Pakistan Journal of Information and Technology 1(3): 294-298, 2002

A Binary ART Neural Network Methodology for Computer-Aided Process Planning of Milling Parameters

Amjed Al-Ghanim
Department of Industrial Engineering, An-Najah National University
Nablus (P.O. Box 1712) – Palestine

Abstract: Artificial neural networks have been successfully employed for providing efficient solutions for decision making problems and gained increased significance for their use in computer integrated manufacturing environments as effective tools for improving productivity and decision quality. The function of process planning in machining operations is a prominent one for neural network applications since it has direct impact on overall manufacturing productivity. This paper presents analysis and results of applying self-organizing neural networks to the selection of machining parameters of milling processes. The importance of this approach stems from the ability of neural nets to handle vague or ill-structured problems and the inherent capability of generalizing solutions to unseen problems. Furthermore, self-organizing neural networks do not require full knowledge of 'output' data needed during the training phase; only a small portion of the data is needed for model calibration. Simulations using ART1 neural model were applied to the selection of tool material type and tool entry strategy, and the results demonstrated a high potential for the development of neural network modules for practical applications.

Key Words: Artificial Neural Network, Machining Operation, Machining Parameters, Milling Process

Introduction

Research Background: Process planning is a function that maps design features to manufacturing features (Bedworth *et al.*, 1991). It represents the link between engineering design and shop floor manufacturing and determines the manufacturing operations required for transforming a part from a rough state to a finished state specified by the engineering drawing. As related to machining, process planning consists of a series of tasks to interpret the product model including selection of machine tools, tool sets, setups, and machining sequences (Cox *et al.*, 1993). As such, process planning is an involved activity that has a large number of variables describing parts to be produced and the production resources.

There are two systematic computer-aided process planning strategies; namely variant planning and generative planning. The distinguishing feature of the variant planning strategy is that former plans are retrieved and modified for new parts, while generative planning is a strategy that strives to create a new plan for a part, from scratch, based on analyzing part Geometry and other related specifications (Bedworth *et al.*, 1991). Group Technology and Expert Systems technology have been employed to assist in process planning where these methodologies are based on algorithms and machinists' expertise to generate process plans (Kusiak, 1991; Gupta, 1988; 1990 and Alting and Zhang, 1989). Various methodologies have been employed to implement generative CAPP expert systems including among others, case-based reasoning (Cox *et al.*, 1993), feature-based recognition (Tsang and Brissuad, 1989 and Houten *et al.*, 1989), and solid modeling techniques (Descote, Latombe, 1983 and Jared 1983).

Virtually all products require a chip producing machine tool in some stage of manufacture and today much of the chip producing is being accomplished through Computer Numerical Control (CNC). In most cases during the planning for the CNC machining operations, it is necessary to consult an expert. Whether that expert is a machinist, part programmer or a knowledge base, the specific set-up and run parameters for machining processes must be determined. To improve the productivity of machining process planning, focus is made on the ability of pattern recognition systems such as neural model to identify manufacturing features that needs to be machined.

Most research efforts have focused on utilizing expert systems technology to implement process planning systems. These systems offer assistance based on capabilities ranging from fast retrieval of existing plans to generating plans based on interpreting product model Geometry. Expert systems have shown success in automating certain aspects of process planning. However, with the increasing level of complexity of manufacturing environments, and the increasing demand for achieving higher level of integration, expert systems methodology suffers some limitations. First, when domain knowledge is complex and intuitive as in the case of process planning, the knowledge acquisition phase of system development becomes a real bottleneck because it is not always possible to extract and represent knowledge in an explicit form (Cox *et al.*, 1993; Alting and Zhang, 1989 and Giarratano and Riley, 1989). Second, when the system rule base gets larger, the knowledge validation process becomes difficult and prone to errors (Al-Ghanim and Talhouni, 1994). Finally, a dynamic manufacturing environment requires utilization of fast adaptive systems. However, expert systems, due to explicit knowledge representation, show limited adaptive capabilities, and they do not tolerate missing or inaccurate data.

