combined volume 1 is removed. The combined volume 1 is bounded by the boundaries 2 and 3 in the x direction and by the boundaries 0 and 3 in the y direction (see Figure 3-16). Its boundary indices are then compared with the numbers of elemental volumes in x and y directions. It is concluded that the combined volume 1 has an open right face since its right boundary index is equal to the number of elemental volumes in the x direction. It
also has an open front and rear ends since its front and rear boundary indices are equal to 0 and the number of elemental volumes in the y direction respectively. The combined volume 4 originally had only one open side, before both the combined volumes 1 and 2 are removed. Before examining the states of its faces, the program has to determine if it intersects with its predecessors. It is shown in Figure 3-19 that its left and right boundaries are shared by both the combined volumes 1 and 2; therefore, it intersects with the combined volumes 1 and 2 and the x boundaries 1 and 2 representing the interfacing planes. The y and z boundaries of the combined volume 4 are then compared with the y and z boundaries of the combined volume 2. Table 7 summarizes the result of the "openness" testing of its left face.

<table>
<thead>
<tr>
<th></th>
<th>Combined Volume # 2</th>
<th>Combined Volume # 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Smallest Y Boundary Index</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>Largest Y Boundary Index</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>Smallest Z Boundary Index</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>Largest Z Boundary Index</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A4 ≥ A2</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B4 ≤ B2</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C4 ≥ C2</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D4 ≤ D2</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Decision: If all Yes ⇒ Left Face Open; otherwise ⇒ Interfaced

Table 7 Openness testing for one sample combined volume

The same procedure is employed to determine the state of every face of each combined volume. Once the states of the faces have been determined, the program has the option of choosing any one of the "open" sides of each combined volume to be the tool's starting position for that combined volume.
The optimization of widths, depths, feeds and speeds was implemented according to the analysis presented in chapters 2.3.2-5. The data required for the optimization includes the cut geometry and properties of the selected combination of tool, material, and machine. The depth of each combined volume can be obtained from the file shown in Figure 3-16. A database was developed to include properties of possible tools, materials, and machines. Table 8 lists the information which is available from the database. The machining database consists of the properties of five different tools, two types of material, and one machine. The exponents and constants in the expanded Taylor tool life equation should be defined for every tool/material pair. For simplicity, however, only one set of the parameters corresponding to the case of a standard HSS cutter and plain carbon steel workpiece is provided. The database was implemented in a pure text format in an attempt to ease the task of adding new data.

<table>
<thead>
<tr>
<th>Tool Data</th>
<th>Material Data</th>
<th>Machine Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of flutes</td>
<td>Specific cutting pressure (psi)</td>
<td>Rated power (hp)</td>
</tr>
<tr>
<td>Diameter (inch)</td>
<td>Hardness (BHN)</td>
<td>Overhead rate ($/min)</td>
</tr>
<tr>
<td>Flute length (inch)</td>
<td>Tool Life Data</td>
<td>Tool change time (min)</td>
</tr>
<tr>
<td>Shank length (inch)</td>
<td>Velocity exponential</td>
<td>Max. speed (rpm)</td>
</tr>
<tr>
<td>Cost ($)</td>
<td>Equiv. chip exponential</td>
<td>Min. speed (rpm)</td>
</tr>
<tr>
<td>Max. depth of cut (inch)</td>
<td>Axial depth exponential</td>
<td></td>
</tr>
<tr>
<td>Force ratio</td>
<td>Thermal cycle exponential</td>
<td></td>
</tr>
<tr>
<td>Max. feedrate (in/tooth)</td>
<td>Tool life constant</td>
<td></td>
</tr>
<tr>
<td>Min. feedrate (in/tooth)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max. stress (psi)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8 Machining database

Algorithms were developed to prepare the APT tool motion statements which represent the tool paths required to machine the combined volumes. The
tool paths generated for each combined volume resemble the paths required to remove the material from a pocket, except for the fact that not all boundaries of a combined volume are closed. The three commonly used volume-clearance routines are uni-directional cutting, staircase cutting, and spiraling. During a uni-directional cutting operation, the tool only cuts in one direction; between the metal removal motions, the tool is retracted to a specified height above the machined surface and shifted to a point vertically above its new starting position. After re-entering the workpiece, the cutting cycle repeats itself (see Figure 3-20a). The staircase cutting routine generates zig-zag like tool paths (see Figure 3-20b). The tool only enters the workpiece once and is not released from the machined surface until the job is finished. Compared to the uni-directional cutting, the staircase cutting routine allows the tool to spend far less time traversing in air; as a result, the amount of time required to finish the same component is reduced (for the same feed rate, of course). The spiral cutting routine, as the name suggests, generates spiral-like tool paths (see Figure 3-20c). Similar to staircase cutting, the tool only enters the workpiece once. In this work, only the staircase cutting routine was implemented. The uni-directional cutting routine was not considered because it is unlikely that it will generate the most economical tool paths. Instead of creating a general spiral cutting routine from scratch, a limited edition of such routine has been provided by the APT system. This routine, which is initiated by the standard APT "pocket" command,
is capable of producing spiral-like tool paths for machining a polygonal volume comprising up to 20 straight boundaries.

![Diagram of tool paths](image)

**Figure 3-20** The tool paths for (a) uni-directional, (b) staircase, and (c) spiral cutting

In this work, an APT-based staircase cutting routine was implemented using AutoLISP. It was soon realized, however, that the generation of staircase-like tool paths for machining a general volume would involve tremendously complex computations. To lessen the effort required in its initial development, it was decided that only a limited version of a general staircase cutting routine would be implemented. The routine is capable of generating zig-zag like tool
paths to machine the volume element bounded by exactly 4 boundaries. Among those 4 boundaries, at least 2 of them have to be parallel. The other boundaries can be straight line and arc segments, although in most cases semi-circular boundaries are the only non-straight boundaries which would be encountered by the system. The tool path specification involves a detailed step-by-step sequence of tool moves. The moves are made along the geometry elements which include the boundaries of each combined volume, the partitioning lines for the width subdivision of each combined volume, and the partitioning planes for the depth subdivision of each combined volume. Each boundary is identified by the side of the combined volume in which it is located. The right boundary of a combined volume (i.e. the face with the largest x boundary index), for instance, is named RIGHTn where the integer n indicates that the associated combined volume is the nth combined volume to be removed according to the optimal cutting sequence. In order to ensure optimality, the partitioning lines for the width subdivision have to be parallel to the longest boundary of the combined volume and twice the tool radius apart (see Figure 3-21). During the tool path generation, the boundaries and partitioning lines are designated as either the drive surfaces or check surfaces, depending on the direction in which the tool is moving. The tool's moves are controlled by the standard APT contour motion commands such as TLRGT (tool-right), TLLFT (tool-left), GOFWD (go forward), GOLFT (go left), and GORGT (go right). One should notice that full slotting in the zig-zag path generation does not guarantee that there will be no cusps left
along the boundaries (see Figure 3-22). In practice, the phenomenon of cusps has been avoided by executing a preceding pass along the boundaries. This capability, however, was not provided by the staircase cutting routine developed in this work.

![Diagram](image)

**Figure 3-21** The volume boundaries and partitioning lines for full slotting

![Diagram](image)

**Figure 3-22** Cusps left during a staircase cutting operation
If the depth of a combined volume exceeds the maximum axial depth of cut allowed by the tool, there is a need for multi-pass operation. The depth of a combined volume is divided into a number of smaller depths according to the analysis presented earlier. Each depth is defined by the standard APT "PLANE" statement. The number of APT motion statements required to describe the tool paths in a multi-pass operation is greatly reduced after realizing that the same set of APT motion statements which describe the tool motions in one depth can be used to describe the tool motions in the others. The APT part program generated by the interface program for the sample part is provided in Appendix A. The information on the tools, workpiece material, and the machine that was used in this example is given in Appendix B.
3.6. Experimental Investigation

Having previously outlined the structure and capability of the integrated CAD/CAPP/CAM system, it is now possible to illustrate the results of running the entire system for a more complicated part. The part geometry of interest is shown in Figure 3-23. It contains three square shoulders and a slot with a curved rear end. The part was designed to demonstrate that the system is able to derive the most optimal low level process plan for a part geometry which has non-orthogonal components.

The ACIS solid model of the part was drafted in AutoCAD. The functions which are needed throughout the entire CAD/CAPP/CAM process are available from the AutoCAD's pull-down menu which has been customized for this work. Before the system can extract the information about the topology of the part a cross sectional plane is drawn at each depth. The "sectioning" function is called from the pull-down menu (see Figure 3-24). The creation of the cross sectional planes is guided by an interactive dialogue between the computer and the user. The system first prompts the user to select the part volume to be analyzed. The dialogue continues by asking the user if all the necessary cross sectional planes have been created and prompting the user to properly locate the cross sectional plane if the answer to the previous question is "no". The cross sectional planes of the experimental part are shown in Figure 3-25. In this example, six cross sectional planes were created.
Figure 3-23  Experimental part geometry

Figure 3-24  AutoCAD pull-down menu - "Sectioning"
After all of the cross sectional planes have been created, the user has to decide whether there is any non-orthogonal geometry which needs to be "approximated". In this example, the slot has to be linearized. The "part linearization" function is, again, called from the pull-down menu (see Figure 3-24). The semi-circular entity is replaced by a straight line entity according to the analysis presented in chapter 3.4 (see Figure 3-26).

