AUTOMATED TOOL CONDITION MONITORING IN MACHINING USING FUZZY NEURAL NETWORKS

by

SHENGMU LI, B. ENG., M. ENG.

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree

Doctor of Philosophy

McMaster University

TOOL CONDITION MONITORING USING FUZZY NEURAL NETWORKS

DOCTOR OF PHILOSOPHY (1996) McMASTER UNIVERSITY

(Mechanical Engineering) Hamilton, Ontario

TITLE: **Automated Tool Condition Monitoring in Machining Using Fuzzy Neural Networks**

AUTHOR: Shengmu Li, B. Eng. (Northeastern Institute of Heavy Machinery)

M. Eng. (Beijing University of Science and Technology)

M. Eng. (McMaster University)

SUPERVISOR: Professor M. A. Elbestawi

NUMBER OF PAGES: xvi, 187.

To Dear MOM, DAD, CHUNHUA and MARK

ABSTRACT

A new approach for automated tool condition monitoring in machining by using fuzzy neural networks is proposed. The *Multiple Principal Component* (MPC) fuzzy neural networks are built based on three major components of *soft computation*, namely fuzzy logic, neural networks, and probability reasoning.

The system architecture is a partially connected neural network with fuzzy classification at neurons and fuzzy membership grades for interconnections. Principal component analyses in multiple directions are implemented for the feature extraction and the "maximum partition". The partitions of the learning samples are based on the distributions of the monitoring indices in the principal component directions. A fuzzy membership function is used to measure uncertainties in the sampled data and to form "soft boundaries" between the classes. A processing element in the network is connected to others through the fuzzy membership grades and other information available. The partial connections make short training times and short routines in classifications.

Three major issues in developing the MPC fuzzy neural networks are *supervised classification*, *unsupervised classification* and *knowledge updating*. The system obtains the knowledge about classifications by learning. The learning samples are obtained from cutting tests performed through a reasonable range of cutting conditions.

Several sensors are used for monitoring feature extraction. The signals from different

types of sensors at different locations insure that the most significant information about the changes in tool conditions is collected. Metal cutting mechanics are first considered for the sensor selection and the sensor allocation. The measured signals are further analyzed and the monitoring features are extracted. These indices are the inputs for the fuzzy neural networks. The tool conditions considered include sharp tool, tool breakage, and a few selected stages of tool wear. The experimental results in turning and drilling have shown good performance of the proposed monitoring system in these tests.

ACKNOWLEDGEMENTS

I would like to express my sincere and heartily appreciations to my supervisor, Dr. M. A. Elbestawi, for his support, encouragement, and guidance throughout the course of this study. Special thanks are extended to Dr. R. L. Judd, Dr. M. P. Sklad and Dr. P. A. Taylor, the members of my PhD supervisory committee, for their continuous interests and encouragements.

Thanks are also given to fellow colleagues and graduate students in the IMMRC and the Department of Mechanical Engineering, with whom I have shared pleasant times and experiences.

The financial support given in the forms of Graduate Scholarship, Teaching Assistantship, and Research Assistantship by the Department of Mechanical Engineering and the NSERC is gratefully acknowledged.

I will always be beholden to my parents, my wife and son for their understanding, support, and patience during my study. The love and care from them, other family members, and friends are remembered forever.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Automated tool condition monitoring is of major importance to manufacturing automation. The need for continuous improvements in product quality, reliability, and manufacturing efficiency has imposed strict demands on automated product measurement and evaluation. Manufactured products of the modern day demand higher precision and accuracy than before. Automated process monitoring becomes very crucial to successfully maintaining high quality products with low cost.

Automated tool condition monitoring implies identifying the machining process and tool conditions without interrupting the manufacturing process operation. Monitoring is performed under minimum human operators' supervision. Clearly, the development of these systems is necessary to prevent machine tool damages as well as improve machine utilization.