To compensate for some of these limitations and enhance the expert systems approach, some research efforts have focused on the use of neural network methodologies for process planning.

Table 1: Input and Output Decision Variables and Related Values for Milling Operations

Input Space Variables	Selected Values of Input Variables				Output Space Variable	Selected Values of Output Variable
Machined Feature Type	Slot	Pocket	Notch	Step	Tool Material Type	-High speed steel -Carbide -Ceramic
Depth of Cut	Very deep	Deep	Medium depth	Shallow		
Part Material Hardness	Very hard	Hard	Medium hard	Not hard		
Part Machinability	Very high	High	Medium	Low	Tool Entry Strategy	-Free entry -Drill a hole -Ramp -Plunge
Fixture Rigidity	Very high	High	Medium	Low		

Table 2: Encoding Scheme of the Input Decision Variables

Input Variable	Values and Codes of Inputs Variables			
Machined Feature Type	Slot 00	Pocket 01	Notch 10	Step 11
Depth of Cut	Very deep 00	Deep 01	Medium deep 10	Shallow 11
Part Material Hardness	Very hard 00	Hard 01	Medium hard 10	Not hard 11
Part Material Machinability	Very high 00	High 01	Medium 10	Low 11
Fixture Rigidity	Very high 00	High 01	Medium 10	Low 11

Table 3: Performance Measurements of the Developed System: Tool Material Type Model

Vigilance factor (rho)	SOC Number of presented cases or Patterns	Rate of identification (%)
0.50	119	83
0.60	141	78
0.65	138	88
0.70	170	91
0.80	200 (terminated at 200)	82

Table 4: Performance Measurements of the Developed System: Cutting Strategy Model

Vigilance factor (rho)	SOC Number of presented cases or patterns	Rate of identification (%)
0.50	139	87
0.60	131	80
0.65	158	88
0.70	160	96
0.80	190	89

For example, Cox *et al.*, (1993) used a multilayer back propagation network for process planning of milling operations (Cox *et al.*, 1995), while Knapp and Wang applied neural networks for the automatic acquisition of process planning knowledge (Knapp and Wang, 1992). Chen proposed an unsupervised neural network methodology for setup generation and feature sequencing (Chen, 1993).

This paper presents methodology and results of applying a Self-organizing neural model to the process planning of milling operations. First, process planning is cast as a pattern recognition problem, and viewed as a mapping function between process features and machining strategies. Second, Adaptive Resonance Theory neural networks (ART) are developed for automating an important aspect of process planning decisions, namely, cutting tool material.

Casting Process Planning as a Pattern Recognition Problem: Pattern Recognition (PR) can be characterized as an information mapping process taking place over a set of metric or non-metric spaces (Shalkof, 1992). A relationship stands as a mapping between class-membership space and a pattern space. Each class corresponds to a subset of patterns in the pattern space where these pattern spaces may overlap allowing patterns from different classes to share the same attributes. In turn, the pattern space is mapped, via another relationship, into to an observation space with feature patterns. In simple terms, the class-membership space, also called the output space, represents the solution set to the problem at hand, while the pattern space represents the input variables or input data that are needed to solve the problem.

Process planning in general can be cast as a pattern recognition problem; that is a set of mappings from one space to another space. First, process planning for machining is a multivariable input/output process including among others part dimensions, material, surface finish, fixture rigidity, tool material type, spindle speed, feed rate, depth of cut, and tool entry strategy. Therefore, to facilitate analysis one should precisely identify the output decision quantities and corresponding input decision quantities. It should be noted that the basic idea of utilizing neural networks as pattern recognizers is that the output decision variables should be chosen such that their values can be grouped into a finite number of classes. Thus, as related to milling operations, the output decision variables can be, for instance, cutting tool material type and strategy for entry of the tool into part material. Such variables represent decisions that the machinist will have to make at one stage during the planning process.

