

Saleh M. Amaitik · S. Engin Kiliç

## An intelligent process planning system for prismatic parts using STEP features

Received: 15 September 2005 / Accepted: 21 November 2005 / Published online: 15 February 2006  
© Springer-Verlag London Limited 2006

**Abstract** This paper presents an intelligent process planning system using STEP features (ST-FeatCAPP) for prismatic parts. The system maps a STEP AP224 XML data file, without using a complex feature recognition process, and produces the corresponding machining operations to generate the process plan and corresponding STEP-NC in XML format. It carries out several stages of process planning such as operations selection, tool selection, machining parameters determination, machine tools selection and setup planning. A hybrid approach of most recent techniques (neural networks, fuzzy logic and rule-based) of artificial intelligence is used as the inference engine of the developed system. An object-oriented approach is used in the definition and implementation of the system. An example part is tested and the corresponding process plan is presented to demonstrate and verify the proposed CAPP system. The paper thus suggests a new feature-based intelligent CAPP system for avoiding complex feature recognition and knowledge acquisition problems.

**Keywords** Process planning · CAPP · STEP · Neural networks · Fuzzy logic

### 1 Introduction

Process planning has become a promising candidate for enhancing the adaptability and flexibility of manufacturing

systems. It deals with the selection of manufacturing processes and parameters to transform a part from its initial form to a final shape according to design specifications [1]. It is a function which systematically determines how a product is to be manufactured, so that the end product will be functional, economical, and of acceptable quality. There is much literature in the area of computer aided process planning (CAPP). A list of more than 100 CAPP systems is provided by Zhang and Alting [1]. These CAPP systems are divided into: (1) variant CAPP systems (based on Group Technology coding), (2) generative CAPP systems (based on built-in decision making logic), (3) hybrid CAPP systems (which combine variant and generative approaches), and (4) knowledge-based CAPP systems (based on expert systems). Recently, many new generation CAPP systems have been developed. These systems have the advantages of implementing advanced artificial intelligence techniques and integrated product data models. In terms of the application of artificial intelligence techniques in the development of CAPP, expert system and knowledge base approaches have been used to build CAPP systems since the 1980s. However, the results are not so promising due to the knowledge acquisition bottleneck [2]. New advanced artificial intelligent techniques such as neural networks and fuzzy logic provide promising approaches for automated knowledge acquisition and can be advantageously used in the building of new generation CAPP systems. This is due to their ability to acquire new knowledge and to cope with the dynamic changes of manufacturing systems. The literature documents several research efforts in the field of application of neural networks and fuzzy logic in CAPP systems. Santochi and Dini [3] applied neural networks in selection of cutting tools. A neural network based CAPP approach for selection of machining operation of rotational components has been reported by Devireddy and Ghosh [4]. Mei et al. [5] presented a neural network approach for CAPP systems to automatically select manufacturing datum for rotational parts. Ming and Mak [6] adopted Kohonen self-organizing neural networks and Hopfield networks to solve the setup planning problem in CAPP. El Baradie [7] developed a

S. M. Amaitik (✉)  
Department of Manufacturing Engineering, Atilim University,  
G-Block, Room A-14,  
Incek, Golbasi, 06836 Ankara, Turkey  
e-mail: amaitik@atilim.edu.tr  
Tel.: +90-312-5868352

S. E. Kiliç  
Mechanical Engineering Department,  
Middle East Technical University,  
06531 Ankara, Turkey  
e-mail: engink@metu.edu.tr

fuzzy logic model for selection of machining parameters. Zhao [8] presented a methodology that uses fuzzy sets and fuzzy decisions to create optimal process plans for metal cutting manufacture.

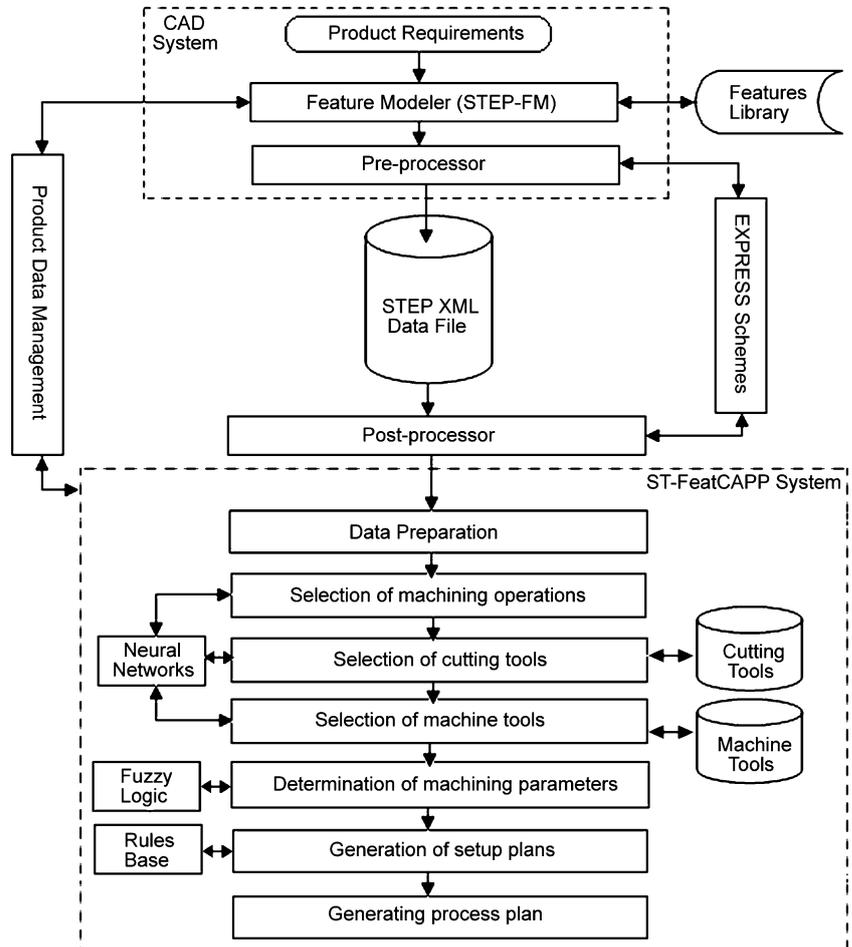
In terms of integrating CAPP systems, the feature-based approaches have been recognized as essential tools for eventually integrating process planning and design. Feature-based approaches are divided into two groups, namely, feature recognition and design by feature. The feature recognition approach examines the topology and geometry of a part and matches them with the appropriate definition of predefined features. The design by feature approach builds a part from predefined features where their attributes are attached. Several research efforts over the last two decades have been focused on feature-based approaches to integrate design and automated process planning. Devireddy and Ghosh [4] presented a methodology of integrating the design and planning of manufacturing by utilizing the concept of feature-based modeling. Patil and Pande [9] reported the development of an intelligent environment for the feature-based design synthesis and process planning of prismatic parts. Khoshnevis et al. [10] described architecture of an integrated process planning system, called 3I-PP, which is comprised of three modules: feature completion, process selection, and pro-

cess sequencing. Tseng [11] developed a feature-based fixturing analysis method for machining multiple sets of features. Joo and Cho [12] presented a feature-based process planning for sculptured pocket machining.

As revealed in the literature presented above, the application of neural networks and fuzzy logic are limited to discrete functions of CAPP systems, but it has also been shown that these approaches can significantly improve the performance of CAPP systems. On the integration side, most of the feature-based models used in process planning have no standard explicit representation and the features are represented in an implicit data format.