The coordinates of the approximated cross sectional planes are saved in DXF format. The parsers which were written to extract the boundary-related information of the part are called up by the user from the pull-down menu. These parsers are named "X boundaries", "Y boundaries", and "Z boundaries" in the menu (see Figure 3-24). The results generated by the parsers are summarized in Table 9.
Figure 3-26  Approximated cross sectional planes for the experimental part

<table>
<thead>
<tr>
<th></th>
<th>X - axis</th>
<th>Y - axis</th>
<th>Z - axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of boundaries</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Locations of the boundaries</td>
<td>6.0</td>
<td>8.0</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>4.5</td>
<td>6.5</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>4.0</td>
<td>1.25</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>2.5</td>
<td>0.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 9  X, Y, and Z boundary-related information of the experimental part

Figure 3-27  Part name defined in a dialogue box
The part name is defined by the user via a dialogue box (see figure 3-27). The function "Feature Extraction" is then called up from the pull-down menu (see Figure 3-28). This function is responsible for producing NLI-type statements which describe both the stock geometry and part details in the manner expected by the automated process planning system. The text file which contains these statements is shown in Figure 3-29.

```
3 4 4 VOLUMES_FOR W
0.75E0 0.5E0 0.25E0 0.5E0 Z_DIM_FOR W
1.5E0 3E0 1.5E0 X_DIM_FOR W
2.5E0 1.5E0 2.5E0 1.5E0 Y_DIM_FOR W
LEFT_RIGHT_SLOT ON_Z 3 ALONG 0 & 1 BETWEEN 0 & 3 W
LEFT_RIGHT_SLOT ON_Z 3 ALONG 1 & 2 BETWEEN 1 & 3 W
LEFT_RIGHT_SLOT ON_Z 3 ALONG 2 & 3 BETWEEN 1 & 2 W
LEFT_RIGHT_SLOT ON_Z 2 ALONG 0 & 1 BETWEEN 0 & 3 W
LEFT_RIGHT_SLOT ON_Z 2 ALONG 1 & 2 BETWEEN 2 & 3 W
LEFT_RIGHT_SLOT ON_Z 1 ALONG 0 & 1 BETWEEN 1 & 3 W
LEFT_RIGHT_SLOT ON_Z 1 ALONG 1 & 2 BETWEEN 2 & 3 W
```

Figure 3-29  NLI-type statements for the experimental part W
The text file is converted into the screen file which is then read by the automated process planning system. The tool, material, and machine data which are used to calculate results for the part W are outlined in Appendix B. The function "volume subdivision", which attempts to derive the most optimal cutting sequence, is called up from the pull-down menu (see Figure 3-28). The commands "1 LOAD" and "OPTIMIZE W" are entered manually at the Forth command prompt in order to start the planning procedure. The information on the tools, workpiece material, and the machine that was used in this example is given in Appendix B. The resulting cutting sequence and tool selections for the part W are presented in Figure 3-30. The result shows that the automated process planning system selects a total number of four combined volumes. The types of these combined volumes and the states of their faces are summarized in Table 10.

<table>
<thead>
<tr>
<th>Combined Volume #</th>
<th>Face</th>
<th>Left</th>
<th>Right</th>
<th>Front</th>
<th>Rear</th>
<th>Volume Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Closed</td>
<td>Open</td>
<td>Open</td>
<td>Closed</td>
<td>Front_Open_Right_Shoulder</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Closed</td>
<td>Open</td>
<td>Open</td>
<td>Closed</td>
<td>Front_Open_Right_Shoulder</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Closed</td>
<td>Closed</td>
<td>Open</td>
<td>Closed</td>
<td>Front_Open_Slot</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Open</td>
<td>Open</td>
<td>Open</td>
<td>Closed</td>
<td>Front_Open_Shoulder</td>
</tr>
</tbody>
</table>

*Table 10  Types of the combined volumes for the part W*
After the cutting sequence has been derived, the system has to retrieve the "real" geometry for each combined volume. This is done manually, as explained in the previous section. The function "APT Programming" is called up from the pull-down menu (see Figure 3-28). The geometry selection process is guided by an interactive dialogue between the system and the user. The system first asks the user to select an origin such that the top surface of each combined volume is placed properly in the drawing editor. The default origin is the lower left-handed corner of the part. The top surface of the each combined volume is
redrawn according to the optimal cutting sequence. The user then selects the top boundaries of each "real" combined volume. The selected boundaries for each combined volume are shown in Figure 3-31.

Figure 3-31  Top boundaries of the combined volume of the part W

After the boundaries of each combined volume have been selected, the system automatically calculates the optimal feeds and speeds based on the cutting geometry and generates the APT statements which describe the tool paths required to machine the combined volumes. Table 11 summarizes the
calculated feeds and speeds for each combined volume. Both the height to which the tool is retracted during rapid traversing and the cusp compensation factor are defined via the dialogue box shown in Figure 3-27. The cusp compensation factor is expressed in terms of the tool diameter; for instance, a factor of 1.0 introduces no cusp compensation, whereas a factor of 0.5 or below eliminates all the cusps except those located at the corners. The default values for both entries are 0.5 inches and 1.0 respectively. In the end, the APT geometry, motion, and post-processing statements for the combined volumes are written to the part program file.

The complete APT part program, which contains the APT statements which describe the tool paths and cutting parameters along the paths for making the part W, is given in Appendix C. The function "G-code Programming", which is accessible from the AutoCAD's pull-down menu, is responsible for generating the G-codes that will be sent to the milling machine. The tool paths are verified by the NC path "preview" function which is provided by Personal APT. The milling post-processor was configured for complete customization of the G-codes generated. The tool path (at each depth) for each combined volume is shown in Figure 3-32.
<table>
<thead>
<tr>
<th>Cut Geometry</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
</table>
| Full slot, Max. depth | Feed: 21 ipm  
Speed: 884 rpm | Feed: 21 ipm  
Speed: 884 rpm | Feed: 21 ipm  
Speed: 884 rpm | Feed: 21 ipm  
Speed: 884 rpm |
| Full slot, Rem. depth | Feed: 36 ipm  
Speed: 1500 rpm | Feed: 36 ipm  
Speed: 1500 rpm | Feed: 21 ipm  
Speed: 884 rpm | Feed: 36 ipm  
Speed: 1500 rpm |
| Rem. width, Max. depth | Feed: 36 ipm  
Speed: 1500 rpm | Feed: 36 ipm  
Speed: 1500 rpm | Feed: 36 ipm  
Speed: 1500 rpm | Feed: 28 ipm  
Speed: 1179 rpm |
| Rem. width, Rem. depth | Feed: 36 ipm  
Speed: 1500 rpm | Feed: 36 ipm  
Speed: 1500 rpm | Feed: 36 ipm  
Speed: 1500 rpm | Feed: 36 ipm  
Speed: 1500 rpm |

Table 11 Summary of feed and speed selections for the part W

Figure 3-32 Tool paths for the combined volumes of the part W
3.7. Dynamic Process Planning

The integrated CAD/CAPP/CAM system attempts to derive the optimal process plans for 2\( \frac{1}{2} \)-D milling components by embedding process-related knowledge into a traditional CAD/CAM system. The quality of the process plans, however, depends quite heavily on the accuracy of the process descriptions. The three levels of optimization in the low level process planning hierarchy usually rely on complete and reliable knowledge of the tool life, tool/material properties, and the parameters describing various process and constraint relationships. Unfortunately, such knowledge is seldom complete at the preprocessing stage. The usual procedure is to use the estimates of these parameters instead and as a result, more conservative and less cost effective process plans are generated. Recent advances in sensor technology and process identification techniques however have allowed researchers to consider the development of so-called "dynamic process planning systems". Such a system, if fully implemented, is capable of utilizing the process information obtained during the machining process to continuously update the process plans and ultimately, improve operating safety and process economics. Seethaler [1] took steps to achieve this goal by developing a prototype system which attempts to integrate identified process parameters into a CNC-based process planning system. His work and the author's contribution to the extension of the existing system will be discussed in the follow paragraphs.
Seethaler proposed the development of the integrated process planning and control system (IPAC) which attempts to integrate preprocessing, real time process identification and control, and dynamic process planning activities within the UBC open architecture CNC system. The UBC open architecture controller allows for multiple masters that can share tasks such as geometric modeling, process planning, and high level CNC control tasks. Its slave boards perform such tasks as real time axis control, process identification and control. The communication channels established between the slaves allow for synchronous execution of the axis position control tasks and process control tasks. The detail of the controller architecture has been described in numerous publications [27, 28] and therefore will not be reiterated here. The preprocessing tasks performed by the system include geometry input, low level process planning based on the estimates of process-related parameters, and G-code generation. A $2^{1/2}$-D milling part geometry is modeled as a set of rectangular volume elements via the NLI interface. As mentioned earlier, the process planning system possesses both preprocessing and real time capabilities. While the integrated CAD/CAPP/CAM system developed in this thesis uses the package for preprocessing activities, the real time capability of optimizing cutting parameters during the machining process has been demonstrated by Seethaler in his work. Within the controller architecture, the monitoring system collects force data from the dynamometer and performs identification of cutting geometry (i.e. axial depths and radial widths), machining parameters (i.e. specific cutting pressure
and force ratios), and runouts. The availability of these identified parameters reaps benefits on two levels. At the lower level, the machining operation becomes safer because the identified force levels and more accurate measurements of equivalent chip thickness improve the feed control allowing tool shank and edge breakage avoidance. At the same time, the machine operates more economically because more efficient control strategies can be adapted to the ever-changing machining environment. At the higher level, the quality of the existing process plan can be enhanced because improved estimates of the process parameters are more readily available. More importantly, however, process recovery from emergency situations such as tool collision and breakage becomes possible. The process planning system, after receiving reports of an emergency from the monitoring system, should be able to replan the cutting sequence, tool trajectories, and machining parameters (probably with a different set of tools), for the remaining part geometry. A fully functioning system which can perform the process replanning activities has yet to be implemented based upon Seethaler's work. Nonetheless, his work has provided the groundwork for the realization of such a system.

