INTERNATIONAL 88 PROCEEDINGS. Ed. BY G. CHRYSSOLOURIS, ET. AL., APRIL 17-20, 1988, ATLANTA GA., Pgs. 209-217.

PROTOCOL FOR FLEXIBLE MANUFACTURING AUTOMATION WITH HEURISTICS AND INTELLIGENCE

D. K. Anand and **J. A. Kirk**, Professors of Mechanical Engineering
M. Anjanappa, Assistant Professor of Mechanical Engineering
D. Nau, Associate Professor of Computer Science
E. Magrab, Professor of Mechanical Engineering

The Systems Research Center
University of Maryland
College Park, Maryland

INTRODUCTION

The approach to successful computer automation [1-7] of the design procedure and subsequent manufacture initially depends on whether the product under consideration is a component or system. The latter can introduce a level of complexity that is several orders of magnitude larger than that of a component. Even for a component it is appropriate to initially speak about a class of components aggregated by the group technology approach. In this paper, we are interested in limiting our consideration to components that can be manufactured on a CNC machining center and adequately represented by 2½D geometry. Such manufacturing of parts in small batches represents approximately 75% of the manufacturing effort in the U.S. Specifically, we are concerned with flexible manufacturing in a cell where we incorporate expert or knowledge-based systems to the fullest extent possible. Also, it is important that we incorporate heuristic reasoning in addition to the algorithmic approach throughout the process from design to manufacture [1-19].

The manufacture of goods requires seven steps, viz. specification, product design including process planning, scheduling, raw material aquisition, production, quality assurance and monitoring, and shipping/distribution. The automation studied here is concerned with steps one, two, five and six. The proposed approach utilizes AI techniques and interface standardization and, as such, it is both intelligent and integrated. The former implies expertness, while the latter implies a standard communication format such as the Manufacturing Automation Protocol (MAP). The tools used for design, optimization and manufacture of the part are typically CAD/CAM, AI, GT, etc.

Although there is no general computation theory for design, a procedural methodology aided by algorithms and reasoning does exist. CAD/CAM is simply a design methodology aided by a relational and heuristic data base. To optimize the design for manufacture (DFM), it is important that the design be compatible with new technology available in manufacturing. There are two optimal strategies. The first is the absolute optimum plan which is generated by ignoring all constraints. The second, and more useful optimum strategy, is based upon equipment capability and fixturing/material handling constraints.

Prior to manufacture a process plan must be generated which includes the establishment of manufacturing constraints on manufacturability. The process plan includes selection of an appropriate machine tool based upon dimensions, tolerances and surface finish requirements of the design. The ideal plan would have considered the interface structure, tooling, control and the performance of each manufacturing operation.

The process plan chooses an optimal set of recommendations for a specific machine tool. These recommendations may include type of work holding device, its location, cutting tool, sequence of cutting operations and cutting parameters for each machining operation. To further aid process planning group technology will allow the grouping of parts in accordance with their processing requirements.

Although not considered here, the issue of timing and scheduling as well as data base handling including data collection, updating and use is a very important element of factory automation and crosses all boundaries. A schematic of our approach is illustrated in Fig. 1. The figure has the following four important and, often, overlapping phases: Design, Process Planning, Machining, Interfacing. To achieve flexible automation of components, each of the phases raise specific issues. A few of these are listed below.

Design: The particular part must satisfy design axioms of independent functional requirements and minimum information content. Important issues being addressed include: reasoning by analogy, innovation via heuristics, algorithmic computation, quality versus cost, range of design and scale, manufacturability, and the maintenance of an iterative design environment, and special attention to Feature Based Design to overcome IGES limitations.

Process Planning: One of the most important issues is that of geometrical representation and feature extraction from the finalized design. Other issues for purposes of planning the machining process are: qualitative versus heuristic rules, spatial and temporal reasoning, interference avoidance, intelligent fixturing, maintanence of accuracy, hard physics versus soft expert advice and cell dynamics.

Interfacing: This phase primarily deals with the need for a standard approach to communication; that is, the ability to translate within the design-to-manufacture path. Important issues include: IGES data representation, M/G Codes for machining, and appropriate languages for different functions (AUTOLISP or PROLOG for planning, BASIC or C for design, etc).

Machining: Although the machine performance is given, it is necessary to define those attributes that will impact the other three functions described above. This often includes machining accuracy, available speeds/feeds, and the internal language of the machine controller and more importantly to interface with it.

This paper is concerned with the automation from, and including, design to manufacture consistent with the concept of the paperless factory of the future. The particular approach is referred to as the Flexible Manufacturing Protocol (FMP) with specific emphasis on intelligence and heuristics. The protocol discussed applies to the manufacture of prismatic parts in general, with particular attention to thin-ribbed microwave guides. With minor modification this approach is also being developed and tested on a turning center. The implementation of the manufacturing cycle is shown in Fig. 2, which is the manufacturing protocol developed at the University of Maryland. The overall control is hierarchical with protocol emphasis placed on the inclusion of heuristics and intelligence in the process of coupling the user with production of the component.