This paper focuses on developing a feature-based intelligent CAPP system called ST-FeatCAPP. The main purposes of this CAPP system are: (1) to integrate a standardized feature-based model with process planning utilizing the concept of STEP-based features, and (2) developing a hybrid intelligent CAPP system by combining artificial neural networks, fuzzy logic, and rule-based techniques, and (3) to generate a digital process plan containing all the data required to produce the given part. The inference functions which will be used by the proposed CAPP system are classified into: (a) a neural network based function which means that certain CAPP sub-functions will be performed by utilizing the neural networks based

Fig. 1 ST-FeatCAPP architecture



inference method, (b) the fuzzy logic inference function which will be performed by adopting the fuzzy logic inference method, and (c) the rule-based function which ensures that certain CAPP sub-functions will be carried out based on a set of pre-defined rules.

---

## 2 ST-FeatCAPP architecture

Figure 1 shows the architecture of the proposed CAD/CAPP system. The overall system consists of two main subsystems: the STEP-based feature modeler (STEP-FM) and STEP-based intelligent process planning (ST-FeatCAPP). The first system uses STEP features as the basic entities for part design and generates the corresponding STEP AP224 XML data [13]. The second system, which is the subject of this paper, consists of the following tasks [14]:

- Integration and preparation of part design data
- Selection of machining operations
- Selection of cutting tools
- Selection of machine tools
- Determination of machining parameters
- Generation of setup plans
- Generation of digital process plan

In the architecture, the part is first created by the feature modeler (STEP-FM) using a feature-based representation. The part design data is then generated by a pre-processor and saved in the STEP XML file. This file will be passed to ST-FeatCAPP through two steps. The first step utilizes a post-processor to map step feature data and its associated attributes, while the second step prepares the mapped data to be fed to various ST-FeatCAPP tasks. Once the part data is entered into the system, the ST-FeatCAPP components can be carried out concurrently. In other words, the process planner can perform any task of ST-FeatCAPP at any time in the process planning cycle. The details on how ST-FeatCAPP components are implemented will be described in the following sections.

---

## 3 Integration of ST-FeatCAPP with CAD

A huge research work has been carried out over the past two decades mainly to integrate design, CAD, and manufacturing, CAM. It was recognized that in order to achieve CAD/CAM integration, it is necessary to devise part models that contain more information than geometry and topology. CAD systems have become a common tool in industries to design parts, but the level of part model data is too low. Part models in CAD systems (2D or 3D) are represented at a low level, for example, geometry for wireframe models, geometry and topology for B-rep models and primitives and operators for CSG models. None of these can be directly used in downstream applications, such as CAPP systems. This is because information on higher-level features is missing such as tolerances, material specification, and technological data

related to manufacturing [4, 9, 13]. For true integration of CAD with CAPP, a standard feature-based model is needed. Such a model should contain explicit high-level part data in terms of feature attributes, tolerances, material specifications, etc. The model data can be used directly with process planning systems without using a complex feature recognition process. STEP is one such standard model to be implemented to solve this problem.

STEP is an unofficial name; the actual designation of the STEP standard is ISO10303 industrial automation systems—product data representation and exchange [15]. STEP provides a computer interpretable representation and exchange of product data in a formal manner suitable for communication, interpretation, or processing by human beings or by computers. The objective of STEP is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product independent of any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving. STEP uses a formal specification language, EXPRESS [16], to specify the product information to be represented. The use of a formal language enables precision and consistency of representation and facilitates development of implementations. STEP uses application protocols (APs) to specify the representation of product information for one or more applications [15]. The application protocol implemented in this research work is based on STEP AP224 [17]. It specifies the requirements for the representation and exchange of information needed to define product data necessary for manufacturing single piece mechanical parts. The product data is based on existing part designs that have their shapes represented by form features.

In this research work, the integration of the ST-FeatCAPP system with CAD is implemented through mapping STEP AP224 XML data generated by a feature-based modeler developed by the authors [13]. This implies that features data which already exist in the feature model need to be presented in a suitable format for process planning [18]. These data are used for obtaining the type of machining features and their attributes (tolerances, surface finish, etc.) which are present in the part.

ST-FeatCAPP has a post-processor designed to convert design data (STEP AP224) into a suitable format to be used as input for process planning tasks. The post-processor mechanism used in the translation process is based on two layers [13]. In the first layer, usually known as the logical layer, EXPRESS language is used to represent each feature in EXPRESS representation format which describes the parameters associated in the definition of the feature. The second layer of the post-processor translator is known as the physical layer. This layer maps the communication file structure called STEP file and converts STEP data into EXPRESS schemes defined in the logical layer. These EXPRESS schemes can be interpreted easily and its contents can be passed to process planning tasks. Figure 2 shows a sample STEP AP224 and the mapping process.

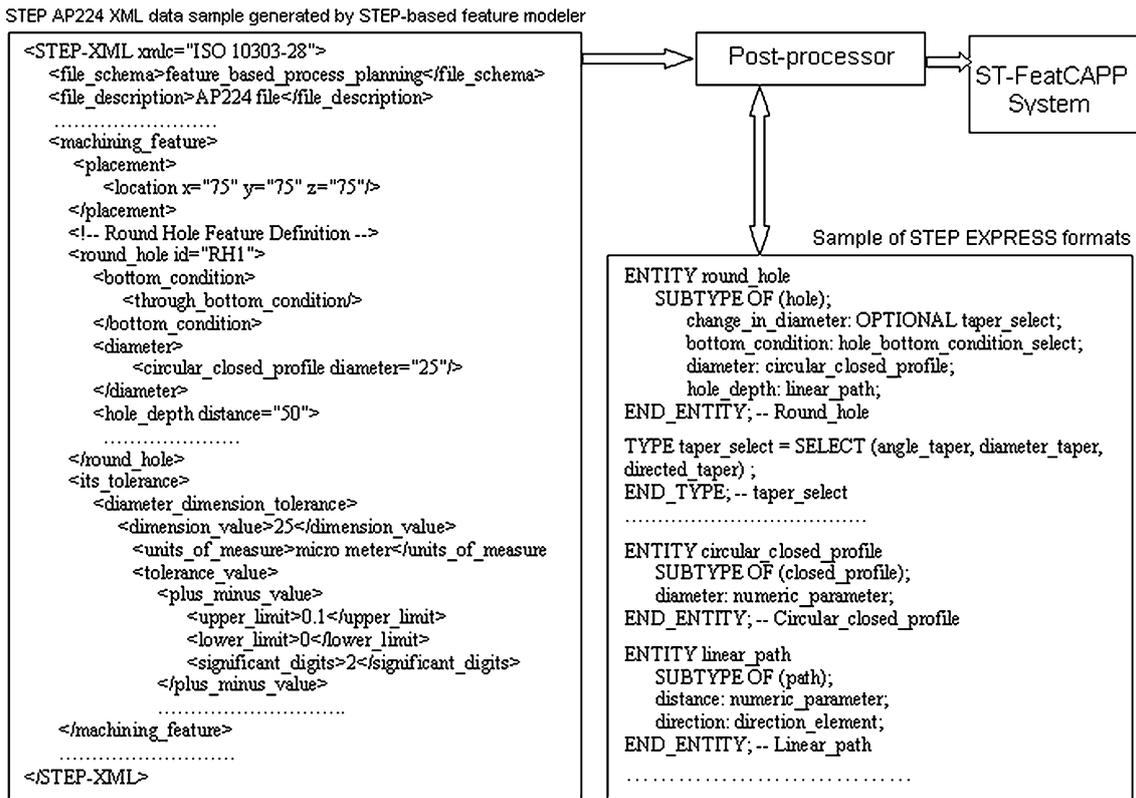


Fig. 2 Mapping STEP AP224 data to the ST-FeatCAPP system

An object-oriented approach has been implemented to facilitate the use of the translation process from STEP XML format to EXPRESS format. All the EXPRESS schemes defining STEP features have been programmed in class files. Each feature has an algorithm that uses the EXPRESS class files to map the required STEP feature data format.