An "Intelligent Sensor System" was defined by Dornfeld (1986) as an integrated system consisting of sensing elements, signal conditioning devices, signal processing algorithms, and signal interpretation and decision making procedures. This system acts not only as a signal collecting device, but also as a sorting and analyzing machine. In the absence of a human operator, the system functions to sense possible signals indicating the process status and its changes, to interpret incoming sensed information, and to decide on the appropriate control action. Such a sensor system can be enhanced in dealing with sophisticated information by obtaining the abilities of self-learning, knowledge updating, and error correction. We define such a system as *Automated/Intelligent Monitoring System.* The system possesses abilities of sensing, analyzing, and knowledge learning, which are essential to machining tool condition monitoring.

When human operators monitor machining processes, they use all possible sensory information, which may be seeing (*e.g.* observation of chip colour, surface finish of the workpiece), hearing (*e.g.* sound generated by rubbing between the tool and the workpiece), and smelling (*e.g.* smell of smoke generated in machining). Associating the sensory cues with machining process and tool conditions depends to a great extent on the knowledge and experiences of the operators'. An automated/intelligent machining process and tool condition monitoring system should be able to emulate as closely as possible the sensing, recognizing, responding, and learning abilities of human operators. To emulate the human monitoring action, an automated tool condition monitoring system has four components: (1) *Sensing Technique*; Typically, indirect sensing techniques such as cutting forces, vibrations, and acoustic emission are used. In many cases, signals from a single sensor may not be sufficient and other information may also be necessary to result in a correct decision. This technique involves using multiple sensors. The sensory data from different locations and different types of sensors are combined to give the maximum useful information. (2) *Feature Extraction*; Ideally, sensory signals contain the necessary information required to discriminate between different process and tool conditions. However, the sensed signals are usually noisy and have to be further processed in order to yield useful features which are highly sensitive to the tool conditions but insensitive to noises. This process is called feature extraction. (3) *Decision Making*; Decision making strategies process the incoming signal features and perform a pattern association task. That is mapping the signal feature to a proper class (tool condition). The processing can be done sequentially or in parallel depending on the architecture of the monitoring system. And, (4) *Knowledge Learning*. In order to make a correct decision, learning algorithms have to be provided. Such algorithms tune system parameters by observing the sample features corresponding to different tool conditions. Like human operators, automated monitoring systems should have the ability to learn from their experiences (past work) as well as the new information from the machining process.

This dissertation presents a novel approach to the development of an automated tool condition monitoring system — *the Multiple Principle Component (MPC) Fuzzy Neural Networks* for automated tool condition monitoring in machining. The thesis consists of six chapters. The main concepts of these chapters are summarized as follows:

Chapter I gives a brief introduction to the objective and components of this thesis.

In *Chapter II*, a general literature survey is provided, which includes explanations of tool condition monitoring tasks, sensor fusion, modelling approaches, expert systems, neural networks, and fuzzy classification. Various approaches to machining process and tool condition monitoring are introduced. The main objective and research issues are presented.

In *Chapter III*, decision making strategies for tool condition monitoring in machining are discussed in detail. Pattern recognition, neural networks, and fuzzy classification are analyzed in separate sections. Each section shows respectively the principles and concepts, as well as some applications, of these strategies.

Chapter IV presents the proposed new strategy for automated tool condition monitoring in machining. It is called as "*the Multiple Principal Component (MPC) Fuzzy Neural Network.*" First, the architecture of the MPC fuzzy neural network is presented. Three major issues of the monitoring system are: supervised learning and classification, unsupervised learning, and knowledge updating. The detailed algorithms for these functions are discussed. The advantages and developments of the MPC fuzzy neural network for automated tool condition monitoring are examined in this chapter.

To verify the performance of proposed the MPC fuzzy neural networks for automated tool condition monitoring, cutting experiments were performed in turning and drilling. The experimental tests were conducted under a range of different cutting conditions. Several sensors were used for collecting signals from cutting forces, vibrations, torque, and spindle motor current. The experimental analyses focus on both supervised and unsupervised learning, as well as the knowledge updating for the monitoring system. The experimental results are demonstrated in *Chapter V*.

Finally, *Chapter VI* presents discussions on all finished research work and the conclusions of this dissertation. Some suggestions for future work are also given.

CHAPTER II

LITERATURE REVIEW

2. 1 INTRODUCTION

Research work for machining process and tool condition monitoring has been one of the most important issues in manufacturing automation. The major goals are to develop self-adjusting and integrated systems that are capable of monitoring in various working conditions with minimum supervision and assistance from operators. The monitoring systems aim at improving quality of the products as well as reducing the costs.