DESIGN

One of the design goals in University of Maryland's Flexible Automation Facility is to create an environment whereby the design engineer can become simultaneously involved in the manufacturing procedures (in our case, machining) of the component being developed. Although it would appear to be somewhat unrealistic to expect one person to have sufficient knowledge and experience to successfully design a component, predict its performance and then generate the procedures of manufacture, it is felt that with access to sufficiently expert programs this goal can often be met.

The initial design process usually consists of the development (or delegation) of a set of functional specifications. These specifications often comprise a combination of: materials, loads and deflections, constraints on connectivity, volume and weight limitations, tolerances and surface finishes, environmental considerations such as temperature and corrosion and the intended use, which implies consideration of such things as safety and reliability. In addition, budgetary and time constraints for the entire process may be stated, as may be the final cost (selling price) of the product. The design engineer can satisfy, usually in an interactive manner, many of the above stated specifications with the assistance of commerically available finite element and optimization programs. What is missing is the designer's ability to also control, modify and otherwise influence the manufacturing process by which the component is to be produced.

The University of Maryland's approach to the above problem is to give the designer access to the manufacturing (machining) process with the assistance of a series of programs that provide for both graphical displays and "expert advice." The first step has been to expand and improve upon feature-based design (FBD) system originally used at NBS [8-10]. The FBD system uses as its primative a host of machinable features, in our case those features obtainable from a vertical machining center. This approach forces the designer to simultaneously consider the design and manufacturing process. However, he does not have to concern himself with the details of the manufacturing process until his design is completed, with the following exception: he must be specific about the size (diameter and length) and shape of the tool to be used to obtain each feature. Upon completion of the design he will have at his disposal several aids, which are frequently used in an interactive fashion. One such aid rearranges the order in which the features are generated according to a selected criterion, e.g., precision machining, rapid material removal, tool life, etc. Another aid will display the fixturing devices and their location. This aid will also perform tool/fixture/ part interference checking for each feature in the order they are produced. The third aid is a program that keeps track of all aspects of tooling. It will contain a library of existing tools that can be searched by attribute; e.g., diameter, length, type, etc. It will also contain a record of each tools estimated remaining cutting life, its location (tool crib, machine tool, etc.) and data to indicate what its solid model representation is for use in interference checking. The last aid will also be a program that generates the numerical control (NC) code to cut each feature. Based on the tool characteristics, the part's material and the criterion selected for the feature ordering, the program will determine the appropriate feeds, speeds and depths of cut.

For the operation of the protocol, several packages (CAD, Process Planning, Manufacturability, IGES Standardization, etc.) have been developed and integrated into the overall shell. The shell itself is intended to be very user friendly allowing for multiple selection criterion from design to manufacture. In the remainder of the paper we discuss several of the more interesting packages and then present an example of the working of the protocol.

HEURISTICS

The inclusion of heuristics in automation is a difficult task at best. This is because unlike algorithmic approaches it is an open-ended methodology. Basically, heuristics implies the codification of data, rules, methods, etc., that generally falls under "rules of thumb" or "experience". Ideally, if we could question (exhaustively) a large number of experts and systematize their responses to form a program, we would theoretically have a 'heuristic' package.

At the University of Maryland we are introducing heuristics into the protocol using two approaches. The first approach involves the introduction of rules in the intelligent process planner and is discussed in a separate section. The second approach involves the preparation of a software module called "manufacturability". In this module we address issues such as tolerances, general rules for compatibility and issues pertaining to practices and procedures in the shop floor. This particular module is under current development and only simple tolerancing can be addressed at the moment. More importantly, the manufacturability module and the process planner are open-ended, thereby allowing the addition of more rules based upon new data and experience.

INTELLIGENT PROCESS PLANNING

AI techniques can be used to automate (at least partially) several of the reasoning activities involved with process planning. One example of this is SIPS (Semi-Intelligent Process Selector), which uses AI techniques for generative selection of machining operations for the creation of metal parts [12-14]. SIPS is also being integrated into the process planning system in the Automated Manufacturing Research Facility project at the National Bureau of Standards.

SIPS (written in LISP) considers a part to be a collection of machinable features--and for each feature, it generates a sequence of machining steps to use in creating that feature. It does this by reasoning about the intrinsic capabilities of each manufacturing operation. The process selection done by SIPS includes both the high-level process selection done by the process engineer (e.g., "mill this face") and the lower-level process selection done by the NC programmer (e.g., "rough-end-mill this face").

SIPS uses a new approach to knowledge representation called hierarchical knowledge clustering, in which the problem-solving knowledge is organized in a taxonomic hierarchy. Each node in the hierarchy is represented by a frame. SIPS contains three such hierarchies: one for machinable features, one for machining processes, and one for cutting tools. For example, Figure 3 shows the names of some of the frames in SIPS's hierarchy of machining processes, as printed out by SIP's graphic interface. Stored in each frame is detailed information about the machining process it represents, including information on what kinds of machinable features the process can create, what the best machining tolerances are that it can satisfy, what kinds of machinable features must already be present in order to perform this process, what the process costs, etc.