#### 4 Application of artificial neural networks in ST-FeatCAPP system

Several research efforts have reported the application of artificial neural networks in manufacturing [2-5]. It is clear that their applicability in automated process planning still has to be verified [3]. The ST-FeatCAPP system uses three neural networks to carry out some tasks of the process planning. These tasks are summarized as follows: selection of machining operations, selection of cutting tool, and selection of machine tools.

##### 4.1 Selection of machining operations

This task functionally receives data for each feature in the part and generates the needed machining operations to realize the feature on the part. Figure 3 shows the neural network used to carry out the task of selection of machining operations in the ST-FeatCAPP system. It consists of three

fully connected layers, namely, the input layer, the output layer and the hidden layers. The input layer of the neural network consists of six input variables. The hidden layers have 15 neurons each determined by conducting a number of experiments. The output layer of the neural network consists of thirteen variables corresponding to the machining operations required for each feature on the part. Each output neuron has a value of 1 or 0. If the output neuron value is equal to 1 then it is interpreted as meaning that the selection of the machining operation is supported.

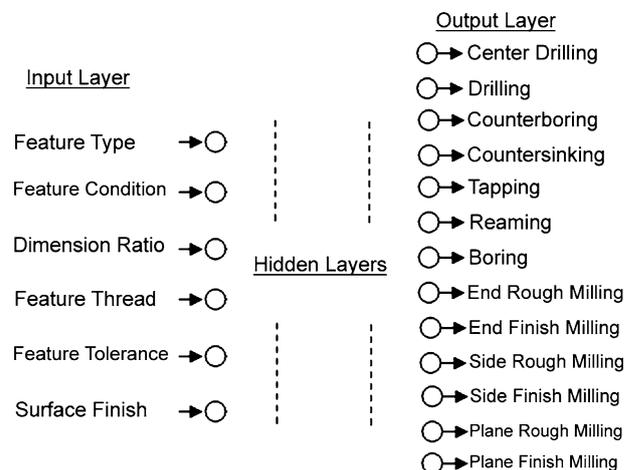


Fig. 3 Machining operations selection neural network model

The input values to the network corresponding to the feature type, feature condition, and feature thread are properly encoded to be interpreted by the neural network. The other input variables corresponding to feature dimension ratio, feature tolerance, and feature surface finish are scaled appropriately to be in the range between 0 and 1 to facilitate the training of the network.

The basic idea in this task is that for every machining feature there is a corresponding machining operation, depending upon the technological requirements of the feature. For example, for the same slot feature dimensions, if the tolerance or surface finish requirements are considered, then a finishing operation might be needed. However, if these requirements are omitted, then a roughing operation alone might be sufficient. The neural network has been trained according to this criterion.

#### 4.2 Selection of cutting tools

In traditional process planning, the selection of cutting tools is based primarily on experience which is error-prone and insufficient with respect to standardization. In order to eliminate these problems, so as to be able to achieve cutting automation, the development of a computer-aided intelligent tool selection model is needed. In this work of ST-FeatCAPP, several neural network models have been developed. The main neural network model is utilized to select the proper cutting tool for each machining feature. The selection is based upon machining feature and its associated machining operation. The basic idea in selection is that for each machining feature and machining operation combination there is a corresponding cutting tool to be used to generate that feature. For example, for square slot to be machined with an end milling operation, a flat end mill might be selected, while round slot with the same machining operation may call for a ball end mill. The neural network is trained based upon this criterion. Figure 4 shows the neural network model for cutting tool selection. It consists of five input variables, two hidden layers with fifteen neurons each and twenty output variables. The input variables correspond to feature type, feature attributes and machining operation type. The input values are appropriately encoded and scaled to facilitate the network training. The output variables correspond to the cutting tool types. Each output variable has a value of 1 or 0. If the output variable value is equal to 1 then it is interpreted that the selection of the cutting tool is supported.

The cutting tools library used in the ST-FeatCAPP system is based on the STEP-NC (ISO14649) standard [19]. It consists of 20 cutting tools as shown in Fig. 4. When the cutting tool is selected by the neural network, ST-FeatCAPP starts searching the standard tool dimensions database to find the proper tool dimensions to fit the machining setup. Besides, for each cutting tool a neural network was designed and trained to select the proper tool geometry. The selection is based upon the information about workpiece material and feature size.

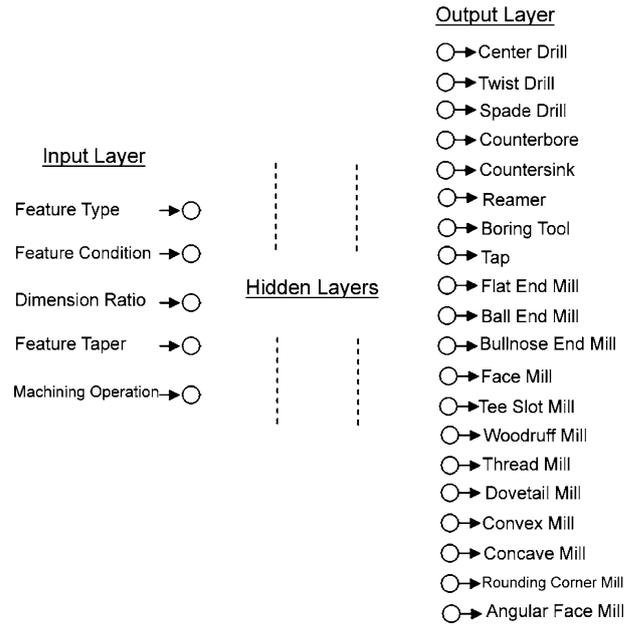


Fig. 4 Cutting tools selection neural network model

#### 4.3 Selection of machine tools

This task of ST-FeatCAPP selects the machine tools on which the machining operations can be performed to produce the given part. This task is implemented using a neural network. The input vector of the neural network includes machining part characteristics (i.e. part type, part size, feature tolerances and feature surface finishes) and machining operation characteristics (i.e. operation type and machining power). The output vector of the neural network contains recommended specifications of the machine tool to be used to perform the task. These recommended specifications can be used as keys to search available machine tool databases to find the proper machine tool. Figure 5 shows the neural network structure used for this task.

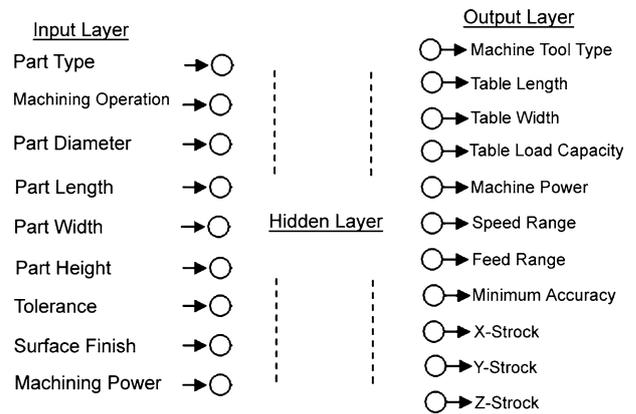


Fig. 5 Machine tools selection neural network model

## 5 Training of neural networks

### 5.1 Training algorithm

Once the neural network has been designed, it has to be trained to produce the expected output values as functions of a predefined pattern of input values. This training operation is accomplished by selecting a proper training algorithm for the problem to be solved. Several training algorithms have been developed for neural networks. Many of these training algorithms are closely connected with a certain network topology. Among various existing training algorithms, the backpropagation algorithm was selected in this research work. It is a commonly used algorithm, relatively easy to apply and has been proven to be successful in practical applications [20]. The backpropagation algorithm is a gradient decent method to minimize the total sum of square error over the entire training data set. Convergence to the optimal solution is accomplished by adjusting the weight connections through the partial derivative of the sum-squared error with respect to the weights. The following steps summarize the implementation of this algorithm for training the designed neural networks [21].

#### Step 1

Set all the necessary network parameters such as the number of input neurons, the number of hidden layers and the number of neurons included in each hidden layer, the number of output neurons, etc.