To demonstrate the modeling process of process planning for milling operations as a pattern recognition problem, focus is made on two encountered output decision variables, which are the cutting tool material type and tool entry strategy. These decision variables are confined to most commonly used values in the output decision space representing tool material types and cutting strategies for milling operations. For the tool material type decision variable, these values are high-speed steel, carbide, or ceramic, while the values of the tool entry strategy decision variable are free entry, drill a hole, ramp, or plunge. In turn, the output variable depends on a number of factors that form the input decision variable or decision space. These variables include machined part features, depth of cut, part material hardness, part material machinability, and fixture rigidity. It is important to note that each of these input variables may assume a very large number of numerical values (theoretically infinity), but to use the language of machinist in expressing their experience, each variable is confined to have only four values.

Based on this selection of input and output decision variables and related possible values, the formulation of milling process planning problem is presented in Table 1 as a pattern recognition problem where the input pattern is represented by input space variables and output pattern is represented by output space variables. The values of these spaces represent the input data set and the solution set respectively.

The research methodology adopted here is aimed at exploring the utilization of ART neural network models as supportive decision-making tools in machining process planning. To carry out this objective, a binary ART neural network model has been used as a pattern recognizer, and implemented and tested to make decisions on tool type material and tool entry strategies. Since binary ART models require binary input data only, the input decision variables need transformation into binary format. The encoding of the input variables into binary numbers according to ART1 neural model is illustrated in section "Input Data and Training Parameters".

Adaptive Resonance Theory Networks: In general terms, a neural network is parallel-distributed information processing system that consists of a huge number of simple interconnected elements called neurons. Each neuron carries the same signal at its single output, which branches into many collateral inputs. Processing takes place within each element using values of the input signals coming through the impinging connections, and also using locally stored values (Zurada, 1992). A neural network is composed of layers of neurons; an input layer that receives data from the environment, an output layer that sends the information (i.e., processed data) to outside world for decision making. In case hidden layers do exist, they lie between the input and output layers and has direct connections with these two layers. Several neural network architectures have been developed and used in various applications including Back propagation, Hopefield nets, Kohonen models, Adaptive Resonance Theory nets, etc., (Zurada, 1992). Adaptive Resonance Theory Networks, developed by Grossberg and Carpenter (Grossberg, 1976 and Grossberg and Carpenter, 1987) represent a form of competitive or unsupervised learning network methodologies.

Competitive learning is an unsupervised learning strategy, which accomplishes a clustering task, based on optimizing a given function (e.g., distance). Data representing environmental patterns, that are to be grouped or clustered, is received at the bottom layer. The received data is then transmitted to the top-layer nodes via the bottom-up connecting links. If the input vector is denoted by x and the weights on the connecting links impinging onto an output node j are represented by vector wj, then the input at node j is computed as the inner product of x and wj (Pao, 1989) and given by $x^t.w_j$.

In a competitive learning network, the output node with the largest product wins the competition and the input pattern x is claimed a member of the winning-node group. The output of the winner assumes a value of 1, while all other losing nodes have 0's as their outputs. The learning process ensures that the weight vector of the winning node be closer to the input pattern x.

The concept of Adaptive Resonance Theory (ART) is based on competitive learning; however, ART networks have the ability to maintain plasticity and remain stable. The stability/plasticity problem has been solved in ART networks by establishing feedback connection links between the top-layer and the bottom-layer nodes. This set of links carries top-down weights, which are used to confirm whether a given pattern truly belongs to the winning node group. Therefore, the first stage of an ART operation is identical to what CL networks would accomplish; however, at this point the two networks diverge. While the CL network declares the node with the maximum input (max. $x^t.w_j$) to be the winner, the ART network makes a reservation; the winner must pass another test to be declared as the winner. This is called the vigilance test, aimed at verifying that the input pattern 'closely resembles' the winner group. The degree of 'close resemblance' is governed by the vigilance factor as explained in the details of the ART learning algorithm. The learning process proceeds by exposing the neural model to a large number of learning examples that will, through the training algorithm, establish weights of the internal connections of the system. Once the training process is completed (i.e., when no more clusters are created), the weights are frozen and the system can then be used for decision making as new cases are presented to the system.