In SIPS, a component part is represented as a collection of machinable features, each of which is an instantiation of one of the frames in the feature hierarchy. SIPS creates plans for each of these features independently. One significant problem with this approach is the feature interaction problem. For example, it may be easy to create a hole in a flat surface if the surface is unobstructed, but the same task may be impossible if some other part of the workpiece interferes with the tool trajectory. To handle tasks such as this, an interface has been created between SIPS and the PADL solid modeler [15]. Before attempting to create a plan for a feature, SIPS queries PADL to see if various global geometric constraints are satisfied. If these constraints are satisfied, SIPS proceeds to create a plan for the feature.

SIPS considers each feature as a goal to be achieved, and decides how to achieve this goal by searching backwards to develop a sequence of machining operations capable of achieving the goal. The search is done by using a best-first Branch and Bound procedure. For each feature, the first plan SIPS finds for creating that feature is guaranteed to be the least costly one.

In order to do tool selection, SIPS considers this as a sequence of three specific tasks: determining what type or family cutting tools can successfully manufacture the part, determining a proper tool size to fit the constraints of the feature, and determining the tool material that can best produce the required finish.

Given a specific machining process, the scope of tool selection is limited to identifying a specific family of tools that can be used for the process, and identifying the best tool material to use for the process. This approach does not satisfy all aspects of tool selection--but it does provide an initial approach to the problem, and this will lead to more effective use and selection of cutting tool materials with future work.

The knowledge representation scheme in SIPS provides a natural way to represent, organize, and manipulate knowledge about machinable features, machining processes, and cutting tools. In a general sense, the search technique SIPS uses for problem solving is similar to the reasoning process that a manufacturing engineering goes through every time a part is considered for planning.

SIPS currently runs on Symbolics lisp machines, Texas Instruments Explorers in Zeta Lisp, and on Sun workstations and Silicon Graphics Iris workstations in Franz Lisp. At present SIPS knowledge base consists of more than 90 frames describing machinable features, machining processes, and cutting tools. The knowledge base is set up primarily to work on prismater parts having machinable features such as holes, pockets, and slots; but it is being extended to deal with lathe-turned parts as well.

Most approaches to the integration of solid modeling with automated process planning have essentially involved using the modeler as a front end to the process system, by taking machinable features from the modeler and sending them to the planning system, which then reasons about these features without further interaction with the solid modeler.

In order to generate correct process plans for complex objects, this approach is not sufficient. For producing correct process plans for complex objects, it is necessary for the process planning system to interact extensively with the solid modeler while the process planning is going on. For this reason, additional work is being done on making it efficient to answer queries and make incremental changes between the modeler and process planner. To satisfy this, a new feature extraction scheme is also under current development.

ACCURACY ENHANCEMENTS

To achieve the total benefits of automation it is necessary to insure that the part tolerances and surface finish are maintained within acceptable limits [16-18]. This can be achieved by controlling the tool path either by precalibrated compensation or active on-line correction schemes.

Tool path error can be defined as the distance difference between the programmed path and the actual path of the tool relative to the workpiece. Tool path errors are classified based on the source and characterized as deterministic (both static & dynamic) and stochastic [19] as shown in Fig. 4. Also, the tool path error can be classified into two major groups based on their sources, viz. 1) cutting force independent errors, and 2) cutting force induced errors. The cutting force independent tool path error includes the errors due to weight deformation, thermal deformation and positional inaccuracies. These errors are static and dynamic deterministic in nature.

In previous work, although static errors have been compensated, dynamic deterministic errors are not included although these could be significant for high feedrates such as that used in high speed machining. The work being conducted here at the University of Maryland represents a comprehensive tool path error correction scheme for a vertical machining center.

The strategy of tool path error correction, for cutting force-independent deterministic errors, is based on nullifying the effect of the error itself by algebraic summation of the programmed position input with the output of the process model. Two methodologies of implementing this strategy are shown in Fig. 5. In path-A, the M&G numerical control machine tool motion codes are modified before being fed into the machine controller. In path-B, the error correction is superimposed over the nominal table movements by taking advantage of magnetic bearing spindle technology [16-17].

The error correction via modified M & G codeny (path-A) is complete and the results will be reported in a detailed paper shortly. An expert system is being developed as part of the methodology in following path-A. Included in this expert system are provisions to produce the required information which is needed to implement the methodology for path-B.

The deformations due to cutting forces are both static deterministic and stochastic in nature [19]. The cutting force erro00 can therefore be considered as (a) deterministic tool path errors due to varying compliance between the tool and workpiece, (b) stochastic tool path errors due to variations in depth of cut and process dynamics.

To minimize tool path errors, both the errors due to the presence of cutting forces are to be minimized. This is done by first considering only the errors in positioning of the cutting tool relative to an infinitely stiff workpiece. These errors, which are deterministic, are minimized in the following manner:

- Calculate the steady state cutting forces using conventional metal cutting theory and directly compensating
- Experimentally generating deflection "look-up-tables" for both the tool and table deflections.