The major contribution made in this thesis to this point in extending IPAC is the provision of a geometric modeling interface which allows non-rectangular volume elements to be evaluated by the existing system. In the preprocessing stage, the interface transforms non-rectangular machined features into rectangular features according to the guidelines presented in Chapter 3.4. The
transformed features are then modeled according to the NLI specifications. The approximate costs of machining these transformed features are calculated based on the estimates of the process-related parameters and then compared. In the end, the most optimal cutting sequence is derived and appropriate machine instructions for the real part geometry are sent to the machine tool. During the machining process, IPAC is responsible for monitoring and optimizing the process in real time. In case of an accident, IPAC would stop the process immediately and send the reports which inform of both the nature of the accident and the identity of the volume element being machined at the time of the accident back to the integrated system. To recover, the geometric modeling interface developed in this thesis would attempt to identify the remaining part geometry and make changes to the original ACIS-based part model via the AutoCAD’s graphical front end. The machining process would then be replanned for the modified part model according to the procedure described in Chapter 3.6. If the accident is related to the instances where the tools have been damaged, the tool library would have to be updated to disregard the damaged tools and/or to include a new set of tools. In the end, the new part program would be sent to the machine tool to resume the operation.
3.8. Summary

This chapter presented the development of an integrated CAD/CAPP/CAM system. The system is primarily designed for 2½-D milling operations. It consists of three major components which include the AutoCAD solid modeling system, an automated process planning system, and Personal APT part programming system. The software interfaces were developed to allow smooth flow of information within the integrated system. The concept of non-orthogonal geometry approximation was proposed in an attempt to allow part geometry with a wider variety of machined features to be processed by the automated process planning system. In particular, the geometric modeling interface was developed to support circular/semi-circular geometry transformation only. The implementation of the software interfaces has been presented and their functions have been demonstrated with the aid of a few examples.

The potential contribution of the integrated CAD/CAPP/CAM system developed in this thesis to the realization of dynamic process planning has been briefly discussed. The integrated system is equipped with a process planning package which possesses both the preprocessing and real time capabilities. While the integrated CAD/CAPP/CAM system uses the package strictly for preprocessing activities, the process planning package's capability to extract process information during the machining operation for optimizing the process in
real time has been demonstrated previously by Seethaler. The work reported in this chapter has demonstrated that the use of the integrated CAD/CAPP/CAM system developed in this thesis together with the real time capability of the process planning system would greatly extend the utility of the approach.
4. Process Monitoring in a Network Environment

4.1. Introduction

The primary objective of the proposed research described in this chapter is to demonstrate the feasibility of allowing real time machining information to be accessible to both on-site and off-site production personnel in a network environment. A successful realization of this concept will reap benefits on two levels. First, the shop floor manager may keep track of the operation of, for instance, a manufacturing cell, by monitoring the performance of the CNC machines in the cell on a single computer screen using the shop floor computer network (e.g. the DNC network). Second, production/manufacturing engineers who do not work on the shop floor are allowed to "visually" monitor the machining operations as if they were on the shop floor. The ability to keep track of the machines' performance and usage while they are running allows the engineers to set up maintenance or tool replacement schedules in advance and to distribute work orders more efficiently. It also becomes possible for the system to automatically record the histories of various types of process information while the machines are running. Thus, the entire process could be "replayed" allowing the correction of process errors.

During the work described in this chapter NRPMS, a Network-based, Real time machining Process Monitoring System has been developed. The prototype
system was built around a low cost, commercially available 3-D virtual reality development package called Meme. NRPMS is capable of distributing real time animations of the machining process across a computer network and allowing remote process shutdown in case of emergency. Detailed discussion on NRPMS's structure and its hardware and software implementation is presented. Examples are given to demonstrate some of the system's capabilities.
4.2. Development of NRPMS

In a NRPMS, machining operations are brought to the user’s desk through the network with simulated, or graphical, representations. A NRPMS consists of four sub-systems: a Virtual Machining Environment (VME), a real time process data acquisition system, a distributed process monitoring system, and a remote process shutdown system. The prototype NRPMS was implemented for 2\(1/2\)-D milling operations and is able to keep track of the tool trajectories and the maximum resultant cutting forces during the machining operation. In the following sections, the implementation of its subsystems are presented in detail.

4.2.1. Development of the Virtual Machining Environment (VME)

The Virtual Machining Environment (VME) represents the core of NRPMS. The VME provides visual feedback of the machining operation by providing 3D animation of the changes in process data gathered during a machining operation. In this work, the VME is responsible for animating the changes of the machine table position and the maximum cutting force measurements obtained in real time during the machining operation. The VME creates and displays the graphical objects which represent the cutting tool (with the spindle), machined surfaces (i.e. drive and part surfaces), and the maximum resultant force measurements. More objects could be added to the VME later to make the simulated machining environment more realistic; they could include, for instance, the components of the machine tool structure such as the machine table and the
vertical column, fixtures, etc. Besides being responsible for displaying the process-related, 3-D graphical objects, the VME needs to provide a means of receiving, in real time, the tool coordinates and the force measurements from the CNC machine(s). These real time data are used to change the properties and behavior of the objects in the VME.

Meme, a virtual reality application development system by Immersive Systems, was chosen as the development platform for NRPMS. It is a low cost, PC-based software package which runs on 32 bit Windows operating systems from Microsoft. Meme is equipped with a rendering engine called RenderWare from Criterion Software to allow fast, realistic 3-D object rendering. It also provides interfaces to both serial and TCP/IP-typed networks. The programming language used to develop Meme applications is a stack-based interpretative language reminiscent of Forth. The system either runs the program file which contains all the program codes or scans the codes which are typed at the command prompt. Meme’s multi-processing mechanism allows the objects in the simulation environment to run as a set of background tasks while providing the programmer with a command prompt in the foreground. Orientation, size, and brightness are among the properties of an object which can be manipulated by typing commands at the command prompt. Results are reflected instantaneously in the simulation environment. In this work, the Meme’s multi-processing capability is particularly useful when a person who uses NRPMS to monitor the machining process on the computer screen may want to look at the part being
machined from a different angle to check the part dimensions, for instance. He/she may enter the necessary commands to alter both the zoom degree and camera view angle at the command prompt using the keyboard while the simulated machining process is being displayed.

There are two types of objects which exist in the VME. The first type is that of visible objects, including the virtual cutting tool, the facets which make up the drive and part surfaces, and the maximum resultant force indicator. The other type comprises the so called invisible objects which include camera and light objects. In Meme, a visible object is described by a shape file which is read at run time. The shape file conforms to the RenderWare's RWX format, which is a text-based graphics format for modeling 3-D objects on the computer screen. In this work, all the visible objects rendered and displayed in the VME were originally created in AutoCAD. Meme's built-in file converter was then used to convert the surface geometry created in Autodesk's AutoCAD into the RWX objects. The invisible objects in the VME are not described by shape files. The camera objects, for instance, only serve to establish one or more view points in the VME. No physical shape is defined for this object.

Each object in the VME has a data structure associated with it. A simplified data structure of an object (visible or invisible) is shown in Figure 4-1. The structure contains different fields which are used to define the state of an object. These attributes include position, orientation, and scale.
The coordinate system used is a right-handed cartesian set. The x, y, and z position fields hold the position that the object has with respect to a global origin. The yaw, pitch, and roll fields hold the rotation angles that set the orientation of the object. The angles are in degrees. The x, y, and z scale factors can be used to shrink or enlarge the object with respect to its size in the shape file. A scale factor of 1.0 leaves the object's size unchanged. The ptr field holds the address of the memory space occupied by the object's definition.

Subroutines have been provided by Meme to manipulate the state of the object [48].