ART1 Training Algorithm: The basic structure of the following learning algorithm is based on a modified version of the training algorithm developed by (Al-Ghanim, 1997) which in turn is based on the Adaptive Resonance Theory networks for binary inputs (ART1) as presented by (Pao, 1989). Therefore, a preprocessing step is necessary to transform the analog process output into a binary-coded form. A binary encoding procedure based on a 10-digit code is employed (section "Encoding of Input Data"). In this encoding scheme, each of the above 5 input variables is represented by 2 binary digits that can encode up to 4 values.

Training Algorithm

Input: 1. A set of training examples (x^l, l=1,...L). It is represented by a matrix of dimension L by k, where L is the number of training examples and k is the pattern length (i.e., k=10).

2. The vigilance factor, rho, the maximum number of allowed clusters, H.

Output: Unlabelled cluster(s) representing various classes of the tool material type.

Steps:
1. Perform encoding to convert training examples into binary format.
2. Start with no cluster centers formed (i.e., all weights are equally initialized, the bottom-up, b_{ji} = 1/(L+1), and the top-down, t_{ji} = 1, for all i's and j's).
3. Perform iterations until no more clusters can be formed. At this point, stop since stability (i.e., convergence) has been achieved.
4. Present a new training example x (if any) to input nodes where $x_i = \{1,0\}$.
5. Use bottom-up processing to obtain the activation value for each output node j (j=1,....H): y_j = Sum $b_{ji}x_i$.
6. Select the output node with the largest y_j value, that is, the winning node representing the closest cluster J.
7. Verify that x truly belongs to the Jth cluster by performing top-down processing; that is, form the weighted sum $t_{Ji}x_i$.
8. Test for vigilance to verify if x belongs to the Jth cluster: (Sum $t_{Ji}x_i$)/x> rho. If the condition is satisfied, proceed to step 9. Otherwise, go to step 10.
9. Update the weights b_{Ji} and t_{Ji} for that specific node J and all i, at the current iteration, *l*+1, using the update rules:

$$t_{ji}(l+1) = t_{ji}(l)\,x_i$$

$$b_{ji}(l+1) = \frac{t_{ji}(l)\,x_i}{0.5 + \sum_{i=1}^{k} t_{ij}\,x_i}$$

10. Since x does not belong to the node that was the most likely, deactivate that node and go to step 6 to start the next winning node. If all nodes are considered and no assignment is made, then form a new cluster with vector x as the cluster center.

Calibration and Classification Algorithm: The network trained using the above algorithm can be used in real operating mode (i.e., classification). Before the system can be utilized in real operation, it requires calibration. This process identifies which nodes of the network model correspond to the different classes of the tool material type and cutting strategy.

Algorithm for Classification of Tool Material Types or Cutting Strategy:

Input: Examples representing input decision variables and data as given by input vector structure whose outputs are known (i.e., corresponding tool material type or cutting strategy is known).

Output: Class codes of the corresponding tool material type or cutting strategy.

Calibration:
1. Supply the trained neural network model with a set of few binary-coded patterns representing tool materials types and another set representing tool entry strategies. The number of such examples required at this step is found to be about 17% of the number of examples used for training, that is (35-40) examples.
2. Determine the set of nodes J that correspond to the clusters of the classes of the output decision variables.

Classification:
1. Supply the network model with unknown binary-encoded example representing a part to be machined.
2. Determine the node j, if it exists, that corresponds to each presented example.
3 If a node j exits and belongs to classes of tool material type or tool entry strategy, then a decision can be made about related output decision, otherwise, no decision can be made.