Many monitoring methods have been studied for automated tool condition monitoring in machining. They include modelling approaches, expert systems and statistical pattern recognition methods *etc*. Neural networks and fuzzy logic are also studied for machining tool condition monitoring.

There are a few major functions in an automated tool condition monitoring system: signal acquisition, signal processing and feature extraction, knowledge learning and decision making. Among them, signal processing/feature extraction and decision making are considered as an "integrated entity" and called "monitoring methods" (Du *et al*, 1995). Moreover, knowledge learning is also highlighted in recent research work.

This chapter gives a comprehensive review of the most recent research work on the development of machining process and tool condition monitoring. Several sections introduce separately the monitoring tasks and monitoring methods by using sensor fusion, modelling approaches, expert systems, neural networks, and fuzzy classifications.

2.2 TOOL CONDITION MONITORING IN MACHINING

The development of reliable and effective machine tool condition monitoring techniques is crucial for the realization of unmanned or partially manned machining. Application of automated tool condition monitoring enables modern manufacturing equipment to work free of errors and guarantees high quality of the products. Significant research work has been performed in this research field from various points of views including analytical forecast, dynamic structure identification, monitoring techniques, and adaptive control approaches. Among those comprehensive surveys on this subject, Tönshoff and Wulfsberg (1988), Tlusty and Andrews (1983), Isermann (1984), and Dornfeld (1990) gave a full description of the development of modern monitoring techniques for machining.

Tönshoff and Wulfsberg (1988) described the vital importance of monitoring machining process in improving the effective machining time of a machine tool, increasing the productivity, detecting the new process phenomena from new materials and new cutting processes, registering the trends in running the machining processes, and diagnosing the reason for a process breakdown. Their paper covers conventional and enhanced methods for monitoring and control of machining process. The review lists the objectives of monitoring machining processes as:

i) to maintain safety,

ii) to prevent fatal damage,

iii) to prevent rejects,

iv) to prevent idling of equipment, and

v) to achieve an optimal use of resources.

In this study, five monitoring subjects are identified: machine, tool, process, tool conditioning, and workpiece. The monitoring functions are also classified into two groups: time critical and non-time critical. The former requires a system response within a range of milliseconds while the later may take seconds or even minutes. The components of a monitoring system are defined as in Figure 2.1. These include: sensors, signal conditioning, models, and strategy.

Sensor techniques deal with the problem of collecting the featured signals about machining processes such as the physical principle of sensors and the technical application of sensors. Multi-sensor systems and intelligent sensors are also applied to monitoring tasks. Critical reviews on the sensors for machining monitoring were given by Tlusty and Andrews

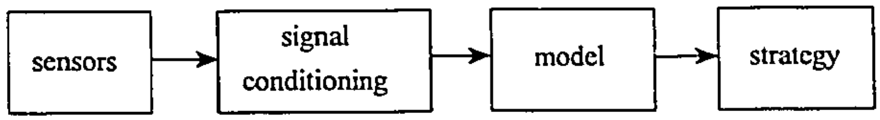


Figure 2.1 Components of Monitoring System / Tönshoff and Wulfsberg, 1988 /

(1983) and Dornfeld (1992). The devices include dimensional and proximity sensors, cutting force dynamometers, spindle (torque and power) sensors, accelerometers, acoustic emission sensors, ultrasonic sensors and the like. Applications of these sensors include geometric corrections, machine diagnosis, surface finish controls, tool condition monitoring, and machining process monitoring.

Signal conditioning and signal processing deal with data condensation in order to extract the monitoring features which are sensitive to the monitoring subjects. The signals are evaluated in both time domain and frequency domain.

Models are required to relate the measured valued to the monitoring and control subjects. Models can either be physically or empirically based. There are fixed models, adapting models, and self-learning models. Multi-model systems are also used.