The prototype VME which has been implemented in this work is shown in Figure 4-2. The size of the monitoring window can be adjusted according to the travel limits of the tool. The tool object navigates in x, y, and z directions within the VME while receiving the actual tool coordinates from the network interface (the implementation of the network is presented in the next section). The speed at which the tool object is moving in the VME is directly proportional to the feed.
rate at which the actual cutting tool is moving. Similarly, the force indicator object indicates the actual cutting force levels by altering its length and color while receiving force measurements from the network interface. This bar-type indicator serves as an example to demonstrate the capability of the virtual reality system to animate the changes of the process-related data in real time. In particular, the indicator was designed to alert the operator to possible damage of the tool and the part surfaces due to high feed rates and large axial depths. The length of the force indicator is directly proportional to the force level. The default color of the force indicator is green. If the force level exceeds the predefined limit, the indicator will turn red, which indicates either the feed rate has to be slowed down or the machine has to be stopped immediately. The facets which represent the drive and part surfaces are not generated within the VME unless the tool is actually cutting material. At this stage of the development, both the drive and part surface facets created in the VME can be oriented in the x or y directions only; the orientation of the surface facets could be corrected later for either diagonal or circular interpolation moves. The height of the drive surface facets and the width of the part surface facets should be dependent on the real time measurements of axial depth and tool immersion respectively. These measurements would be obtained from the cutting geometry identification process which have been implemented and demonstrated by Seethaler in his thesis [1]. In this work, however, real time monitoring of cutting geometry is not possible with NRPMS because the identification routines were not implemented
in the prototype system. The height of the drive surface facets and the width of
the part surface facets could either be fixed at predefined values or determined
by the process planning system. Two invisible objects were implemented in the
VME. The angle from which the visible objects in the VME were observed was
defined by a camera object. A light object was defined to provide a light source
for the VME.

![Image](image.png)

**Figure 4-2 Virtual Machining Environment (VME)**

### 4.2.2 Real Time Process Data Acquisition

To monitor a machining process using NRPMS, process data needs to be
sent continuously from the CNC machine to the server during the machining
process. In this work, a simple but efficient networking solution which uses a RS-
232 standard serial connection for real time process data transfer between the
CNC machine and the VME installed in the server was employed. To establish
the serial connection, both the sending end (the CNC software) and receiving end (the VME) had to be set up properly. At the sending end, a generic serial interface has been provided by the Forth-based software which controls the CNC machine. The interface is responsible for determining the port location, parity bit, stop bits and the baud rate for the serial connection. The process data which are sent to the VME include the coordinates of the machine table and the maximum resultant force measurements. The coordinates are expressed either in millimeters or inches and the force measurements are expressed in millivolts. The maximum resultant force measurements are collected from a dynamometer. Special routines were written to convert the floating point values of the position and force measurements to the corresponding string-type data which would be transmitted to the VME by the serial connection. At the receiving end, the VME provides support for a RS-232 serial connection. The set-up of the network interface is straightforward. The baud rate and other communication parameters are set by issuing the command "SerialOpen". Once the serial port has been initialized, string-type data can be read and written with the commands "SerialRead" and "SerialWrite" [48]. To utilize the string-type data which are read from the serial line, special routines were written to convert these string-type data back to the floating point numbers for numerical computation.
4.2.3. Distributed Process Monitoring

The distributed process monitoring system has been developed to enable the sharing of real time process information among computers which are connected to a local area computer network. This system aims to utilize the existing office-based network which has been used predominately for tasks such as DNC, resource sharing, and file transfer in a typical manufacturing facility. The implementation of such a system, which involves both the selection of the networking solution and the allocation of tasks to different components within the network, is presented in this section.

The prototype system consists of servers and workstations. These computers are connected to a local area network (LAN). Ethernet and IBM's token ring are the two physical networking technologies which have been commonly employed to implement a LAN. To comply with both networking technologies, the distributed process monitoring system was designed such that, as long as the network allows remote file access and sharing, the process data obtained during the machining operations can be distributed throughout the network regardless of the type of LAN being installed. Besides being connected to the LAN, each server is linked by serial cables to a number of CNC machines. The servers are responsible for receiving real time process information from the CNC machines via the serial connections (as described in Chapter 4.2.2) and sending the process information to the workstations which are connected to the
same network. Figure 4-3 shows the schematic of the overall system network architecture.

![Diagram of overall system network architecture]

**Figure 4-3 Overall system network architecture**

Networking software was needed to provide the remote file access and sharing service and other network management tools. In particular, the network operating systems which run in the workstations and servers and the communication protocol for network file sharing needed to be determined. A protocol is a special set of rules or procedures that make it possible for one computer to exchange data with another. In this work, Microsoft's Windows 95 and NT 4.0 Workstation were chosen as the operating systems for the workstations and servers respectively and their built-in "File and Printer Sharing
for Microsoft Networks’ services were used to provide remote file access and sharing between the servers and workstations. Either Microsoft’s Netbeui or the non-proprietary TCP/IP suite can be used as the primary network protocol for file sharing within local area networks. While the selected operating systems are delivered with both protocols, the TCP/IP suite can be brought in by other vendors’ products (e.g. NCSA’s Telnet, Chameleon, Trumpet Winsock, etc.). Netbeui is non-routable, which means this protocol can only be used if both the server and its workstations reside in the same local area network. If the server and its workstations are connected to different local area networks (within a wide area network, for instance), the routable TCP/IP has to be used for remote file access and sharing.

Besides being able to perform typical CAD/CAM activities, each server is dedicated to four general tasks: collecting real time process data from the CNC machine(s), representing process changes graphically, and updating process data and writing them to the data files. The first task has been discussed in Chapter 4.2.2 and is not further discussed here. A software program was written to automatically scan the serial network interface, retrieve the most recent measurements of the machine table position and the cutting force level, and represent the changes of those parameters graphically for a single machining operation. In this work, the VME was chosen to display the changes of both the tool position and the maximum cutting force on the server’s screen. It is possible for the VME to represent and display the changes of other information such as
the coolant level and tool condition when suitable sensors are available. In the case when the server is serially connected to more than one CNC machine within the shop floor network, the multi-tasking environment provided by the operating system allows multiple VMEs to be activated for multi-process monitoring. The process data which are written to the text-formatted data files during the machining operation include the x, y, and z coordinates of the machine table and the maximum resultant cutting force. These data files, which store the most recent measurements of the machine table position and the maximum resultant cutting force, reside in the server's local storage space and are distributed to the workstations on-demand to realize real time, distributed process monitoring. Further development of this prototype system would require these real time measurements (plus other measurements which may reveal the status of the on-going machining process) be collected and stored in large, separate storage devices (e.g. a separate hard disk, a ram drive, etc.) as the machining operation progresses so that the entire histories of these measurements could be recorded.

Each workstation has access to the process data which are collected by the server(s) during the machining operation(s). To reduce any unnecessary network traffic, the network only carries the real time, process-related data; the graphical animations (which involve 3-D object rendering in the VME) of any process changes are performed locally at each workstation. To remotely monitor a machining operation, the engineer needs to log onto the server which is
serially connected to the CNC machine which is being monitored from one of the available workstations. Once the server is successfully logged on, the user activates the VME and the software running inside the workstation then automatically opens the data files which reside in the server's storage space using the "File and Printer Sharing in Microsoft Networks" function. At this stage, these data files only contain the most recent measurements of both the tool position and the maximum cutting force collected during the machining operation. The software then extracts these measurements from those data files and animates their changes in the VME which has been installed in the workstation. Again, multiple VMEs can be activated to allow multi-process monitoring. While the machining process is being animated, the monitoring system allows on-line adjustment of the zoom degree and camera view angle with the keyboard. The remote emergency stop function enables the engineer who is responsible for monitoring the machining operation(s) remotely using the VME installed in the workstation to shutdown any malfunctioning machine immediately. More detail on its implementation is given in the next section.

4.2.4. Remote Process Shutdown

Although most modern CNC machines may have already been equipped with various types of safety measures to protect themselves from being seriously damaged due to accidents such as tool collision and tool breakage, a ultimate safety mechanism which allows the emergency shut-down of any machine under
any circumstances is still needed. In this work, a simple process stoppage mechanism which utilizes both the serial network previously established for real time process data acquisition (see Chapter 4.2.2) and the office-type network has been implemented to demonstrate the capability of NRPMS to allow remote shut-down of the machines that are being monitored by multiple VMEs installed in the workstations. The software which runs inside the server is responsible for not only scanning signals from the serial network interfaces, but also scanning for an emergency stop signal issued by the workstation. A Windows-based operator control console which provides a "one-click" button for issuing the emergency stop command has been implemented (see Figure 4-4).