To control the remaining tool path error, which is stochastic, the cutting process is modeled as a discrete stochastic process by including the uncertainties as uncontrollable imputs. The uncontrollable inputs are represented as a white noise of limited band width [19].

THE FLEXIBLE MANUFACTURING PROTOCOL (FMP)

The protocol illustrated in Fig. 2 involves the user providing design input via a computer aided design (CAD) environment. This is followed by a feature extractor which expresses the part representation into a set of form features. Following this decomposition, an expert system is used to establish the manufacturability of the design with feedback to the user for additional input. Next, an intelligent process planner is available for determination of optimal component machining. The success of this planner may be compared directly to a parallel path consisting of an ordered process planner. Following either path results in a CNC machining plan which is placed in the CAM database. Through repeated automatic queries to the global database, the CAM database is interwoven with cell command information prior to downloading to a flexible manufacturing cell for fully automatic production of the component.

The flexible manufacturing protocol is in phase II of implementation at the University of Maryland. A CNC vertical machining center and a simulated material handling robot and AGV are interfaced through a HP 9836 cell controller which is coupled to a SUN workstation. The protocol can also be accessed by an IBM/AT. The entire protocol is capable of operating with commercial CAD software systems in which the CAD data is available in standard IGES format.

One of the important considerations in this work has been the use of standardization at different stages of the protocol. The standardization of the design data base in IGES format and the use of M & G machine codes allows the protocol to be easily interfaced with a variety of commercial CAD packages [on the design side] and different CNC machines [on the manufacturing side].

AUTOMATED MACHINING VIA FMP

The current version of the FMP, shown in Fig. 2, has been implemented for controlling a Matsuura MC-510 vertical machining center. The FMP as implemented is capable of using either a commercial based CAD system that is IGES compatible or a feature based design system. In the following section an example using the commercial CAD path of the protocol is discussed.

Typically, the protocol operates when a part is created on a CAD system using entities such as lines, arcs, points, fillets and champfers. For the present work the protocol is limited to prismatic parts having one or more milling based features consisting of linear slots, circular slots, rectangular pockets, circular pockets, and drill holes. The user starts the design by first sketching a rectangular block for the raw workpiece. He then designs the part by a substraction process in which features are positioned at various location on the blank. The FMP recognizes that a feature indicates a material removal operation within the boundary of the feature. By suitable placement of these features the user defines the complete material removal operation needed to produce the finished part.

In the present implementation milling features are stored internally in the database structure which is particular to the CAD system being used. This internal representation is then converted to a standard IGES database consisting of a wireframe model of the part.

After the part is expressed in standard IGES format the IGES file is stripped of extraneous data and is reduced to a form which has the minimum data necessary to define the part. This file represents a compact listing of all the machinable features which are required to produce the part.

The next step performed by the FMP is to identify the machinable features which are present in the stripped IGES file. This task is identified by the "Feature Extractor" block in the FMP flowchart. The techniques of feature extraction are under continuous refinement and in the present implementation an algorithmic approach is used. For example, a rectangular pocket is identified by the presence of a flag entity (the "point" entity is currently used) followed by a line entity in the IGES data base. The next twenty three lines of the stripped IGES file are then read in as a description of the pocket data. The data of the rectangular pocket consists of one flag entity, eight arcs, and sixteen line segments. Figure 6 shows a typical IGES data for a rectangular pocket with the first four lines representing the four straight lines in the top view of the pocket From these lines the coordinate data of the boundary of the pocket is extracted. The next four lines represents the four fillets on the top of the pocket. From these four arcs information on the radius of the fillet is obtained. The ninth line provides the information on the depth of the pocket. The remaining lines are not used. A similar procedure is then repeated for every feature present in the stripped IGES database, thus creating a new feature database.

The feature database is the input to the process planning module. In the present FMP an ordered process planner arranges the feature database in the same order in which the designer has sketched his part. While this ordering is taking place the user is presented with each feature and is requested to enter the machining parameters for the feature. Before initiating feature reordering and machining input, the user is presented with a tool library containing information on the type, diameter and length of tools which are available in the machine tool inventory. The user can edit, add or delete from this library and eventually create the tool library particular to the machine tool which is being used. Once the tool library has been defined the first feature is presented to the user, along with its dimensions, and the user is prompted for machine tool parameters. For the rectangular pocket the dimensions of the pocket - length, width, depth and the coordinate of one location point are displayed on the top half of the computer screen. The bottom half of the screen has prompts for the machine parameters which include - tool library number for the rough cut tool, tool library number for the finish cut tool, feed rate and type of cut (finish/rough).

Once machine data input for all the features is supplied a CAM data base is generated. This database contains the geometric data of the part combined with the machining data. The CAM database contains all the information that is needed to machine the part and an example of the database is shown in Figure 7.