Materials and Methods

Input Data and Training Parameters: The simulation methodology aims at implementing the above training/testing policy which provides a classification capability of new cases (i.e., new parts to be machined). A set of 2 hundred training examples is used; that is, the training period is forced to terminate when a maximum of 2 hundred examples are presented. Training and testing of the network take place using various values for the vigilance factor. The vigilance factor controls the required degree of resemblance (or similarity) among input examples. Therefore, it controls the number of possible clusters formed for a particular class; the higher the vigilance factor, the higher the number of clusters. Values for this parameter are selected in the range (0.5 - 0.8). A simulation run consists of three phases:
1. Training the network until it converges (e.g., forming a stable set of clusters to represent the three classes pertaining to tool material type or four classes of tool entry strategies),
2. Testing the network using various sample examples,
3. Evaluating the performance measures.

Two hundred cases are used for training and another one hundred cases for testing in each simulation run. Each simulation run is performed at a specific vigilance level, that is, training and testing are performed using the same vigilance factor. Since two machining decisions are considered in this research, two ART1 network models have been built, one for tool material type and one for tool entry strategies.

Encoding of Input Data: The input vector contains 5 features where each feature is encoded into 2 binary digits, thus, making the length of the input vector 10 binary digits. This encoding scheme will allow representation the four values of each input feature given in Table 1. This scheme is summarized in Table 2 below. To illustrate this procedure, consider a machining part with the following features: Feature Type = Slot, Depth of Cut = Medium deep, Material Hardness = Hard, Part Machinability = Very high, and Fixture Rigidity = Low. The code for this example will be (0001010011). It is important to note that the order of input variables in the input vector is as given in Table 2.

This binary form allows the extraction of the relevant structural information needed to characterize output decision classes (Al-Ghanim, 1997). Convergence is achieved when no new clusters are constructed to account for new input examples. Alternatively, if convergence is not likely to be realized especially when the vigilance factor is close to 1, the learning process is forced to terminate.

Performance Measures: To evaluate the performance of the system based on the proposed training policy, a set of performance measures is defined. Rate of Identification (ROI) and the Speed of Convergence (SOC) were used as given in (Al-Ghanim, 1997).

The Rate of Identification: The Rate of Identification (ROI) characterizes the ability of the system to correctly identify new input examples. It is the percent of correctly classified patterns when the system is exposed to a certain number of new examples. This measure gives the classification accuracy of the system.

Speed of Convergence: Speed of Convergence (SOC) describes the speed of convergence at which a stable set of clusters is established. This measure is very informative, as it is important, in a real application, to know how fast the system would learn the environmental process data and be reliably ready for on-line process monitoring. This measure can be computed as the number of input patterns needed to form a stable set of clusters.

Results and Discussion

Once the self-organizing neural network has converged, the training period is terminated. To test each trained model, 100 input patterns representing various cases of machined parts are used. Results of the network performance are summarized in Tables 3 and 4. As seen from the results, the Speed of Convergence, SOC, decreases as the vigilance factor increases as shown by the number of required input patterns needed for convergence. Under this training and testing policy, the ROI has shown an increasing trend as the vigilance factor is increased.

As mentioned earlier, the objective of utilizing unsupervised learning policy is to allow the system capture machining expertise and use it in decision making throughout process planning of machined parts. The utilization of the system is mainly to classify new machining cases into one of the output classes given by the three types of tool material type or the four types of tool entry strategies. As evidenced by the results, this approach can be applied to automate the process planning in a real-life setup as part of an integrated CAM system. The average convergence speed and the rates of identification obtained are within acceptable limits when compared with results in the literature for different types of methodologies.

Conclusion

This paper has presented a methodology and analysis results of applying self-organizing neural nets for automating process planning of machining operations. The methodology lends itself for automation within an integrated CAM system, and the results demonstrated that system accuracy is comparable with similar system using supervised learning techniques (Cox *et al.*, 1993; Knapp and Wang, 1992). On advantage of this system is that does not require full input data with respect to the output decision variables. Further research should make use of this methodology for other important process planning decisions such as cutting tool size and number of flutes of cutter, and cutting strategies. Several other encoding schemes can also be experimented based on more than 2-digit code.