![Figure 4-4 Operator Console for "Emergency Stop"](image)

When the red-colored "STOP" button is pressed, the emergency stop command will be initiated. The "File and Printer Sharing for Microsoft Networks" function bounded to TCP/IP or Netbeui allows the locally initiated emergency stop command to be written to the data file which resides in the server's storage. The software which runs in the server is continuously scanning the file as the
machining process progresses. Once the software verifies the command it is able to send a "shut-down" code directly to the CNC software via the serial cable. The CNC software interprets the code and issues an emergency stop command to stop the on-going machining operation. The use of the existing LAN technologies could also be extended to allow other machine tool functions to be controlled remotely; functions such as feed hold, feed rate override, etc. could be activated remotely in case of unmanned operations.
4.3. Experimental Investigation

Two examples are presented here to demonstrate some of the prototype NRPMS's capabilities. In this demonstration, the prototype NRPMS simulates a simple 2-D linear contouring operation and a simple multi-pass peripheral milling operation that creates a square shoulder. The animation of the changes of some of the process parameters is distributed across the local area network in real time in an attempt to realize distributed process monitoring. Snapshots of the process animation will be taken in an effort to explain the system's capabilities. Figure 4-5 shows the detailed schematic of the experimental system.
4.3.1. Laboratory Setup

The experimental system consists of four main components: a CNC vertical milling machine, a force collection system, a workstation, and a server. A Holke vertical milling machine, which has been retrofitted with the UBC open architecture machine tool controller, was used as the test machine. It is equipped with a three horsepower spindle with a manual, infinitely variable, belt transmission. Two 1.5kW Glentek dc-servo motors are used to drive the machine table in x and y directions, and the actuator in the axial direction is a 0.8kW Glentek dc-servo drive. The motors are matched with pulse-width modulated amplifiers from Glentek. The UBC controller is used to control the drives. An in-house designed pendant and PLC are used to provide an interface to the operator.

The force collection system collects the resultant cutting force data to allow real time cutting force monitoring within the VME. The dynamometer used in this work uses three piezo-electric force transducers to measure the three orthogonal components of the resultant cutting force. The unit is fastened between the workpiece and the steel plate that is clamped on the machine table. The output of the dynamometer is then fed through Kistler charge amplifiers and a KRON-HITE analog filter for signal conditioning. The output from the filter is sampled using a ROBOTROL RBX311 8-channel differential 12 bit A-to-D converter card that is connected to the SBX interface of the process monitoring
board. The force sampling is synchronized with spindle rotation. The output from a capacitive switch which triggers off a splined surface on the spindle is fed through a buffer to the external interrupt pin of the monitoring board. The monitoring board is a Universal Systems STD UE9001A card that is based on the Intel 80c196 CPU with a clock speed of 16-MHz. Once an interrupt is initiated by the switch a force sample is collected, and 60 force samples are collected per spindle revolution. The average value of the force samples collected per spindle revolution is calculated and the result is then placed in the bi-port RAM of the monitoring board. The main processor board then extracts the force value from the bi-port RAM of the monitoring board via the backplane bus, and sends the force value to the server computer using the RS-232 serial connection.

The workstation is responsible for obtaining the most recent machine table's position and maximum cutting force measurements from the server, and animating their changes in real time in the VME. It is a Pentium 90-MHz personal computer with 32 Mbytes of RAM. The operating system is Microsoft's Windows 95. The connection to the local area network is achieved by an ISA-based Intel EtherExpress Pro LAN ethernet adapter. The graphics accelerator is an ATI Graphics Pro Turbo Mach 64 video card with 2 Mbytes of RAM.

The server collects the machine table's position and maximum cutting force measurements from Holke, animates their changes in real time in the VME,
and records the most recent measurements on the data files. It is a Pentium 200-MHz MMX personal computer with 32 Mbytes of random-access memory. The operating system is Microsoft's Windows NT 4.0 Workstation. The connection to the local area network is achieved by an ISA-based Intel EtherExpress Pro LAN ethernet adapter. The graphics accelerator is an ATI Graphics Pro Turbo Mach 64 video card with 2 Mbytes of RAM. The connection from one of its COM ports to the serial interface of the UBC controller is achieved by a RS-232 serial cable. This serial connection was set to transmit data at 19200 bits per second.

4.3.2. Case Study I: 2-D Linear Contouring

The first example demonstrates NRPMS's ability to both animate the tool/machine table's movement in a simple 2-D contouring operation in the VME and alert the operator to excessive cutting force experienced during the operation by the force indicator displayed in the VME. The pre-planned tool path is shown in Figure 4-6. The feed rate override was set at 50% to allow the manipulation of the feed rate during the operation. The resultant force (in millivolts) limit above which the force indicator turns red was set at a lower value (2000 millivolts, which corresponds to around 89 lbfs) on purpose such that the limit could be easily exceeded. The milling tool used in this experiment was a three-tooth, high-speed steel (HSS) cutting tool with a diameter of 0.5 inches. The workpiece chosen for this experiment is made of an aluminum alloy.
The snapshots of the animation which are shown here were taken at the server. Identical images would, however, be obtained if the snapshots were taken at the workstation. Figure 4-7a shows that the tool was currently located at the cold start position (i.e the machine zero position) and that the length of the force indicator was in its default, or non-cutting, state. The camera view angle was set up such that both the drive and part surfaces could be observed. Three snapshots were taken at the positions arbitrarily chosen along the path while the tool was cutting the workpiece (see Figure 4-7).

When the tool started cutting, both the drive and part surfaces were created and the length of the force indicator changed. The force constraint was not violated when the tool was at positions (b) and (c). The feed rate override was then raised gradually until the force constraint was violated. The snapshot which was taken when the tool was at position (d) shows that the force indicator turned red, which means the force constraint had been violated. The feed rate
override was then reduced manually until the force indicator turned back to green. Figure 4-8 shows that the tool had already returned to its starting position and the force indicator had returned to its default (non-cutting) length.

Figure 4-7 Snapshots of the tool movement in case study (I)
4.3.3. Case Study II: Multi-Pass Peripheral Milling

The second experiment demonstrates NRPMS's ability to simulate a simple multi-width and depth peripheral milling operation in the VME. A volume clearance routine was developed to create a square shoulder. The tool path created by this routine was of the staircase-type. The isometric view of the pre-planned tool path is shown in Figure 4-9. Two passes were needed to create the shoulder. The milling tool used in this experiment was a three-tooth, HSS cutting tool with a diameter of 0.5 inches. The workpiece chosen for this experiment is made of an aluminum alloy. Again, snapshots of the animation displayed in the VME were taken at the workstation to show various stages of the operation.
Before the process started the tool remained at the cold start position and the length of the force indicator was in its default, or non-cutting, state. The camera view angle was set up such that both the drive and part surfaces could be observed. Figure 4-10a shows that the tool had already started the operation and was located at position (a). Both the drive and part surfaces were being created. After the tool turned at the end of the workpiece (position (b)), the previously created drive surface was being removed while the new drive surface was being created in an attempt to simulate the zig-zag volume clearance routine (see Figure 4-10b). It should be noted, however, that the drive surface wouldn't have been removed if the lateral movement of the tool exceeded the diameter of the cutting tool.
Figure 4-10c shows that both the part and the "second" drive surfaces remained after the tool had finished the first pass. The tool was then retracted to its starting position and re-entered the workpiece at a greater depth. The new drive and part surfaces were being created while the tool was cutting the workpiece. In the mean time, the old part surface was being removed (see
Figure 4-10d). Again, after the tool turned at the end of the workpiece, the previously created drive surface was being removed while the new drive surface was being created. At the end of the multi-pass operation, the part surface created during the second pass and the two sets of drive surfaces remained in the VME (see Figure 4-11).

Figure 4-11  Multi-pass peripheral milling operation
4.4. Process Monitoring via Internet

Chapter 4.2.3 has discussed the feasibility of using the file sharing function to enable the sharing of real time process data among the workstations which are connected to the local area network. The prototype system which allows real time distributed process monitoring has been implemented and demonstrated in the previous sections. In fact, the same function could allow real time, process-related information to be shared among the computers that have access to the internet. The process information which would be distributed over the internet may include the usage of the machines, the cost of the current jobs, and other information that may reveal both the tools' and machines' health. These types of information may only need to be updated and captured once in a while (e.g. 30 seconds) by the servers. The files which contain the most recent version of these data are kept in the servers' storage space. The "File and Printer Sharing for Microsoft Networks" function provided by Microsoft's operating systems then allows remote access and sharing of those data files on the internet. Netbeui cannot be used for file sharing on the internet because it is a non-routable protocol. Instead, the routable TCP/IP suite is usually chosen as the primary networking standard to deliver this service. Tailor-made software programs could be created and installed in the remote computers to interpret the up-to-the-minute process information available from the shared files and display the results on the remote computers' screens. Thus, the authorized personnel
who have access to the servers could monitor the performance of the machines from anywhere in the world as if they were sitting in the control room.