Monitoring and diagnostic systems measure the conditions of a machine tool or of its process and try to find functional or causal relations between failures and their origins. They are all open loop systems. Strategies are employed to decide how information is acted upon in the process control. Some of the monitoring functions provided on today's machining tool are used mainly for control purposes, even though studies showed that most tool failures in NC machining were due to the problems with mechanical components rather than the controller (Kegg, 1984). It is obvious, however, that the monitoring systems are the most critical link between the machining process and effective controls.

2.3 SENSOR FUSION

In most cases, signals coming from only one sensor are typically insufficient to give enough information for machining and tool condition monitoring. Using several sensors at different locations simultaneously was proposed for data acquisition (Ruokangas *et al*, 1986; McClelland, 1988; Cryssolouris and Domroese, 1988, 1989; Dornfeld, 1990; Cryssolouris *et al*, 1992). Signals from different sources are integrated to provide the maximum information needed for monitoring and control tasks. A schematic diagram of using multiple sensors in monitoring systems is shown in Figure 2.2.

Sensor Fusion generally covers all the issues of linking sensors of different types together in one underlying system architecture (McClelland, 1988). The strategy of integrating the information from a variety of sensors will increase the accuracy and resolve ambiguities in the knowledge about the environment. The most significant advantageous aspect of sensor fusion is its enriched information for feature extraction and decision making strategy. It provides much more "strong" data for the decision making process with low uncertainty which may be created by inherent randomness or noise in the sensor signals. The

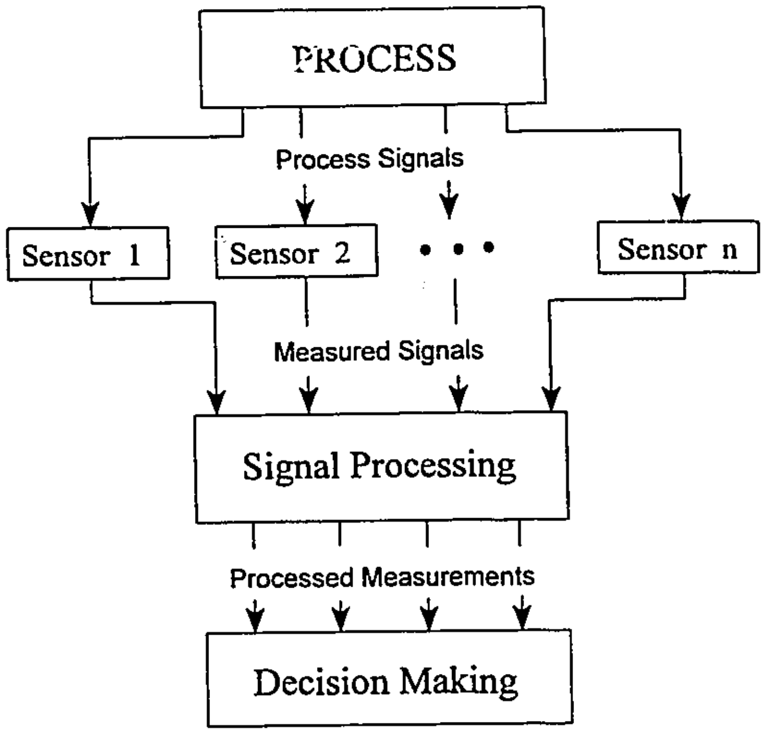


Figure 2.2 Multiple Sensors in Monitoring

features from integrated information are significant within a broader range of cutting conditions. Sensor fusion is able to accommodate changes in the operating characteristics of the individual sensors due to calibration, drift, failure and so on.

Lots of efforts for developing and applying the methodologies for sensor fusion have been reported in machine tool condition monitoring. Linear discriminant functions are the most commonly used (Matsushima and Sata, 1980; Dornfeld and Pan, 1985; Liu and Wu, 1988, 1990; Marks and Elbestawi, 1988). Matsushima's and Sata's objective (1980) was to automatically recognize the cutting state and detect tool failures as part of an intelligent control system. A linear discriminant function was used to integrate machining parameters, such as cutting speed and feed rate, with four features from the power spectrum of the cutting force. Dornfeld and Pan (1985) applied linear discriminant functions to integrate acoustic emission signals with machining parameters (cutting speed and feed rate) for the detection of continuous or discontinuous chip formation during cutting. Liu and Wu (1988, 1990) used a two-category linear classifier to process the sensor signals from both acceleration and thrust force for drill wear detection. Marks and Elbestawi (1988) used a dynamometer, an accelerometer, and the spindle motor power, combined with cutting conditions, for tool condition monitoring using a pattern recognition method. Most of these earlier approaches to sensor integration suffered from the necessity of a time consuming training procedure as well as high sensitivity needed to process conditions, rendering them inefficient for real-time use (Dornfeld, 1990).