#### Step 2

Set all network weights to small random values, positive and negative (-0.3 to 0.3).

#### Step 3

Initialize the iteration (epoch) number ( $m=1$ ) and presentation (example) number ( $n=1$ ).

#### Step 4

Apply one training sample to the input layer  $[X_1, X_2, \dots, X_{N_k}]$  and note the corresponding desired output  $[O_1, O_2, \dots, O_{N_k}]$ , where  $N_k$  is the number of neurons in layer  $k$ .

#### Step 5

Calculate the output of the neurons layer by layer through the network, from the second layer to the output layer using:

$$O_j^{N_k}(n) = F\left(\sum_{i=1}^p W_{ji}(m) O_i^{N_{k-1}}(n)\right)$$

for each neuron  $j$ ,  $1 \leq j \leq N_k$  and  $2 \leq k \leq L$

where  $L$  is the number of layers in the network.

- $F$  is a sigmoid activation function of the form  $F(a) = 1/(1 + e^{-a})$ .
- $W_{ji}(m)$  is the weight connecting neuron  $i$  in layer  $k$  to neuron  $j$  in layer  $k+1$ .
- $O_i^{N_k}$  is the output of neuron  $N$  in layer  $k$ .

#### Step 6

Calculate the error gradient  $\delta$  for every neuron in every layer in backward order from output to the first hidden layer. The error for the output layer neurons is computed by

$$\delta_j^{N_k}(n) = O_j^{N_k}(n) \left(1 - O_j^{N_k}(n)\right) \left(T_j(n) - O_j^{N_k}(n)\right)$$

for every neuron  $j$ ,  $1 \leq j \leq N_k$ , and  $k=L$  where  $T_j(n)$  is the target vector.

Then, successively, the error gradients for all hidden layer neurons are computed from

$$\delta_j^{N_k}(n) = O_j^{N_k}(n) \left(1 - O_j^{N_k}(n)\right) \sum_{i=1}^{N_{k+1}} \delta_i^{N_{k+1}}(n) W_{ij}(m)$$

for every neuron  $j$ ,  $1 \leq j \leq N_k$ , and  $k=L-1, \dots, 2$ .

At the end of the error backward propagation step, neurons of the network will have error values (except input layer neurons,  $L=1$ ).

#### Step 7

Adjust the network weights for every layer. Starting at the output layer neurons and working back to the first hidden layer recursively adjust weights according to the generalized delta rule.

$$\begin{aligned} W_{ji}(m+1) &= W_{ji}(m) + \eta \delta_j^{N_k}(n) O_i^{N_k}(n) \\ &+ \alpha [W_{ji}(m) - W_{ji}(m-1)] \end{aligned}$$

for every neuron  $j$ ,  $1 \leq j \leq N_k$ , and  $k=L, L-1, \dots, 2$  where  $\alpha$  is a momentum constant ( $0 < \alpha < 1$ ) to smooth out the weight change and accelerate convergence of the network and  $\eta$  is learning rate ( $0 < \eta < 1$ ) which controls the step size for weight adjustments.

#### Step 8

Repeat actions in steps 4 to 7 for every training sample.

#### Step 9

Calculate the average sum-squared error resulting at the end of every training cycle. This error can be evaluated by the following expression:

$$sse = \frac{1}{2N} \sum_{j=1}^n \sum_{i=1}^{N_L} \left(T_{ij} - O_{ij}^L\right)^2$$

where  $T_{ij}$  is the target value desired for the  $i$ th output and for the  $j$ th example.

#### Step 10

Compare the average sum-squared error ( $sse$ ) with the tolerance value ( $\epsilon$ ) of the error, if it is less than the tolerance value, then stop; otherwise, increase the num-

ber of iterations and randomize the order in the training set and return to step 4.

## 5.2 Training data patterns

A successful neural network requires that the training data set and training procedure be appropriate to the problem. The training data set must span the total range of input patterns sufficiently well so that the trained network can make generalizations about the data. In order to have extrapolation and interpolation capabilities, neural networks must be trained on a wide enough set of input data to generalize from their training sets. To achieve this goal and demonstrate the applicability of the designed neural networks, a number of training patterns (each pattern is formed by input and output vectors) are generated for each task. The input values of the training patterns are selected from within a specified range for each input parameter. The output values are based upon the limitations put on each variable.

## 5.3 Training experiments

Several training experiments have been performed to select the optimal structure and training parameters of the neural networks. The results obtained are presented in Table 1.

The graph shown in Fig. 6 represents the training set average error on the  $y$ -axis against the number of epochs elapsed on the  $x$ -axis. Epochs represent a complete pass through the network of the entire set of training patterns. The graph illustrates downward movement of the error rate

Table 1 Architectures and training parameters of the neural network models

Network parameter	Machining operations selection network	Cutting tools selection network	Machining tool selection network
Input neurons	6	5	9
Output neurons	13	20	11
Hidden layers	2	2	2
Neurons in hidden layers	15	25	13
Learning rate	0.1	0.1	0.25
Momentum	0.9	0.9	0.9
Number of training patterns	434	214	25
Number of training cycles (epoch)	5000	5000	3000
Average sum-squared error	0.0003	0.0015	0.0007
Mode of training	Batch	Batch	Batch

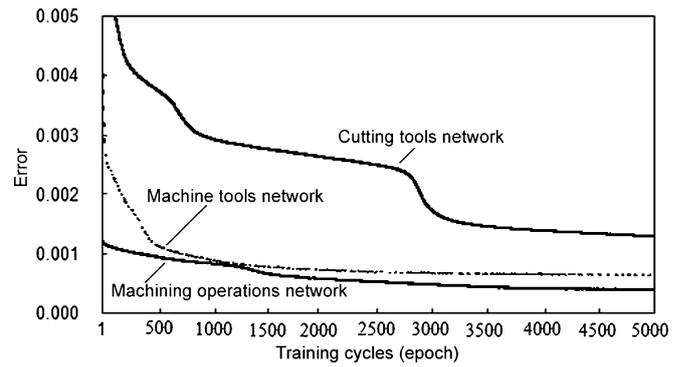


Fig. 6 Training progress of the neural networks

as learning progressed, indicating that the average error decreased between actual and predicted results.

## 6 Application of fuzzy logic in ST-FeatCAPP system

Machining parameters such as cutting speed and feed rate affect the production rate, quality and cost of a component during a machining operation. The selection of process parameters is traditionally carried out by process planners or experienced machinists, either on the basis of their experience on the shop floor or with the help of handbooks, for achieving the desired surface quality on the component. The parameters thus selected are generally on the conservative side to avoid machining failure [22].

Over the past decade, process planners have optimized the machining parameters using some optimization algorithms. The optimization of machining parameters is a nonlinear optimization with constraints, so it is difficult for traditional optimization algorithms to solve this problem because of the problems of convergence speed or accuracy [23].

Recently, process planners have started using artificial intelligent techniques, such as neural networks, fuzzy logic and genetic algorithms to select the machining parameters and have made some progress. In this research work, a fuzzy logic model is proposed to select machining parameters in drilling and milling operations.