References

Alting, L. and H. Zhang; 1989. 'Computer-Aided Process Planning: state-of-the-ART survey', International J. of Production Research, Vol. 27, No. 1-4,(553-585).

Al-Ghanim, A. and B. Talhouni; 1994. 'A Diagnostic Expert Sys. for IBM PCs and Compatibles', Muta J. for Scientific Research, Vol. 9, 3, Muata Uni., Jordan.

Al-Ghanim, A., 1997. "Unsupervised Learning Neural Algorithm for Identifying Process Behavior on Control Charts and a Comparison with Supervised Learning Approaches", Computers and Industrial Eng., Vol. 32, No. 3,

Bedworth, D., R. Henderon and M. Wolfe; 1991. Computer-Integrated Design and Manufacturing', McGraw-Hill.

Chen, C. L . P., 1993. "Setup Generation and Feature Sequencing using Unsupervised Learning Algorithm", in Proceedings of the NSF Design and Manufacturing Systems Conference, Charlotte, California,

Cox, L., D. Culler and A. Al-Ghanim; 1993. 'Artificial Intelligence Application Utilization in Computer Numerical Control Manufacturing', in Proc. of IIE Conference, Los Angeles, 490-496.

Cox, L., D. Culler and A. Al-Ghanim; 1995. "Machining Strategies Collected, Stored, and Utilized with Artificial Neural Networks", Proceeding of ANNIE'95 International Conference, 967-972, St. Louis, Missouri.

Descote, Y. and J.C. Latombe; 1983. 'GARI: An Expert Sys. for Process Planning, Solid Modeling Computers', Plenum Press, New York.

Gupta, T., 1988. 'A State of the Art Survey of Expert Systems in Process Planning', IIE Integrated Sys. Conference in St. Louis, 269-274.

Gupta, T., 1990. 'A Survey of Expert Systems in Manufacturing and Process Planning: Current Dev. and it Future', Computers and Industrial Eng., 18:1: 69-80.

Giarratano, J. and G. Riley; 1989. 'Experts Systems: principle and programming', PWS publishing company, Boston.

Grossberg, S. and G. Carpenter; 1987. "ART2: Self-organizing of Stable Category Recognition Codess for Analog Input Patterns", App. Optics, 26:4919-46.

Grossberg, S., 1976. "Adaptive Pattern Classification and Universal Coding: I. Parallel Dev. and Coding of Neural Feature Detectors", Biological Cybernetics, 23:121-134.

Jared, G., 1983. 'Shape Features in Geometric Modeling, Solid Modeling by Computers', Plenum Press, New York.

Knapp, G. M. and H.P. Wang, 1992. "Acquiring, storing and utilizing process planning knowledge using neural networks", J. of Intelligent Sys. in Manufacturing, 30: 4.

Kusiak, A., 1991. 'Group Technology for Flexible Manufacturing Systems', Handbook of Flexible Manufacturing Systems, Academic Press, San Diego, CA.

Pao, Y. H., 1989. 'Adaptive Pattern Recognition and Neural Networks', Addison-Wesley, Mass.

Shalkof, R. J., 1992. 'Pattern Recognition: Statistical, Structural, and Neural Approaches', John Wiley and Sons, Inc., USA.

Tsang, J. P. and D. Brissuad; 1989. 'A Feature-Based Approach to Process Planning', Computers in Industrial Eng.

Van Houten; van's Erve and Kals; 1989. 'A feature Based Computer Aided Process Planning', Laboratorium voor Produktietechniek, Univ. Twente, Netherlands, in Proc. of the 21st CIRP International Seminar on Manufacturing Systems, Stockholm.

Zurada, J. M., 1992. 'Introduction to Artificial Neural Systems', West Publishing Company, USA.