The next step in the implemented FMP is to process the CAM database into standard cutting codes capable of driving the machining center. These standard codes are called M and G codes and are specified in EIA RS-274. The G codes are known as preparatory functions and they define machining moves and cutter tool movement such as point to point positioning and linear interpolation. The M codes are known as miscellaneous function control codes and include program stop, spindle on/off, clamp/unclamp, tool change, etc.

In the present work only four basic G codes [i.e. G00, G01, G02, G03] have been used for defining all cutting operations. Specific routines have been written to generate all features in terms of these 4 basic G codes. These G codes are standard and found in all CNC machines accepting M & G codes, thus insuring that the M & G codes generated by the FMP are machine independent. However, besides these four G codes other G codes have also been used. These G codes are for defining the different coordinate systems, tool length compensation, cutter radius compensation, plane selection, etc. A list of the M & G codes used in the FMP can be found in reference [3].

The generation of M & G codes for any part is divided into three sections. The Start section contains the machine initialization codes. This section defines the machine coordinate system, the workpiece coordinate system, dimension system (inch/metric) and sets the absolute mode. The middle section contains the actual cutting moves required to machine the part, and may include several automatic tool changes. The End section stops the spindle and coolant, takes the tool to the zero position in the machine coordinate system and stops the machine.

At present, after the M & G code database has been generated the prismatic raw stock is manually fixtured on the machine tool. Fixtures are placed on the raw stock by observing a picture of the final part and choosing clamping locations in which no fixture intercepts occur during machining [3]. The M & G codes are then downloaded to the machine after the cutting tool magazine is loaded to be in agreement with the tool library. The tool length offsets and cutter compensations are entered for each tool and then cutting is initiated. At the end of all cutting the finished part is currently manually unloaded. A typical sample part is shown in Figure 8. This part consists of 2 pockets and a number of holes with the prismatic boundary of the blank clearly identified.

CONCLUSIONS

A Flexible Manufacturing Protocol [FMP] is under current development for automated component manufacturing at the University of Maryland. The protocol is concerned with creating an environment whereby the design engineer becomes involved in the manufacturing process of the component being developed. The input format to the FMP is either an IGES representation of the designer's part or a feature based representation. Either type of input is analyzed with algorithmic, heuristic, or artificial intelligence based software, and control codes required to produce the part are automatically generated. An initial version has been develojped and validated on a vertical CNC machining center.

The flexible manufacturing cell has been established to validate the element of the FMP. This cell consists of a CNC vertical machining center, a simulated material handling robot and automated guided vehicle. A version of the FMP has been implemented in the flexible manufacturing cell and a number of parts, represented in IGES data format, have been produced in the cell.

Although the FMP has been developed and is being demonstrated in a machining application, the overall concept and module integration is equally applicable to other areas such as flexible automation for Printed Circuit Board assembly.

ACKNOWLEDGEMENTS

This work is supported in part by the Engineering Research Center, Systems Research Center, National Science Foundation and The National Bureau of Standards.

REFERENCES

1. Anand, D.K., "Research in Flexible Manufacturing," Second Annual SRC Conference, September 1986, College Park.

2. Anand, D.K., Kirk, J.A., Anjanappa, M., and Pecht, M.G., "Supercomputers and Hierarchical Control: A

Systems Viewpoint," Lawrence Livermore Laboratory - NSF Proceedings, September 1984, Livermore, CA.

3. Uppal, R. "Flexible Manufacturing Protocol Driver For a Vertical Machining Center", M.S. Thesis, University of Maryland, Aug. 1987.

4. Gershwin, S.B. "A Central Perspective on Recent Trends in Manufacturing Systems", Conference on Decision and Control, December 1984, Las Vegas, Nevada.

5. Baer, T., "Simulating The Factory," Mechanical Engineering, December 1986.

6. Anand, D.K., Kirk, J.A., and Anjanappa, M., "Tool Path Error Control of End Milling of Microwave Guides", Proceedings of 7th World Congress on the Theory of Machines and Mechanisms, September 1987, Universidad de Sevilla, Sevilla, Spain.

7. Francis, P.H., "Toward a Science of Manufacturing", Mechanical Engineering, Vol. 108, No. 5, 1986, pp. 32-37.

8. Kramer, T.R. and Jun, J., "Software for an Automated Machining Workstation", Proceedings of the 3rd Biennial International Machine Tool Technical Conference, Sept. 1986.

9. Magrab, E.B., "The Vertical Machining Workstation of the AMRF: Software Integration", Integrated and Intelligent Manufacturing, C.R. Lin and T.C. Chang, Eds., ASME DED - Vol. 21 (Dec. 1986), pp. 83-100.

10. McLean, C.R., "The Vertical Machining Workstation of the AMRF: Software Integration", Integrated and Intelligent Manufacturing, C.R. Lin and T.C. Chang, Eds., ASME DED - Vol. 21 (Dec. 1986), pp. 101-116.

11. P. Brown and S. Ray, "Research Issues in Process Planning at the National Bureau of Standards," Proc. 19th CIRP International Seminar on Manufacturing Systems, June 1987, pp. 111-119.

12. N.C. Ide, "Integration of Process Planning and Solid Modeling through Design by Features," Master's thesis, Computer Science Dept., University of Maryland, 1987.