An example of one of these models capable of determining the machining parameters of a drilling operation is presented in order to explain the steps in the development of a fuzzy model. There are three basic components of the fuzzy model: fuzzification of the input, fuzzy rules application and defuzzification of the output [7, 24, 25]. The hardness of the work material and the hole diameter are the input variables and cutting speed and feed rate are the output variables of the fuzzy model for drilling. The fuzzy sets of the input and output variables are shown in Table 2. Triangular shaped membership functions were employed to describe the fuzzy sets. Figure 7 shows the membership functions for the input and output variables. The universe of input, material hardness and hole diameter has been partitioned according to the minimum and maximum values allowed to control the model. Similarly, the universe of output, cutting speed and feed rate has been

Table 2 Fuzzy sets of input and output variables

Input variables			Output variables		
Fuzzy set	Range	Abbreviation	Fuzzy set	Range	Abbreviation
Material hardness [BHN]			Cutting speed [m/min]		
Very soft	<100	VS	Very low	<15	VL
Soft	50–250	S	Low	10–40	L
Medium	150–350	M	Medium	25–80	M
Hard	250–450	H	High	52–110	H
Very hard	>350	VH	Very high	>80	VH
Hole diameter [mm]			Feed rate [mm/rev]		
Very small	<3	VS	Very slow	<0.1	VS
Small	2–10	S	Slow	0.05–0.3	S
Medium	6–20	M	Medium	0.15–0.45	M
Large	13–40	L	Fast	0.3–0.9	F
Very large	>26.5	VL	Very fast	>0.45	VF

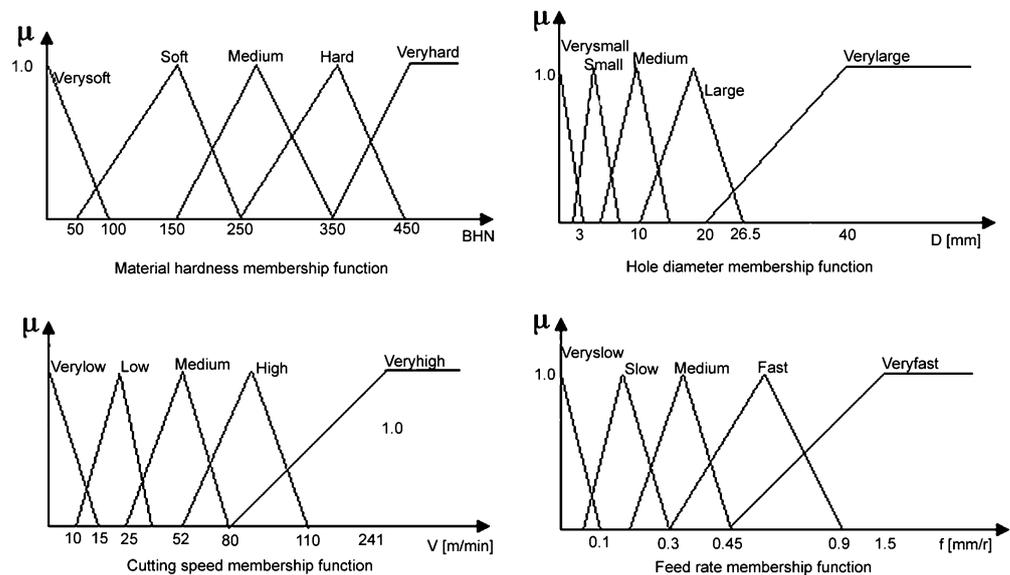
partitioned according to the required range for each output. A set of fuzzy rules has been developed for different work and tool material combinations. Table 3 shows fuzzy rules for the drilling operation with a high speed steel tool. Several defuzzification techniques are available in the literature. The choice of defuzzification method may have a significant impact on the accuracy of the fuzzy model output [21, 25]. The most frequently used method is the *centroid* or *center of area* (COA) which is used in the developed fuzzy models.

## 7 Application of rule-based knowledge in ST-FeatCAPP system

The setup planning activity in ST-FeatCAPP is composed of three steps: setup generation, operation sequence, and

setup sequence [6]. The *setup generation* is a procedure to group the machining operations into setups such that the manufacturing features which have common approach directions are grouped into the same setup. The *operation sequence* arranges the machining operations in each generated setup into an order so that the constraint of the feature precedence relationships in each setup is satisfied. In addition, the cutting tool changes among the operations are reduced to a minimum. The *setup sequence* is to arrange the generated setups into an order so that setups with fewer number of machining features are machined first. A set of rules guiding the setup planning have been developed for prismatic parts and implemented in the ST-FeatCAPP system. The following subsection explains the main steps of the implemented algorithm to generate the required setup plans.

Fig. 7 Membership functions for input and output variables



**Table 3** Fuzzy rules for cutting speed and feed rate with high speed steel tool

Material Hardness	Hole Diameter					
	S	VS	M	L	VL	
VS	V	M	M	M	M	M
	f	VS	S	M	F	F
S	V	L	L	L	L	L
	f	VS	S	M	F	F
M	V	L	L	L	L	L
	f	VS	S	M	M	F
H	V	VL	VL	VL	VL	VL
	f	VS	S	S	M	F
VH	V	VL	VL	VL	VL	VL
	f	VS	S	S	M	M

**7.1 Setup planning algorithm**

In this research work, a feature-based model is used to aid computerizing the algorithm for setup planning. The algorithm is illustrated in the following steps [6, 11]:

**Step 1**

Generation of setup plans

1. Define the part coordinate system and assign tool access direction (TAD) for a prismatic part machined on a 3-axis milling center (see Fig. 8). Only one of the six TADs can be set per setup:

TAD<sub>1</sub> defined by (1, 0, 0), +X direction, also named YZ-left face.

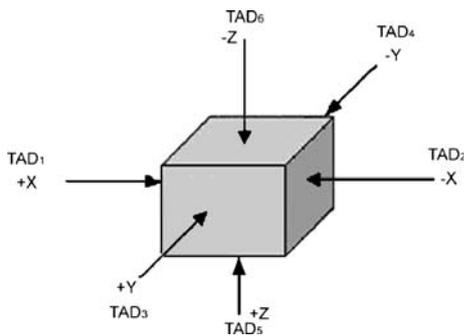
TAD<sub>2</sub> defined by (-1, 0, 0), -X direction, also named YZ-right face.

TAD<sub>3</sub> defined by (0, 1, 0), +Y direction, also named XZ-front face.

TAD<sub>4</sub> defined by (0, -1, 0), -Y direction, also named XZ-back face.

TAD<sub>5</sub> defined by (0, 0, 1), +Z direction, also named XY-bottom face.

TAD<sub>6</sub> defined by (0, 0, -1), -Z direction, also named XY-top face.



**Fig. 8** Tool access directions

2. Define six setup plans corresponding to each TAD:  $S_{+x}$  for TAD<sub>1</sub>,  $S_{-x}$  for TAD<sub>2</sub>,  $S_{+y}$  for TAD<sub>3</sub>,  $S_{-y}$  for TAD<sub>4</sub>,  $S_{+z}$  for TAD<sub>5</sub>,  $S_{-z}$  for TAD<sub>6</sub>
3. Assign a definite TAD to every feature so that it can be assigned to a definite setup.

**Step 2**

Sequence of machining operation in each setup.

Every setup plan ( $S_{+x}$ ,  $S_{-x}$ ,  $S_{+y}$ ,  $S_{-y}$ ,  $S_{+z}$ ,  $S_{-z}$ ) contains features.

4. Sequence machining operations according to the machining features precedence.
5. Arrange machining operations according to the natural operation sequence (roughing operations prior to finishing operations).
6. Rearrange the drilling type operations according to the following sequence: center drilling + drilling + counterboring or countersinking + tapping + boring or reaming or milling.
7. Minimize the number of tool changes by rearranging same type machining operations.