Recent work on machining monitoring involved neural networks for integrating the information from multiple sensors (Rangwala and Dornfeld, 1987, 1990; Dornfeld, 1990; Liu and Ko, 1990). Rangwala and Dornfeld (1987, 1990) used a neural network to integrate the signals from acoustic emission and force sensors for flank wear monitoring in turning. The neural network succeeded in filtering out noise in the sensor data and made it possible to applying the strategy over a range of machining conditions. In their work, the measurement vector with 768 dimensions, coming from the force and the AE spectra, was reduced to a feature vector with dimensions of six by maximizing the following separation index:

$$J = trace\,(S_w^{-1}\,S_b)\,, \tag{2.1}$$

where S_w is the within class scatter matrix and S_b is the between class scatter matrix. The six features which were the most sensitive to tool wear were selected using the Sequential Forward Search algorithm (Whitney, 1971). A three-layered neural network was used for sensor fusion and decision making. Liu and Ko (1990) explored on-line classification of drill wear using sensor fusion and artificial neural networks. Acceleration and thrust signals were used as the inputs to the neural networks. The results were compared with those obtained from a linear discriminant approach and it was shown that the artificial neural network had better performance. Most other neural networks for tool condition monitoring also used multi-sensors in data collection.

The integrating information from multiple sensors was also implemented by sensor synthesis. These approaches included multiple least-squares regression and the Group Method of Data Handling (GMDH) (Chryssolouris and Domroese, 1988; Chryssolouris *et al*, 1992). Using multiple least-squares regression, the tool wear and wear rates measured during some initial machining tests are regressed on the corresponding estimates provided by the models. A typically linear regression model is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i + \epsilon\,, \tag{2.2}$$

where y is the synthesized state variable estimate, β_i is a regression coefficient, x_is are state variable estimates provided by process models, and ϵ is a random error. The GMDH is a heuristic method of predicting a dependent variable from a set of independent variables. By laying several sets of quadratic polynomials such that the output of one set is the input to the next set, the method allows the dependent variable to be predicted by a very complex, high-order polynomial of the independent variables. The GMDH algorithm is self-organizing. Variables which are least valuable for prediction of the dependent variable are dropped from further consideration during the construction of each layer.

Sensor fusion was also combined with a fuzzy logic approach for tool condition monitoring in turning (Du *et al*, 1992). In this study, 11 monitoring indices were experimentally and analytically selected from six measured process signals. The sensors included a three-dimensional force transducer, two accelerometers, and a spindle motor power sensor. In another work on the fuzzy logic approach for multi-sensor process monitoring in machining (Li *et al* 1992), sensor signals were fused not only in the signal selection, but also in the feature extraction and decision making. The partial least squares method (which will be discussed in detail later in this thesis) was used for the most significant features in machining tool condition monitoring.

2.4 MODELLING APPROACHES

Modelling approaches have historically been used to discover the root causes of problems with existing equipments and to understand the mechanism of machining processes. There has been a significant amount of modelling approaches for machining tool condition monitoring. Some models considered are: dynamic structure models for vibrations, such as multi-degree of freedom machining system structure response model (Endres *et al*, 1990), and one and two ends supported beam models (Lu and Klamecki, 1990), dynamic structure models for cutting force, such as empirical cutting force model (Endres *et al*, 1990), dynamic models of acoustic emission and feed motor current (Diei and Dornfeld, 1987; Emel and Kannatey-Asibu, 1987, 1988; Stein *et al*, 1986), dynamic models for tool wear such as diffusion wear models, adhesive wear models (Kannatey-Asibu, 1985) and empirical model (Koren *et al*, 1991), linear steady-state models for tool wear and cutting forces (Koren, 1978; Koren *et al*, 1987; Matsumoto *et al*, 1988), and parametric models including AR (Auto-Regressive) for chatter (Yang *et al*, 1982; Tsai *et al*, 1983), AR for tool wear (Liang and Dornfeld, 1989), and AR for tool breakage (Takata and Sata, 1986).