13. D.S. Nau, "Hierarchical Abstraction for Process Planning," Proc. Second Internati. Conf. Applications of Artificial Intelligence in Engineering, June 1987.

14. D.S. Nau and M. Gray, "SIPS: An Application of Hierarchical Knowledge Clustering to Process Planning," Proc. Symposium on Integrated and Intelligent Manufacturing at ASME Winter Annual Meeting, Anaheim, CA, Dec. 1986, pp. 219-225.

15. H.B. Voelcker, A.A.G. Requicha, E.E. Hartquist, W.B. Fisher, J. Metzger, R.B. Tilove, N.K. Birrell, W.A. Hunt, G.T. Armstrong, T.F. Check, R. Moote, and J. McSweeney, "The PADL1.0/2 System for Defining and Displaying Solid Objects," ACM Computer Graphics 12:3, Aug. 1978, 257-263.

16. Kirk, J.A., Anand, D.K., Anjanappa, M., "Magnetic Bearing Spindle Control in Machining," Proceedings of 13th Conference on Production Research and Technology, November 1986, pp. 127-130.

17. Anand, D.K., Kirk, J.A., Anjanappa, M., "Magnetic Bearing Spindles for Enhancing Tool Path Accuracy," Advanced Manufacturing Processes, Vol. 1, No. 1, pp. 245-268, 1986.

18. Anjanappa, M., Kirk, J.A., Anand, D.K., "Tool Path Error Control in Thin Rib Machining," Proceedings of 15th NAMRC, pp. 485-492, May 1987.