**Step 3**

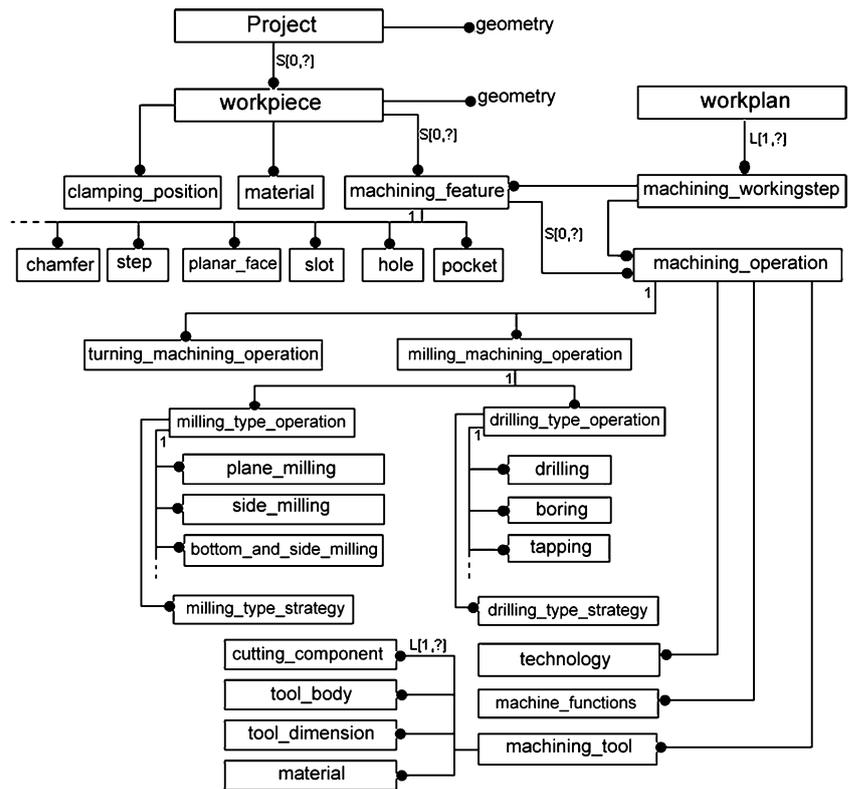
Sequence of setups

8. Sequence the setup plans so that setups with less machining features are machined first.

**8 Generation of digital process plans**

A preprocessor has been developed to generate a physical STEP-NC process plan file. The main function of this preprocessor is to receive information related to the workpiece (features, tolerances, material, etc.) and information produced by process planning (machining operations, cutting tools, machining parameters, etc.) to generate a STEP-NC process plan. STEP-NC, ISO 14649 [26], basically consists of structure feature-based process plans for manufacturing processes such as milling operations, turning operations, etc. Currently, STEP-NC is developed for milling operations based on feature information from ISO 10303-AP224 [27]. As shown in Fig. 9, each STEP-NC process plan must include exactly one top-level entity called *project*. The project indicates the workplan to be executed upon interpretation of this workplan, and it may also provide the workpiece upon which actions are to be performed. The entity workplan allows combination of several workingsteps in a linear order. These workingsteps provide the basis of the process plan to manufacture the component. Each workingstep defines a machining operation to be performed on a machining feature. The machining features describe what the machine shall do and the workingsteps and operations describe how the features should be manufactured with specific machining tool and technological parameters.

Fig. 9 Information contents of STEP-NC process plan



Consider the example part shown in Fig. 10. This part was built using the feature modeler (STEP-FM) and then processed by ST-FeatCAPP to generate a STEP-NC process plan [26, 28]. The Appendix illustrates the partial physical file in STEP XML format for the example part. This file contains the definition of the part, the machining features, the cutting tools, the material and tolerances, machine tool instructions, etc. Some of the important features of this STEP-NC process plan are as follows:

- The STEP-NC process plan provides an object oriented data model for CNCs with a detailed and structured

data interface that incorporates feature based programming where there is a range of information such as the feature to be machined, type of tools used and the operations to perform [29].

- The STEP-NC process plan not only eliminates the costly and inefficient process of generating post processors, but also establishes a collaborative environment for the exchange of information between product design applications, manufacturing process planning, and the machine tool on the factory floor.

Fig. 10 Example part

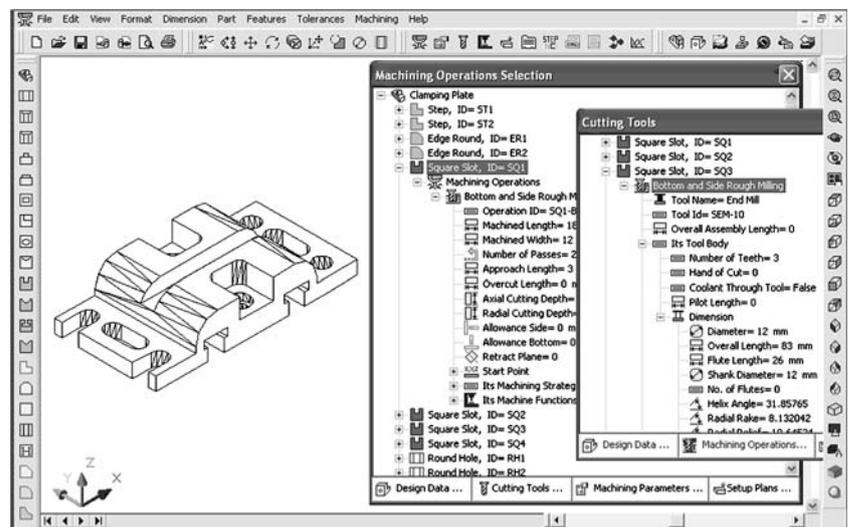
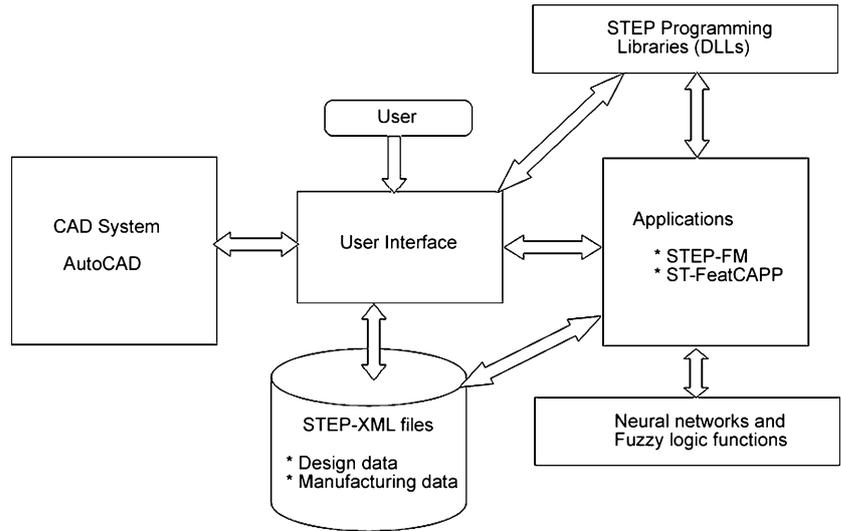


Fig. 11 System architecture



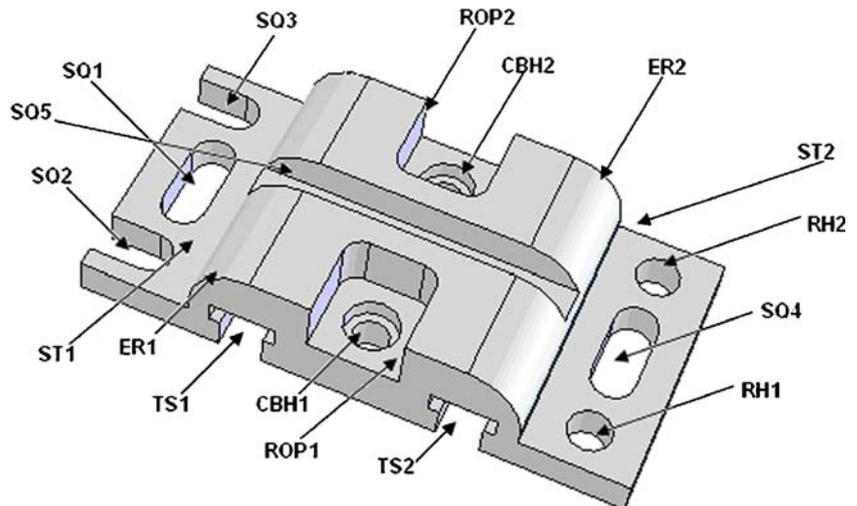
### 9 System implementation

This section discusses the implementation of the STEP-based feature modeler [13] and the feature-based intelligent process planning systems. The implemented systems are named: STEP-FM, an acronym for STEP-feature modeler; and ST-FeatCAPP, an acronym for STEP feature-based process planning. Figure 11 shows the architecture of the two systems in an integrated environment with AutoCAD. The integrated environment of the developed system consists of the following: (1) AutoCAD, (2) STEP-FM, (3) ST-FeatCAPP, (4) STEP features DLLs, (5) STEP XML data files, and (6) user interface. AutoCAD is used as an integrator (container) environment for the developed systems. The advantage of using AutoCAD is that it includes a complete Activex automation interface with functions or methods that can be called from visual BASIC. Furthermore, AutoCAD shares the same solid modeling engine as several other CAD systems.