Since the internet is publicly shared and its bandwidth varies geographically, it becomes very difficult to prevent the process of sharing real time information across the internet from being delayed by unpredictable network traffic flow. This would undoubtedly present a serious problem for the development of the system which attempts to distribute real time process animation over the internet as the timely delivery of the real time measurements of the tool trajectories and cutting force to the remote terminals may be difficult. Therefore, a slightly different approach has been proposed such that the realistic animation of the machining processes could still be generated by the remote terminals in a "near-real-time" fashion. In this approach, the servers on the shop floor would first record the real time process data collected during a short time interval (e.g. 10 seconds) on the data files. These data files, which contain the brief history of process information, would be then transmitted (e.g. using FTP) to the remote terminals via the internet. The VME which has been installed in each remote terminal would open the data files, extract the process information from those files, and create the animation. In the mean time, the old data files would be updated with the newly recorded process data collected by the servers and then waited to be transmitted to the remote terminals.
At this stage of the development, the NRPMS's capability of providing the real time process monitoring service over the internet has yet to be implemented. However, the author believes that enough groundwork has been laid in place for making NRPMS "internet-ready" in the future.
Chapter 4  Process Monitoring in a Network Environment  144

4.5. Security Issue

NRPMS's ability to allow remote monitoring and control (shutdown) of ongoing machining operations has created some security concerns. To prevent any unauthorized access to the servers from the network-ready workstations, a reliable access control scheme needs to be employed. The simplest security measure is to control the access to the resources and capabilities provided by the servers using passwords. In this scheme, a person is either authorized, or not allowed, to use any resources provided by the servers. To provide more flexibility a more complicated user-level access control scheme which specifies users and groups who have access to certain system resources and capabilities could be employed. This scheme allows some personnel to log onto the servers remotely from any workstations to both monitor and control on-going machining operations whereas others may only be allowed to use the system's monitoring capability. During the log-on process, the user's password needs to be verified by the system's security checking facility to determine which resource(s) is accessible to the user. The same security measure could also be applied when the someone tries to log onto one of the servers from the internet. For the safety reason, however, it is recommended that the emergency process shutdown command should not be initiated using internet connections.

The prototype NRPMS which has been demonstrated in Chapter 4.3 uses a simple access control scheme provided by Microsoft's Windows NT operating
system as the only security measure implemented in the system. This scheme allows authorized personnel to log onto the server from the workstation, share process information collected by the server, and initiate the emergency process shutdown command from the workstation. The workstation, however, is accessible to everyone. The same security measure has been implemented for controlling access to the server from the internet.
4.6. Summary

This chapter investigated the feasibility of allowing process information collected during the machining operation to be distributed across a computer network to production personnel regardless of their geographical locations. In an attempt to realize this concept, a Network-based, Real time machining Process Monitoring System (NRPMS) has been developed in this thesis. The prototype system was designed to transmit both the tool position and maximum resultant cutting force measurements that are collected during the machining operation to the workstations connected to the local area network. The prototype system also allows the process shutdown commands to be initiated remotely from those workstations in case of an accident. A commercially available virtual reality development system was used to develop the Virtual Machining Environment (VME) which has been installed in each of the workstations that are used for process monitoring. The VME attempts to represent the machining operation virtually on the computer screen. The responsibilities of the VME include reading and interpreting the real time process data received from the CNC machines and animating their changes. The VME was designed to simulate the real-life shop floor environment. The prototype NRPMS also features a simple access control scheme which only allows authorized personnel to gain access to the server from the locally connected workstations. The system potentially allows the real time process information collected during the machining operation(s) to be distributed across the internet to the remote computers.
5. Conclusions

5.1. Research Summary

This thesis has investigated new strategies in an attempt to improve the quality of the machining operation at its preprocessing as well as real time stages. The first part of this thesis has examined the possibility of embedding low level process knowledge into a commercially available CAD/CAM system. To demonstrate the benefits of an integrated approach to the part production, a PC-based, integrated CAD/CAPP/CAM system which attempts to encompass the design, planning, and the manufacture of the part within a highly integrated environment has been developed. The prototype system was primarily designed for $2^{1/2}$-D milling operations, and attempts to automate both the low level process planning and tool path programming activities based on the geometrical model of the part. The concept of non-orthogonal geometry approximation was proposed in an attempt to allow part geometry with a wide variety of machined features to be processed by the process planning system which was designed primarily for prismatic components. The process planning system extracts and interprets the approximated part geometry from the DXF representation of the part, and derives the optimal operation sequence based on the approximated geometry. The results of the tool path planning and the machining parameter optimization which are produced based on the real geometry of the part are represented by a number of APT statements. These APT statements are post-processed to
generate the G-codes which are then transferred to the machine tool. A sample part which is composed of a number of linear and non-orthogonal removal volumes was used as a demonstration to showcase the capability of the integrated CAD/CAPP/CAM system developed in this thesis. At the end of the discussion, the author has suggested that the use of the integrated CAD/CAPP/CAM system developed in this thesis together with the real time capability of the process planning system would allow process re-planning following a tool failure or other unexpected circumstances which cause the stoppage of the process.

The second part of this thesis has examined the possibility of utilizing the office-type network which has already been established in a typical production facility to provide authorized personnel who are not in the close proximity to the machine tools an opportunity to "virtually" monitor the machining operations as if they were on the shop floor. To partially realize this concept, a network-based, real time machining process monitoring system (NRPMS) for $2 \frac{1}{2}$-D milling operations has been developed in this thesis. The system was implemented on low cost personal computers, and is capable of transmitting both the tool position and maximum resultant cutting force measured by a dynamometer during the milling operation not only to the computers connected to the local factory-wide network, but potentially to all of the computers that are internet-ready, provided that the local factory-wide network can be entered via internet connections. Both the tool motion and the change in force values are animated
graphically by a commercially available virtual reality development system and are displayed on the computer screens. The system is equipped with a machine tool shutdown mechanism which allows the process shutdown commands to be initiated remotely via the local area network in case of an accident. Two examples have been presented to demonstrate the system's capabilities to display the virtual tool motions and machined surfaces, and the maximum cutting force measurements during linear contouring and volume clearance operations.

5.2. Future Work

One of the contributions made in this thesis demonstrates the feasibility of integrating machining process knowledge with commercially available CAD and CAM systems. There are, however, a number of improvements that are required for a practical system. In this work, algorithms have been derived to extract only the geometric information of a part from its DXF representation; the provision of an interface for extracting the associated technological data and then carrying it over to the process planning system, however, has not been realized. The lack of support for this type of data stems from the way the technological data is associated with a part geometry in the CAD system. It may be necessary, in the author's view, to modify the CAD system itself to provide better support for the representation of technological information. Up to this point, only volumes with circular/semi-circular edges can be dealt with by the non-orthogonal geometry approximation algorithm implemented in this work. Future research should attempt to expand the possible volume types in order to enable the integrated
system to handle other non-orthogonal components (e.g. triangular holes and bosses). Finally, the possible contribution of the integrated CAD/CAPP/CAM system developed in this work to the realization of dynamic process planning especially in the cases where process recovery is needed has yet to be demonstrated.

There are still a number of areas that need to be dealt with to enhance the practicality of NRPMS. The only pieces of information that the prototype NRPMS can distribute across the network are the maximum cutting force measurements collected by the dynamometer and the tool locations. With the availability of suitable sensing equipment, other information such as the tool wear data, ambient temperature measurements, and vibration levels could be captured and distributed on-line. Currently, the widths and depths of the facets which represent both the part and drive surfaces in the virtual environment are either fixed or obtained from the solutions to the path planning problem. The real time measurements of the actual width and depth of cuts are not available because the monitoring system which collects force data from the dynamometer and performs the identification of cutting geometry (i.e. axial depths and radial widths) has yet to be implemented in this work. Although enough groundwork has been laid in place to allow NRPMS to distribute real time process information across the internet, the actual coding has yet to be written.
References


References


References


Appendix A: APT Program for the Sample Part *Partname*

Below is a listing of the APT statements generated by the interface program for the part *Partname* shown in Figure 3-11 in pp. 76.

```
PARTNO APT PART PROGRAM Partname
MACHIN/MMPOST,1
PRINT/ON
READ/2,DMSSYN
draft/BAKGRD,GRAY,GEOMTY,cyan,MOTION,white,intens,tool,yellow,ARROWS,ON
DRAFT/DRAW, -1,-1,-2,10,3,2
REDEF/ON

LEFT1=LINE/XCOORD,4.5
TOP1=LINE/YCOORD,4.25
RIGHT1=LINE/XCOORD,7.5
BOTTOM1=LINE/YCOORD,0.0
mx1=matrix/transl,0,0,2
tracut/mx1

DEPTH1=1
PASSES1=5
Depth_Inc1=0.2
XPASS1=1

PL0=PLANE/0,0,1,0.5
CUTDIA1=1
cutter/CUTDIA1
CUTLIM1=0.2
CUSPDIA1=1
Locstart1=point/6,2.125,0.5
From/Locstart1

FEEDMR1=21
SPMR1=884
FEEDMF1=21
SPMF1=884
FEEDRR1=21
SPRR1=884
FEEDRF1=21
SPRF1=884

AGA11 : PLA1=PLANE/0,0,1,(-Depth_Inc1*XPASS1)
JUMPTO/LOOP1
AGA12 : PLA1=PLANE/0,0,1,-DEPTH1
FEEDMR1=FEEDRR1
SPMR1=SPRR1
FEEDMF1=FEEDRF1
SPMF1=SPRF1
JUMPTO/LOOP1

LOOP1 : rapid,goto/7,-1,0.5
```
Appendix A

APT Program for the Sample Part "Partname"

RAPID,GO/PLA1
RAPID,GO/TO,RIGHT1,PLA1,TO,BOTTOM1
FEDRAT/FEEDMR1,IPM
SPINDL/SPMR1
TLLFT,GOFWD/RIGHT1,TO,TOP1
LY=LINE/PARLEL,RIGHT1,XSMALL,(1.5*CUSPDIA)
TLLFT,GOLFT/TOP1,ON,LY
TLON,GOLFT/LY,TO,1,INTOF,BOTTOM1