19. Anjanappa, M., "Error Minimization in Machining", Ph.D. Dissertation, Univ. of Maryland, 1986.

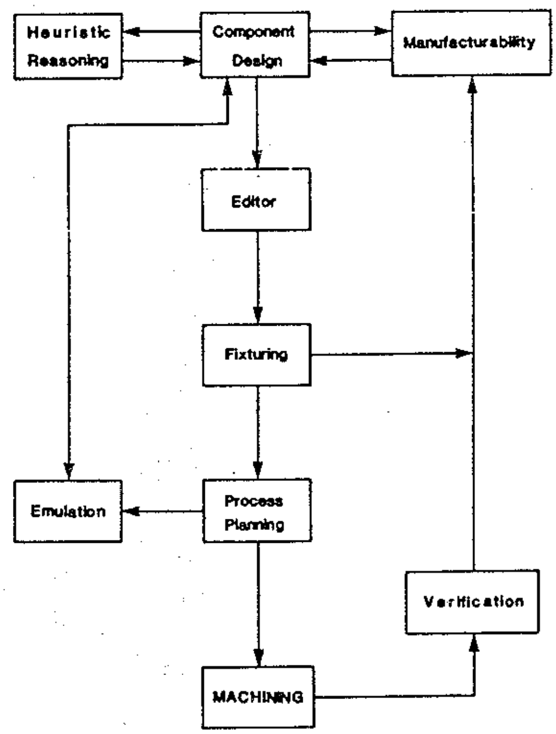

Figure 1 Approach for Automated Machining

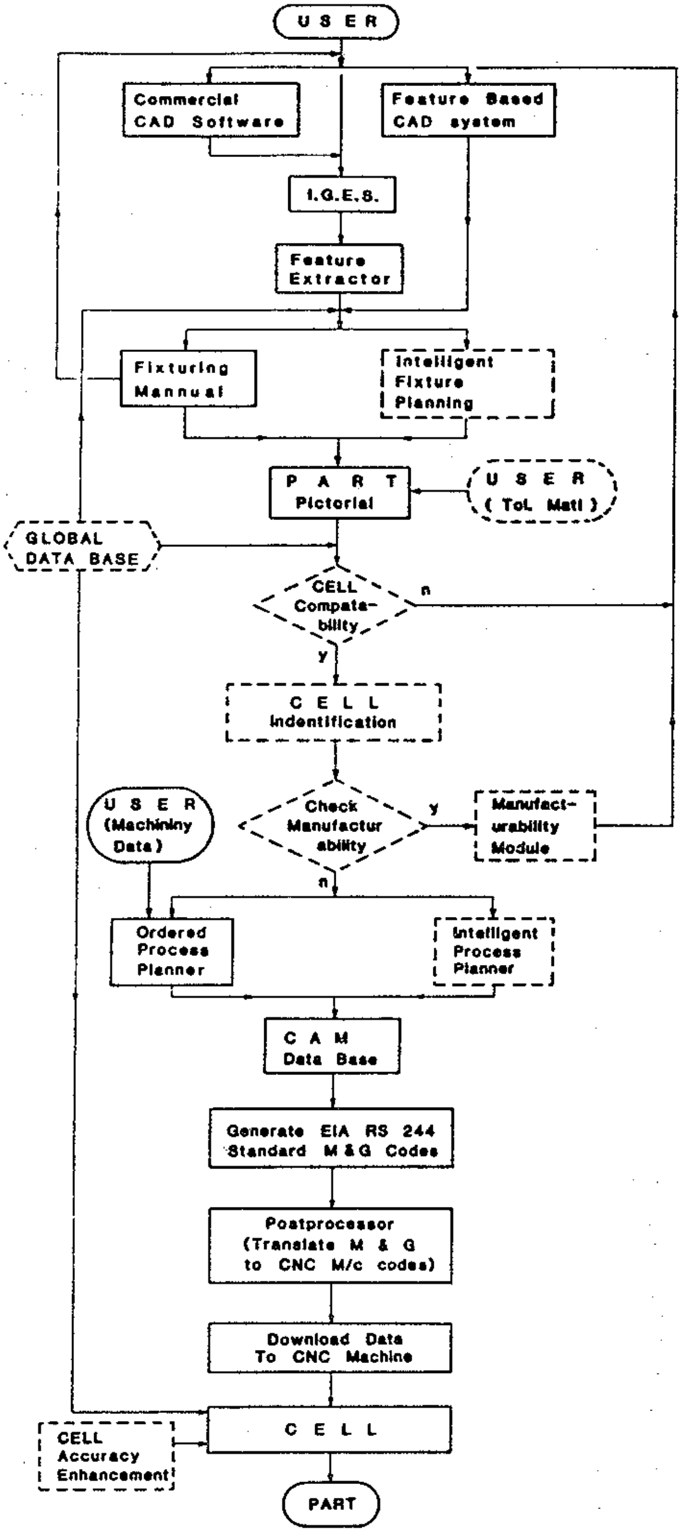

Figure 2 Implemented Flexible Manufacturing Protocol

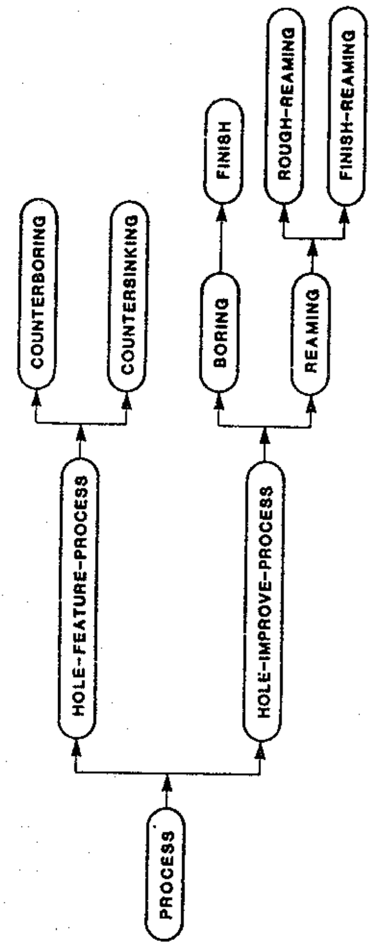

Figure 3 An example of the Frames included in SIPS hierarchy of machining process

	CUTTING FORCE-INDEPENDENT ERRORS			CUTTING FORCE-INDUCED ERRORS	
TYPE OF ERROR	POSITION ERROR	THERMAL DEFORMATION	WEIGHT DEFORMATION	CUTTING FORCE DEFORMATION	
NATURE OF ERROR	DETERMINISTIC			DETERMINISTIC	STOCHASTIC
FUNCTIONAL DEPENDENCY	POSITION, FEEDRATE (P,f)	TEMPERATURE (T)	MASS (M)	COMPLIANCE (k)	DEPTH OF CUT , PROCESS DYNAMICS