The user interface has been developed using Visual BASIC for Application (VBA). Microsoft VBA is an object-oriented programming environment designed to provide rich development capabilities similar to those of Visual BASIC (VB). The main difference between VBA and VB is that VBA runs in the same process space as AutoCAD, providing an AutoCAD-intelligent and very fast programming environment. VBA sends messages to AutoCAD by the AutoCAD Activex automation interface. AutoCAD VBA permits the VBA environment to run simultaneously with AutoCAD. This coupling of AutoCAD, Activex automation, and VBA provides an extremely powerful interface not only for manipulating AutoCAD objects, but for sending data to or retrieving data from other applications [30].

This research work applies the object-oriented approach to construct the product data model and corresponding design and process planning function models. The data model implemented in the developed system is based on

Fig. 12 Example part model



the application objects defined within STEP AP224 (ISO 10303-224) and STEP-NC (ISO 14649). These objects cover information related to: manufacturing features, tolerances, technological attributes, part administrative data, machining operations, cutting tools, etc. Each application object is programmed in a class file with its public properties and operations. The inheritance feature of the object-oriented approach is used to define other common properties and to interface with the operations required to be performed on the instanced objects [31, 32].

---

## 10 Conclusions

Integration of CAPP with CAD systems will greatly improve the efficiency of manufacturing enterprises. In addition, the development of STEP-based intelligent CAPP systems will be very useful to the manufacturing engineering specialists working in a concurrent engineering environment. The present work demonstrates a hybrid intelligent process planning system using STEP-based features (ST-FeatCAPP). The system maps a STEP AP224 XML data file, without using a complex feature recognition process, and produces the corresponding machining operations to generate the process plan and corresponding STEP-NC in XML format. ST-FeatCAPP carries out several stages of process planning utilizing a hybrid approach of most recent techniques of artificial intelligence as the inference engine. Artificial neural networks have been used to select machining operations, cutting tools, and machine tools. Several neural networks have been designed, trained and implemented in the ST-FeatCAPP system to carry out the above mentioned tasks of process planning. A number of fuzzy logic models have been implemented to select machining parameters for several machining operations, cutting tool materials, and workpiece material combinations. Setup planning is carried out based on a number of machining rules utilized by a simple setup planning algorithm. The use of a hybrid approach of neural networks and fuzzy logic has enabled the development of a flexible CAPP system that can be trained to handle new knowledge.

The output of the ST-FeatCAPP system is a STEP-NC process plan which can be used to control the next generation of intelligent machine tool controllers.

This research takes major steps towards the application of advanced artificial intelligence in computer aided process planning by developing a hybrid approach consisting of artificial neural networks and fuzzy logic. However, this approach can be extended to carry out other tasks of process planning including setup planning, fixture planning, and manufacturing cost estimation. In addition, the use of optimization algorithms for learning neural networks and fuzzy logic can enhance the ability of the CAPP system to handle complex parts.

The STEP-NC process plan is currently being validated in pilot projects both in the US (titled as ‘Super Model’) and in Europe (titled as ‘STEP-NC’). To date, validation has focused on feature-based machining for a limited number of features. This research has focused on producing STEP-NC process plans for a limited number of machining features without tool path planning. In STEP-NC, tool path planning is optional information which can be added to the process plan at later stages. Perhaps the most important future work on STEP-NC will include more machining features and generating path planning to enable true intelligent control of CNC machines.

**Acknowledgements** The authors gratefully acknowledge the Turkish State Planning Department (DPT), the Turkish Scientific and Technical Research Council (TÜBİTAK) and the Integrated Manufacturing Technologies Research Group (IMTRG) for their support. Also, thanks to SAYYSAL GRAFYK industrial and trade comp. for providing the AutoCAD package.

---

## Appendix

Figure 12 shows an example part tested by the developed CAPP system. The corresponding STEP-NC process plan, generated by the system, is partially presented in this appendix to demonstrate the results of the ST-FeatCAPP system.

```

<STEP-XML xmlc="ISO 10303-28">
  <file_schema>integrated_cnc_schema</file_schema>
  <file_description>ISO14649 file</file_description>
  <file_name>example</file_name>
  <project its_id="Example - 1">
    <its_workpiece>
      <part its_name="Clamping Plate">
        <its_material>
          <material_id>Carbon steel</material_id>
          <material_description>Medium Carbon Steel</material_description>
          <material_hardness>
            <scale>BHN</scale>
            <high_value>350</high_value>
            <low_value>150</low_value>
            <nominal>250</nominal>
          </material_hardness>
        </its_material>
        <its_rawpiece>
          .....
        </its_rawpiece>
        <its_geometry>
          <block_base_shape width="75" height="30" length="150">
            <placement>
              <location x="50" y="50" z="25"/>
              .....
            </placement>
          </block_base_shape>
        </its_geometry>
      </part>
      <!-- Part administrative data -->
      <administrative_data>
        .....
      </administrative_data>
      <!-- Here come the main process plan-->
      <main_workplan>
        <its_elements>
          .....
          .....
          <machining_workingstep>
            <its_id> WS653</its_id>
            <its_feature>
              <usage_name>Fixing</usage_name>
              <placement>
                <location x="65" y="87.5" z="35"/>
                .....
              </placement>
              <!-- Square Slot Feature Definition -->
              <slot id="SQ1">
                <end_condition>
                  <first_end_condition>
                    <radiused_slot_end_type/>
                  </first_end_condition>
                  <second_end_condition>
                    <radiused_slot_end_type/>
                  </second_end_condition>
                </end_condition>
                <course_of_travel distance="30">
                  </course_of_travel>
                <square_u_profile>
                  <first_angle>0</first_angle>
                  <first_radius>0</first_radius>
                  <second_angle>0</second_angle>
                </square_u_profile>
              </slot>
            </its_feature>
          </machining_workingstep>
        </its_elements>
      </main_workplan>
    </project>
  </its_workpiece>
</STEP-XML>