TLRG,TORG/BOTTOM1,TO,LEFT1
FEDRAT/FEEDMF1,IPM
SPINDL/SPMF1
TLRG,TORG/LEFT1,TO,TOP1
RAPID,GO/PL0
XPASS1=XPASS1+1
IF (PASSES1-XPASS1) QUIT1, AGA12, AGA11
QUIT1: RAPID,GOTO/Locstart1
TRACUT/NOMORE

LEFT2=LINE/XCOORD,0.0
TOP2=LINE/YCOORD,4.25
RIGHT2=LINE/XCOORD,1.5
BOTTOM2=LINE/YCOORD,0.0
mx2=matrix/transl,0,0,2
tracut/mx2

DEPTH2=1
PASSES2=5
Depth_Inc2=0.2
XPASS2=1:

PL0=PLANE/0,0,1,0.5
CUTDIA2=1
cutter/CUTDIA2
CUTLIM2=0.2
CUSPDIA2=1

Locstart2=point/0.75,2.125,0.5
GOTO/Locstart2

FEEDMR2=21
SPMR2=884
FEEDMF2=36
SPMF2=1500
FEEDRR2=21
SPRR2=884
FEEDRF2=36
SPRF2=1500

AGA21 : PLA2=PLANE/0,0,1,(-Depth_Inc2*XPASS2)
JUMPTO/LOOP2
AGA22 : PLA2=PLANE/0,0,1,-DEPTH2
FEEDMR2=FEEDRR2
SPMR2=SPRR2
FEEDMF2=FEEDRF2
SPMF2=SPRF2
JUMPTO/LOOP2
Appendix A  APT Program for the Sample Part "Partname"

1

LOOP2 : rapid,goto/0.5,5.25,0.5

RAPID,GO/PLA2
RAPID,GO/TO,LEFT2,PLA2,TO,TOP2
FEDRAT/FEEDMR2,IPM
SPINDL/SPMR2
TLLFT,GOFWD/LEFT2,TO,BOTTOM2
TLLFT,GOLFT/BOTTOM2,TO,RIGHT2
FEDRAT/FEEDMF2,IPM
SPINDL/SPMF2
TLLFT,GOLFT/RIGHT2,TO,TOP2

RAPID,GO/PL0
XPASS2=XPASS2+1
IF (PASSES2-XPASS2) QUIT2, AGA22, AGA21
QUIT2: RAPID,GOTO/Locstart2
TRACUT/NOMORE

LEFT3=LINE/XCOORD,0.0
TOP3=LINE/YCOORD,1.25
RIGHT3=LINE/XCOORD,7.5
BOTTOM3=LINE/YCOORD,0.0
mx3=matrix/transl,0,0,2
tracut/mx3

DEPTH3=1.5
PASSES3=8
Depth_inc3=0.2
XPASS3=1

PL0=PLANE/0,0,1,0.5
CUTDIA3=0.75
cutter/CUTDIA3
CUTLIM3=0.2
CUSPDIA3=0.75
Locstart3=point/3.75,0.625,0.5

GOTO/Locstart3

FEEDMR3=24
SPMR3=1500
FEEDMF3=24
SPMF3=1500
FEEDRR3=24
SPRR3=1500
FEEDRF3=24
SPRF3=1500

AGA31 : PLA3=PLANE/0,0,1,(-Depth_inc3*XPASS3)
JUMPTO/LOOP3
AGA32 : PLA3=PLANE/0,0,1,-DEPTH3
FEEDMR3=FEEDRR3
SPMR3=SPRR3
FEEDMF3=FEEDRF3
SPMF3=SPRF3
JUMPTO/LOOP3

LOOP3 : rapid,goto/-0.75,0.38,0.5
Appendix A  

APT Program for the Sample Part "Partname"

RAPID.GO/PLA3
RAPID.GO/TO,BOTTOM3,PLA3,TO,LEFT3
FEDRAT/FEEDMR3,IPM
SPINDL/SPMR3
TLLFT,GOFWD/BOTTOM3,TO,RIGHT3
TLLFT,GOLFT/RIGHT3,TO,TOP3
FEDRAT/FEEDMF3,IPM
SPINDL/SPMF3
TLLFT,GOLFT/TOP3,TO,LEFT3

RAPID.GO/PL0
XPASS3=XPASS3+1
IF (PASSES3-XPASS3) QUIT3, AGA32, AGA31
QUIT3: RAPID,GOTO/Locstart3
TRACUT/NOMORE

LEFT4=LINE/XCOORD,1.5
TOP4=LINE/YCOORD,4.25
RIGHT4=LINE/XCOORD,4.5
BOTTOM4=LINE/YCOORD,2.75
mx4=matrix/transl,0,0,2
tracut/mx4

DEPTH4=0.5
PASSES4=3
Depth_Inc4=0.2
XPASS4=1

PL4=PLANE/0,0,1,0.5
CUTDIA4=0.75
cutter/CUTDIA4
CUTLIM4=0.2
CUSPIDA4=0.75

Locstart4=point/3,3.5,0.5
GOTO/Locstart4

FEEDMR4=24
SPMR4=1500
FEEDMF4=24
SPMF4=1500
FEEDRR4=24
SPRR4=1500
FEEDRF4=24
SPRF4=1500

AGA41 : PLA4=PLANE/0,0,1.0,(-Depth_Inc4*XPASS4)
JUMPTO/LOOP4
AGA42 : PLA4=PLANE/0,0,1.0,-DEPTH4
FEEDMR4=FEEDRR4
SPMR4=SPRR4
FEEDMF4=FEEDRF4
SPMF4=SPRF4
JUMPTO/LOOP4

LOOP4 : rapid,goto/0.75,3.13,0.5
RAPID.GO/PLA4
RAPID.GO/TO,BOTTOM4,PLA4,TO,LEFT4

Combined
Volume 4
APT Program for the Sample Part "Partname"

FEDRAT/FEEDMR4, IPM
SPINDL/SPMR4
TLLFT, GOFWD/BOTTOM4, TO, RIGHT4
TLLFT, GOLFT/RIGHT4, TO, TOP4
FEDRAT/FEEDMF4, IPM
SPINDL/SPMF4
TLLFT, GOLFT/TOP4, TO, LEFT4

RAPID, GO/PL0
XPASS4=XPASS4+1
IF (PASSES4-XPASS4) QUIT4, AGA42, AGA41
QUIT4: RAPID, GOTO/Locstart4
TRACUT/NOMORE

SPINDL/OFF

END
FINI
Appendix B: Tool, Material, and Machine Data

The following tables summarize the tool, tool life, workpiece material, and the machine data that were used in the examples presented in Chapter 3.

<table>
<thead>
<tr>
<th>Tool #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Material</td>
<td>HSS end mill</td>
<td>HSS end mill</td>
<td>HSS end mill</td>
<td>HSS end mill</td>
<td>HSS end mill</td>
</tr>
<tr>
<td>Diameter (in)</td>
<td>0.50</td>
<td>0.75</td>
<td>1.00</td>
<td>1.50</td>
<td>2.00</td>
</tr>
<tr>
<td>Max. Depth of Cut (in)</td>
<td>0.05</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td># of Teeth</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>Flute Length (in)</td>
<td>0.9675</td>
<td>1.500</td>
<td>1.6875</td>
<td>2.1875</td>
<td>2.75</td>
</tr>
<tr>
<td>Shank Length (in)</td>
<td>2.25</td>
<td>3.0</td>
<td>3.75</td>
<td>4.25</td>
<td>4.75</td>
</tr>
<tr>
<td>Force Ratio</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Max. Feed (ipt)</td>
<td>0.004</td>
<td>0.004</td>
<td>0.004</td>
<td>0.004</td>
<td>0.004</td>
</tr>
<tr>
<td>Min. Feed (ipt)</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>Max. Stress (psi)</td>
<td>180000</td>
<td>180000</td>
<td>180000</td>
<td>180000</td>
<td>180000</td>
</tr>
<tr>
<td>Cost ($)</td>
<td>10.1</td>
<td>18.0</td>
<td>32.5</td>
<td>61.4</td>
<td>107.8</td>
</tr>
</tbody>
</table>

Table 12 Tool data

<table>
<thead>
<tr>
<th>Tool Life Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Velocity Exponent</td>
<td>2.0</td>
</tr>
<tr>
<td>Equivalent Chip Exponent</td>
<td>1.0</td>
</tr>
<tr>
<td>Axial Depth Exponent</td>
<td>0.5</td>
</tr>
<tr>
<td>Thermal Cycling Exponent</td>
<td>2.0</td>
</tr>
<tr>
<td>Taylor Equation Constant</td>
<td>180000000</td>
</tr>
</tbody>
</table>

Table 13 Tool life data

<table>
<thead>
<tr>
<th>Machine Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Power (horsepower)</td>
<td>3.0</td>
</tr>
<tr>
<td>Overhead Rate ($/min)</td>
<td>0.05</td>
</tr>
<tr>
<td>Tool Change Time (min)</td>
<td>2</td>
</tr>
<tr>
<td>Max. Speed (rpm)</td>
<td>1500</td>
</tr>
<tr>
<td>Min. Speed (rpm)</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 14 Machine data