Among the modelling approaches to machining tool condition monitoring, Dynamic Data System (DDS) methodology (Wu, 1977) was broadly applied. This method uses dynamic data in the form of a time (or space) series to develop a physically meaningful, stochastic difference/differential equation, so that the hidden mechanistic features of the system can be extracted from the experimental or operating data. It assumes that a process can be described by an Auto-Regressive Moving Average (ARMA) model:

$$X_t - \phi_1 X_{t-1} - \phi_2 X_{t-2} - \dots - \phi_n X_{t-n} = a_1 - \theta_1 a_{t-1} - \dots - \theta_n a_{i-n-1} , \qquad (2.3)$$

where X_t is time series, a_t is the residuals with normal distributions (white noise), Φ_is are the auto-regressive parameters, and θ_is are the moving average parameters. When using the DDS methodology, a major concern is the parameter estimation of the ARMA model. Although an ARMA model is a linear model, the estimate of the model parameters requires non-linear

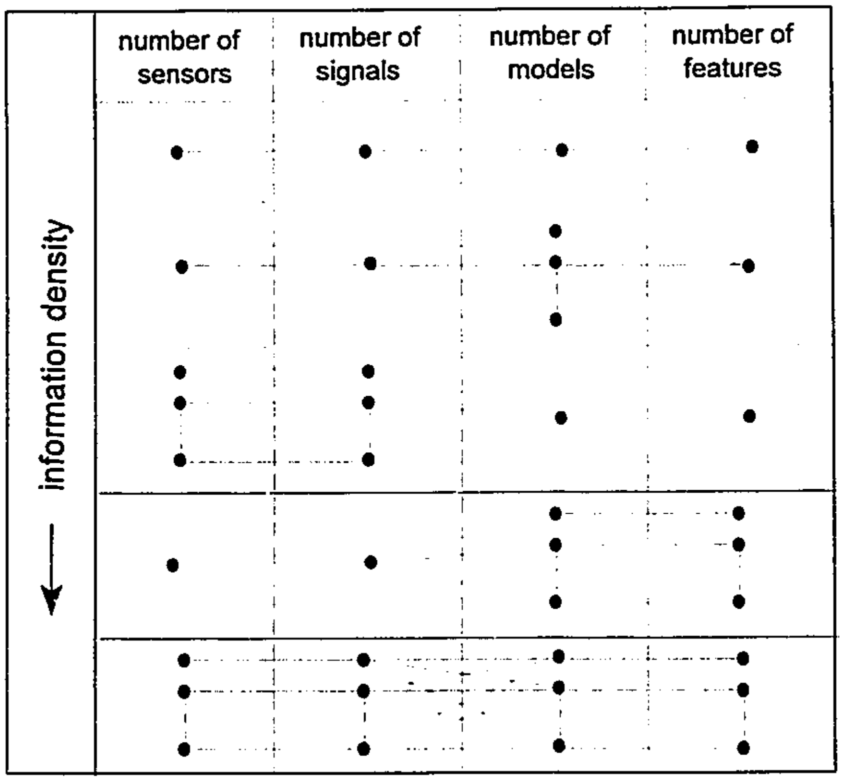


Figure 2.3 Some Possibilities to Combine Sensors and Models / Tönshoff and Wulfsberg, 1988 /

computation for which many algorithms have previously been developed. Some examples of applying time series model in machining monitoring are Wu *et al* (1980), Kim *et al* (1982), Chung *et al* (1993) and the like.