```

```

    <second_radius>0</second_radius>
    <width>12</width>
    <depth>10</depth>
  </square_u_profile>
</slot>
<its_tolerance>
<!-- Size Tolerance -->
<curved_dimension_tolerance>
  <dimension_value>6</dimension_value>
  <units_of_measure>milli meter</units_of_measure>
  <significant_digits>3</significant_digits>
  <dimension_description>Slot Radius</dimension_description>
  <tolerance_value>
    <plus_minus_value>
      <upper_limit>0.1</upper_limit>
      <lower_limit>0</lower_limit>
      <significant_digits>0</significant_digits>
    </plus_minus_value>
  </tolerance_value>
</curved_dimension_tolerance>
</its_tolerance>
</its_feature>
<its_operation>
<!-- End milling Machining Operation -->
<bottom_side_rough_milling its_id="SQ1-BSRM-R">
  <overcut_length>0.00</overcut_length>
  <axial_cutting_depth>4.50</axial_cutting_depth>
  <radial_cutting_depth>5.00</radial_cutting_depth>
  <allowance_side>1.00</allowance_side>
  <allowance_bottom>1.00</allowance_bottom>
  <its_machining_strategy>
    <overlap>0.00</overlap>
    <allow_multiple_passes>True</allow_multiple_passes>
    <unidirectional/>
  </its_machining_strategy>
<!-- End Mill Cutter -->
<its_tool>
  <endmill its_id="SEM-10">
    <its_toolbody>
      <number_of_teeth>3</number_of_teeth>
      <coolant_through_tool>False</coolant_through_tool>
      <dimension>
        <diameter>10.00</diameter>
        <tool_length>72.00</tool_length>
        <flute_length>22.00</flute_length>
        <helix_angle>32.09</helix_angle>
        <radial_rake>8.04</radial_rake>
        <radial_relief>10.42</radial_relief>
        <radial_clearance>15.37</radial_clearance>
        <cutting_edge_angle>0.99</cutting_edge_angle>
        <end_relief>6.00</end_relief>
      </dimension>
    </its_toolbody>
    <its_cutting_edge>
      <its_material>
        <material_id>HSS</material_id>
        <material_description></material_description>
        <material_hardness>
          <scale>BHN</scale>
          <high_value>550.00</high_value>
          <low_value>250.00</low_value>
          <nominal>350.00</nominal>
        </material_hardness>
      </its_material>
    </its_cutting_edge>
  </endmill>
</its_tool>
</its_operation>
</its_feature>

```

```

        </material_hardness>
      </its_material>
    </its_cutting_edge>
  </endmill>
</its_tool>
<its_technology>
  <cutting_speed>19.76</cutting_speed>
  <feed_speed>93.755</feed_speed>
  <spindle_speed>629</spindle_speed>
  <total_depth_of_cut>9.00</total_depth_of_cut>
  <depth_of_cut>4.50</depth_of_cut>
  <passes>2</passes>
  <total_width_of_cut>10.00</total_width_of_cut>
  <width_of_cut>10.00</width_of_cut>
  <loops>1</loops>
  <machined_length>20.00</machined_length>
  <machined_time>0.43</machined_time>
  <removal_rate>4,218.98</removal_rate>
  <power_at_spindle>0.25</power_at_spindle>
  <power_at_motor>0.32</power_at_motor>
  <torque_at_spindle>3.84</torque_at_spindle>
</its_technology>
<its_machine_functions>
  <coolant>True</coolant>
  <cutting_fluids>
    <Oils-Light_Duty>
      <Straight_Mineral_Oil/>
    </cutting_fluids>
  <coolant_pressure>0</coolant_pressure>
  <mist>False</mist>
  <through_spindle_coolant>False</through_spindle_coolant>
  <through_pressure>0.00</through_pressure>
  <chip_removal>False</chip_removal>
</its_machine_functions>
</bottom_side_rough_milling>
</its_operation>
</machining_workingstep>
.....
.....
</its_elements>
</main_workplan>
</its_workpiece>
</project>
</STEP-XML>

```

## References

- Zhang H, Altung L (1994) Computerized manufacturing process planning systems. Chapman & Hall, London
- Zhang H-C, Huang SH (1995) Applications of neural networks in manufacturing: a state-of-the-art survey. *Int J Prod Res* 33:705–728
- Santochi M, Dini G (1996) Use of neural networks in automated selection of technological parameters of cutting tools. *CIM Syst* 9:137–148
- Devireddy CR, Ghosh K (1999) Feature-based modeling and neural networks-based CAPP for integrated manufacturing. *Int J Comput Integ Manuf* 12:61–74
- Mei J, Zhang H-C, Oldham WJB (1995) A neural network approach for datum selection in computer-aided process planning. *Comput Ind* 27:53–64
- Ming XG, Mak KL (2000) Intelligent setup planning in manufacturing by neural networks based approach. *J Intell Manuf* 11:311–331
- El Baradie MA (1997) A fuzzy logic model for machining data selection. *Int J Mach Tools Manuf* 37:353–1372
- Zhao Z (1995) Process planning with multi-level fuzzy decision-making. *CIM Syst* 8:245–254
- Patil L, Pande SS (2002) An intelligent feature-based process planning system for prismatic parts. *Int J Prod Res* 40:4431–4447
- Khoshnevis B, Sormaz DN, Park JY (1999) An integrated process planning system using feature reasoning and space search-based optimization. *IIE Trans* 31:597–616
- Tseng Y-J (1998) Feature-based fixturing analysis for machining parts represented with multiple sets of features. *Int J Prod Res* 36:2743–2770
- Joo J, Cho H (1997) Efficient feature-based process planning for sculptured pocket machining. *Comput Ind Eng* 33:493–496
- Amaitik SM, Kilic SE (2005) STEP-based feature modeler for computer-aided process planning. *Int J Prod Res* 43:3087–3101
- Amaitik SM, Kilic SE (2004) STEP feature-based intelligent process planning system for prismatic parts. In: *Proceedings of UMTIK 2004, Antalya, Turkey, 13–15 October*, pp 243–252

15. International Organization for Standardization (1992) ISO 10303 part1- overview and fundamentals principles. International Organization for Standardization, Geneva
16. International Organization for Standardization (1997) ISO 10303 part 11 - descriptive methods: the EXPRESS language reference manual. International Organization for Standardization, Geneva
17. International Organization for Standardization (2000) ISO 10303-224 industrial automation systems and integration-product data representation and exchange - application protocol: mechanical product definition for process planning using machining features. International Organization for Standardization, Geneva
18. Wong TN, Wong KW (1995) A feature-based design system for computer-aided process planning. *J Mat Proc Tech* 52:122–132
19. International Organization for Standardization (2000) ISO/DIS 14649-111 Industrial automation systems and integration - physical device control - data model for computerized numerical controllers - tools for milling. International Organization for Standardization, Geneva
20. Tsai Y, Chen JC, Lou S-J (1999) An in-process surface recognition system based on neural networks in end milling cutting operations. *Int J Mach Tools Manuf* 39:583–605
21. Tsoukalas LH, Uhrig RE (1997) Fuzzy and neural approaches in engineering. Wiley, New York
22. Prasad AV, Rao PN, Rao UR (1997) Optimal selection of process parameters for turning operations in a CAPP system. *Int J Prod Res* 35:1495–1522
23. Liu Y, Wang C (1999) A modified genetic algorithm based optimization of milling parameters. *Adv Manuf Tech* 15:296–799
24. Wong SV, Hamouda AMS, El Baradie MA (1999) Generalized fuzzy model for metal cutting data selection. *J Mat Proc Tech* 89–90:310–317
25. Amaitik SM, Kiliç SE (2002) Fuzzy logic model for selection of machining parameters in CAPP. In: *Annals of DAAAM*, Vienna, Austria, 23–26 October, pp 11–12
26. International Organization for Standardization (2000) ISO/DIS 14649-10 industrial automation systems and integration - physical device control - data model for computerized numerical controllers - general process data. International Organization for Standardization, Geneva
27. Suh S-H, Cho J-H, Hong, H-D (2002) On the architecture of intelligent STEP-compliant CNC. *Int J Comput Integ Manuf* 15:168–177
28. International Organization for Standardization (2000) ISO/DIS 14649-11 industrial automation systems and integration - physical device control - data model for computerized numerical controllers - process data for milling. International Organization for Standardization, Geneva
29. Newman ST, Allen, RD, Rosso Jr RSU (2002) CAD/CAM Solutions for STEP compliant CNC manufacture. In: *Proceedings of the 1st CIRP seminar on digital enterprise technology*, Loughborough, Leicestershire, UK, pp 123–128
30. Clark JE (2002) VBA for AutoCAD 2002. Prentice-Hall, New York
31. Usher JM (1996) A STEP-based object-oriented product model for process planning. *Comput Ind Eng* 31:185–188
32. Luo JLH et al (1997) Object-oriented modeling for CAPP open architecture. *Int J Comput Integ Manuf* 10:74–82