<table>
<thead>
<tr>
<th>Material Parameters</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Aluminum Alloy</td>
</tr>
<tr>
<td>Specific Cutting Pressure (psi)</td>
<td>280000</td>
</tr>
<tr>
<td>Hardness (BHN)</td>
<td>85</td>
</tr>
</tbody>
</table>

Table 15 Material data
Appendix C: APT Program for the Part W

Below is a listing of the APT statements generated by the interface program for the part W shown in Figure 3-23 in pp. 100.

```
PARTNO APT PART PROGRAM W
MACHIN/MMPOST.1
PRINT/ON
READ/2,DMSSYN
draft/BAKGRD,GRAY,GEOMTY,cyan,MOTION,white,intens,tool,yellow,ARROWS,ON
DRAFT/WINDOW,-1,-1,-2,10,3,2
REDEF/ON

LEFT1=LINE/XCOORD,1.5
TOP1=LINE/YCOORD,2.5
RIGHT1=LINE/XCOORD,6.0
BOTTOM1=LINE/YCOORD,0.0
mx1=matrix/transl,0,0,2
tracut/mx1

DEPTH1=1.25
PASSES1=7
Depth_Inc1=0.2
XPASS1=1

PL0=PLANE/0,0,1,0.5
CUTDIA1=1
cutter/CUTDIA1
CUTLIM1=0.2
CUSPDIA1=1

Locstart1=point/3.75,1.25,0.5
From/Locstart1

FEEDMR1=21
SPMR1=884
FEEDMF1=36
SPMF1=1500
FEEDRR1=36
SPRR1=1500
FEEDRF1=36
SPRF1=1500

AGA11 : PLA1=PLANE/0,0,1,(Depth_Inc1*XPASS1)
JUMPTO/LOOP1
AGA12 : PLA1=PLANE/0,0,1,-DEPTH1
FEEDMR1=FEEDRR1
SPMR1=SPRR1
FEEDMF1=FEEDRF1
SPMF1=SPRF1
```

Combined Volume 1
JUMPTO/LOOP1

LOOP1 : rapid,goto/7,0.5,0.5

RAPID,GO/PLA1
RAPID,GO/TO,BOTTOM1,PLA1,TO,RIGHT1
FEDRAT/FEEDMR1,IPM
SPINDL/SPMR1
TLRGT,GOFWD/BOTTOM1,TO,LEFT1
LX=LINE/ PARREL,BOTTOM1, YLARGE,(1.5*CUSPDIA1)
TLRGT,GORGT/LEFT1,ON,LX
TLON,GORGT/LX,TO,1,INTOF,RIGHT1

TLLLFT,GOLFT/RIGHT1,TO, TOP1
FEDRAT/FEEDMF1,IPM
SPINDL/SPMF1
TLLLFT,GOLFT/TOP1, TO, LEFT1

RAPID,GO/PL0
XPASS1=XPASS1+1
IF (PASSES1-XPASS1) QUIT1, AGA12, AGA11
QUIT1: RAPID,GOTO/Locstart1
TRACUT/NOMORE

LEFT2=LINE/XCOORD,4.5
TOP2=LINE/YCOORD,4.0
RIGHT2=LINE/XCOORD,6.0
BOTTOM2=LINE/YCOORD,2.5
mx2=matrix/transl,0,0,2
tracut/mx2

DEPTH2=1.25
PASSES2=7
Depth_{inc}2=0.2
XPASS2=1

PL0=PLANE/0,0,1,0.5
CUTDIA2=1
cutter/CUTDIA2
CUTLIM2=0.2
CUSPDIA2=1

Locstart2=point/5.25,3.25,0.5
GOTO/Locstart2

FEEDMR2=21
SPMR2=884
FEEDMF2=36
SPMF2=1500
FEEDRR2=36
SPRR2=1500
FEEDRF2=36
SPRF2=1500
AGA21: PLA2=PLANE/0,0,1,(Depth Inc2*XPASS2)
JUMPTO/LOOP2
AGA22: PLA2=PLANE/0,0,1,-DEPTH2
FEEDMR2=FEEDRR2
SPMR2=SPRR2
FEEDMF2=FEEDRF2
SPMF2=SPRF2
JUMPTO/LOOP2

LOOP2: rapid,goto/7,3,0.5
RAPID,GO/PLA2
RAPID,GO/TO,BOTTOM2,PLA2,TO,RIGHT2
FEDRAT/FEEDMR2,IPM
SPINDL/SPMR2
TLRGT,GOFWD/BOTTOM2,TO,LEFT2
TLRGT,GORGT/LEFT2,TO, TOP2
FEDRAT/FEEDMF2,IPM
SPINDL/SPMF2
TLRGT, GORGT/TOP2, TO,RIGHT2

RAPID,GO/PL0
XPASS2=XPASS2+1
IF (PASSES2-XPASS2) QUIT2, AGA22, AGA21
QUIT2: RAPID,GOTO/Locstart2
TRACUT/NOMORE

LEFT3=LINE/XCOORD,1.5
TOP3=CIRCLE/3.0,5.0,1.5
CONCAVITY_T=-1
RIGHT3=LINE/XCOORD,4.5
BOTTOM3=LINE/YCOORD,2.5
mx3=matrix/transl,0,0,2
tracut/mx3

DEPTH3=0.5
PASSES3=3
Depth Inc3=0.2
XPASS3=1

PL0=PLANE/0,0,1,0.5
CUTDIA3=1
cutter/CUTDIA3
CUTLIM3=0.2
CUSPDIA3=1

Locstart3=point/3,4.5,0.5
GOTO/Locstart3

FEEDMR3=21
SPMR3=884
FEEDMF3=21
SPMF3=884
FEEDRR3=36
Appendix C

APT Program for the Part W

SPRR3=1500
FEEDRF3=36
SPRF3=1500

AGA31 : PLA3=PLANE/0,0,1,(-Depth_inc3*XPASS3)
JUMPTO/LOOP3
AGA32 : PLA3=PLANE/0,0,1,-DEPTH3
FEEDMR3=FEEDRR3
SPMR3=SPRR3
FEEDMF3=FEEDRF3
SPMF3=SPRF3
JUMPTO/LOOP3

LOOP3 : rapid,goto/2,1.5,0.5

RAPID,GO/PLA3
RAPID,GO/TO,LEFT3,PLA3,TO,BOTTOM3
FEDRAT/FEEDMR3,IPM
SPINDL/SPMR3
TLRGT,GO_FWD/LEFT3,TANTO,TOP3
LY=LINE/PARALLEL,LEFT3,X_LARGE,(1.5*CUSPDIA)
TLRGT,GO_FWD/TOP3,ON,LY
TLON,GOR_GT/LY,TO,1,INTOF,BOTTOM3

TLLFT,GOL_FT/BOTTOM3,TO,RIGHT3
FEDRAT/FEEDMF3,IPM
SPINDL/SPMF3
TLLFT,GOL_FT/RIGHT3,TANTO,TOP3

RAPID,GO/PL0
XPASS3=XPASS3+1
IF (PASSES3-XPASS3) QUIT3, AGA32, AGA31
QUIT3: RAPID,GO_TO/Locstart3
TRACUT/NOMORE

LEFT4=LINE/XCOORD,0.0
TOP4=LINE/YCOORD,2.5
RIGHT4=LINE/XCOORD,1.5
BOTTOM4=LINE/YCOORD,0.0
mx4=matrix/transl,0,0,2
tracut/mx4

DEPTH4=0.75
PASSES4=4
Depth Inc4=0.2
XPASS4=1

PL0=PLANE/0.0,1.0,0.5
CUTDIA4=1

CLUSP Dia4=1
CUTLIM4=0.2
CUTLIM4=0.2
CUSPDIA4=1
Locstart4=point/0.75,1.25,0.5
Appendix C

APT Program for the Part W

GOTO/Locstart4

FEEDMR4=21
SPMR4=884
FEEDMF4=36
SPMF4=1500
FEEDRR4=28
SPRR4=1179
FEEDRF4=36
SPRF4=1500

AGA41 : PI_A4=PI_ANE/0,0,1,(-Depth_Inc4*XPASS4)
JUMPTO/LOOP4
AGA42 : PLA4=PLANE/0,0,1,-DEPTH4
FEEDMR4=FEEDRR4
SPMR4=SPRR4
FEEDMF4=FEEDRF4
SPMF4=SPRF4
JUMPTO/LOOP4

LOOP4 : rapid,goto/0.5,-1,0.5

RAPID,GO/PLA4
RAPID,GO/TO,LEFT4,PLA4,TO,BOTTOM4
FEDRAT/FEEDMR4,IPM
SPINDL/SPMR4
TLRGT,GOFWD/LEFT4,TO,TOP4
TLRGT,GORGT/TOP4,TO,RIGHT4
FEDRAT/FEEDMF4,IPM
SPINDL/SPMF4
TLRGT,GORGT/RIGHT4,TO,BOTTOM4

RAPID,GO/PL0
XPASS4=XPASS4+1
IF (PASSES4-XPASS4) QUIT4, AGA42, AGA41
QUIT4: RAPID,GOTO/Locstart4
TRACUT/NOMORE

SPINDL/OFF

END
FINI