The use of multiple sensors for machining process and tool condition monitoring gives extended information about the process. As most process variables have an influence on one another, highly sophisticated models, or even more than one model, are needed to work up the sensor signals. In the most practical applications, the measured variable from one sensor

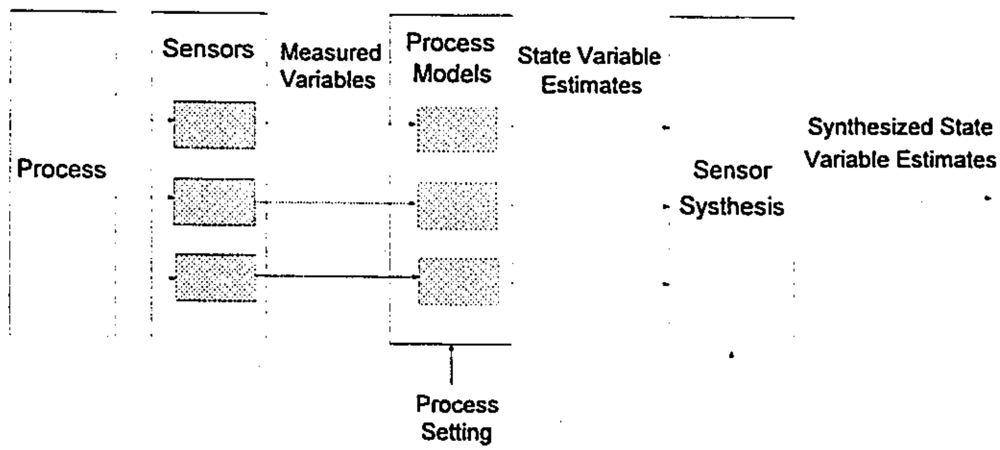


Figure 2. 4 Sensor Synthesis with Multi-Models / Chryssolouris *et al*, 1992 /

is built up by one model to deliver monitoring features. On the other hand, one can build up the signal of one sensor through several models to get more interesting features of the object. This system is called "Multi-Model-System" (Tönshoff and Wulfsberg, 1988). The different possibilities are the number of sensors and the number of models. The use of more sensors and models results in a more reliable and more flexible monitoring system. The increase of information density is indicated by the combination, as shown in Figure 2.3. A monitoring process performed by several sensors feeds the sensor signals into several process models. Each of these models contains a mathematical expression based on the mechanics of the cutting process. The information provided by the process models is then synthesized to determine the best estimation for the state variables. This strategy is called "Sensor Synthesis", which is illustrated in Figure 2.4 (Chryssolouris *et al*, 1988, 1992). Utilizing estimates from several sensor-based models can be considered analogous to taking several samples from a random distribution. As more sensor-based information is considered, the certainty of the estimated parameter values improves and the uncertainty due to randomness in the sensor signals is reduced.

2.5 EXPERT SYSTEMS

Expert Systems are a product of artificial intelligence (AI) (Rich, 1983). An expert system consists of three components: *Inference Engine*, *Man-Machine Interface*, and *Knowledge Base*. The inference engine is actually a search mechanism that finds an answer from the knowledge base for a given problem. The man-machine interface is the communication between human operators and the machine, which is designed for operators to use the system. Hence, the so-called expert system shells are developed to perform their tasks. The knowledge base contains the information about the given problem domain. The knowledge is usually represented by numerous "IF ... THEN ..." rules. It depends completely on the given problem domain and has to be developed independently. Therefore, the major effort in building an expert system is to establish the knowledge base.

Expert systems were reported to be used mainly in manufacturing for designs, process planning, and production control (Gupta and Ghosh, 1988; Iwata, 1988). However, the use of expert systems for machining process monitoring and diagnosis was also shown by Kishi (1986), Kumar and Ernst (1987), Spiewak and Wu (1988), Chryssolouris and Guillot (1988, 1990), Ramamurthi and Hough (1993) and others.

Kishi (1986) showed an example of a diagnostic expert system for automobile engines. This system consists of two main processing units: the unit which identifies the faulty parts, and the unit which searches for the causes of the problems. Kumar and Ernst (1987) proposed an architecture for expert systems that was based on a predictive monitoring control strategy. The architecture employs a hierarchy of models, and each model is detailed and "deeper" than the previous one. The performance of these models is monitored to recognize situations where the knowledge base has failed to produce the correct answer. These failures are used as guide to improve performance in the next cycle.