Figure 4 Tool Path Error Classification

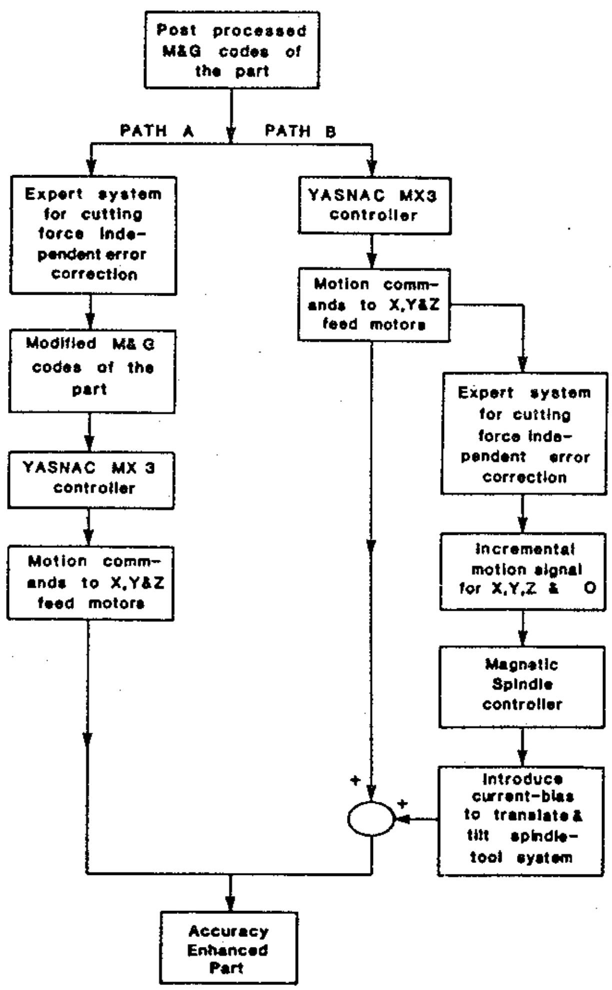

Figure 5 Error Correction Methodologies

```
110,4.7,2.,0.,2.3,2.,0.                    ; #1 Line
100,0.,2.3,2.3,2.,2.3,2.3,2.               ; #5 Arc
110,4.7,2.,-1.2,2.3,2.,-1.2                ; #9 Line
100,-1.2,2.3,2.3,2.,2.3,2.3,2.             ; #24 Arc
```

Figure 6 IGES data for a rectangular pockek

```
BLANK
 6 , 5 ,-2

RPF
 1.9375 , 1.9375 , 5.9375 , 5.0625 ,-.3 , .5 , 4 , 3 , 6.5 ,Y

RPF
 2.5 , 2.5 , 5.4375 , 4.5 ,-.5 , .3 , 3 , 2 , 5.5 ,N

CPK
 3.125 , 3.875 , .3499999 ,-.C , 3 , 5.5 ,CL,Y

DRILL
 3.125 , 3 , .3499999 ,-.8 , 2 , 4.5 , 2
```

Figure 7 CAM database

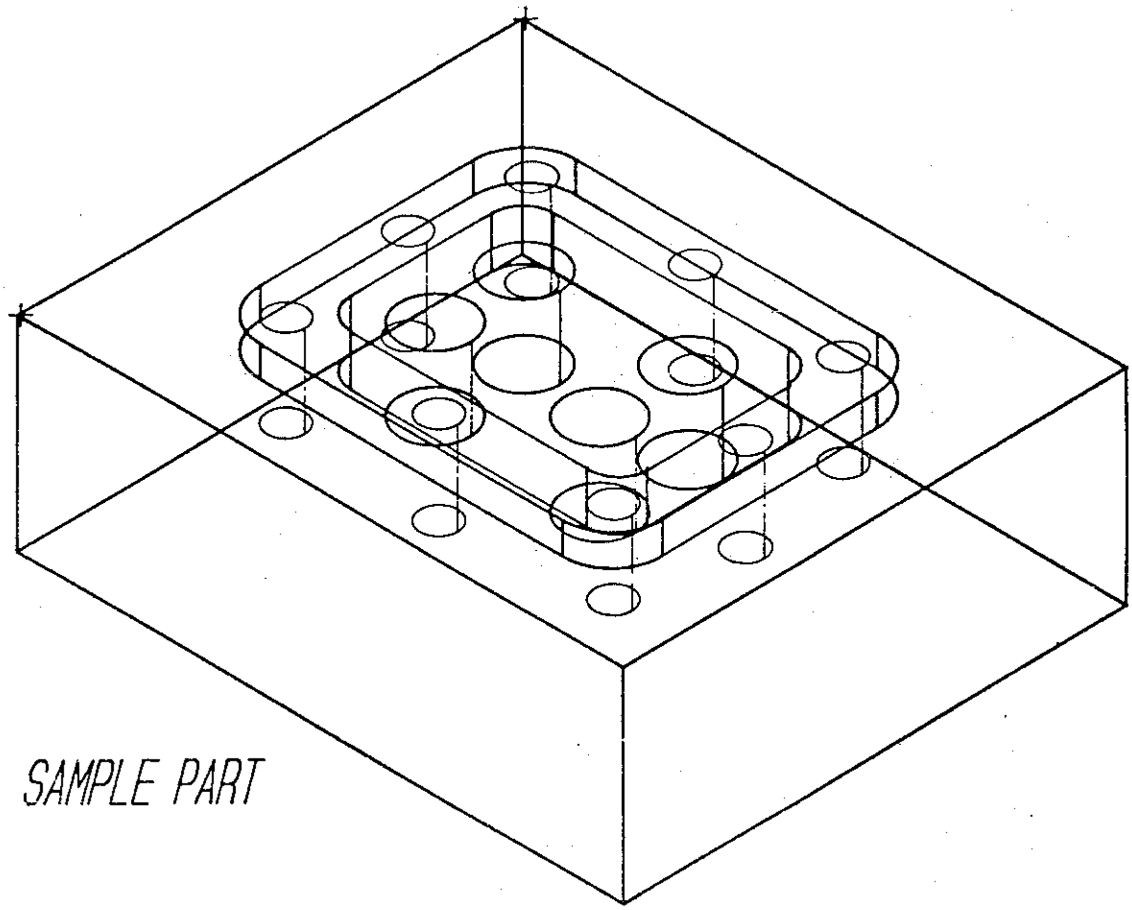

Figure 8 Sample part machined on a Matsuura 510 V machining center