Spiewak and Wu (1988) presented a method of pre-processing measured force signals, based on the on-line identification of the cutting process. This method, called "intelligent filtering", allows decomposition of the measured force into several "streams" associated with major sources causing force fluctuations. They are: (1) the interrupted and superimposed cutting action of individual inserts, (2) vibrations forced by deterministic components of cutting forces, and (3) "white noise" excitation resulting from physical phenomena associated with the metal shearing process. In tool wear and failure monitoring, components of the cutting force associated with different sources of fluctuations are separated and analyzed independently. Tool condition estimates based upon individual signals are combined, with weights reflecting their confidence levels, to generate the final estimates. A block diagram illustrating this strategy is shown in Figure 2.5.

An artificial intelligence approach to the selection of process parameters in intelligent machining was proposed by Chryssolouris and Guillot (1988). This approach combines process modelling with rule-based expert system. The modelling techniques considered in this

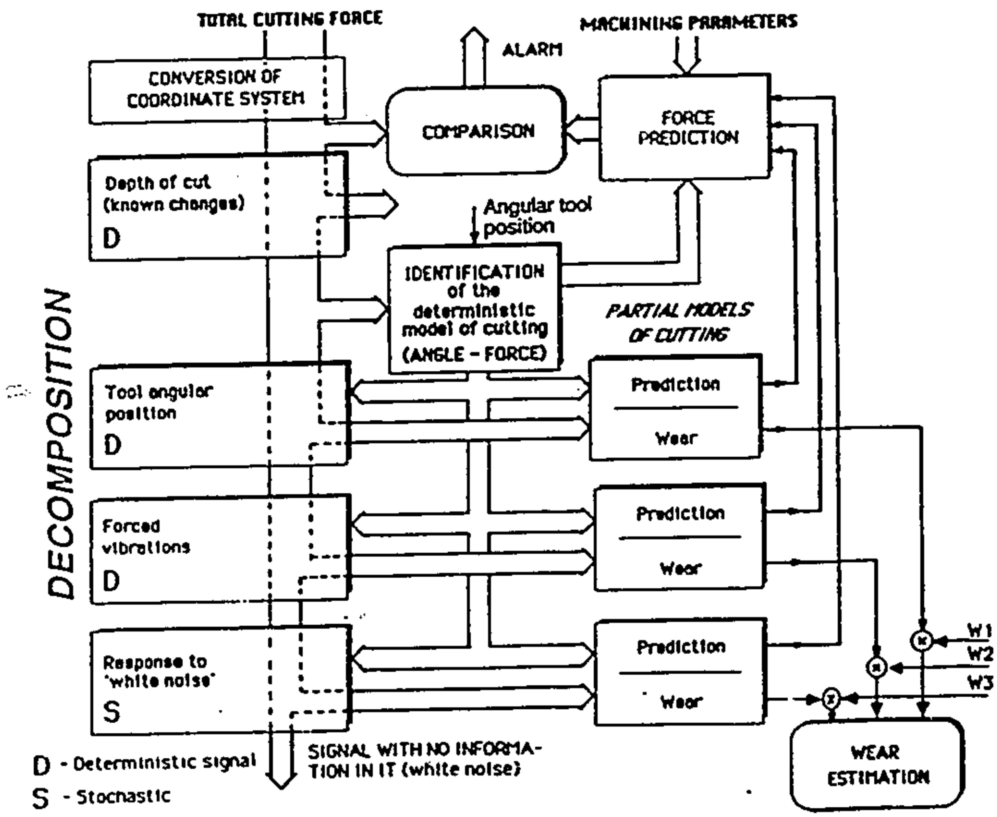


Figure 2. 5 "Intelligent Filtering" for Tool Wear and Failure Detection / Spiewak and Wu, 1988 /

work include multiple regression analysis, group method of data handling, and neural network. A rule-based module determined the final operational range of control parameters based on user information and modelling predictions. The different modelling techniques have been evaluated using data from orthogonal cutting.

Ramamurthi and Hough (1993) developed a generalized Machining Influence Diagram (MID) for real-time predictive diagnostics for cutting tools. It is formulated for modelling different modes of failure in conventional metal cutting process. A formal methodology is used to tune the knowledge base during training. The diagnostic system consists of four major subsystems: (1) machining influence diagram model, (2) speed algorithm, (3) numerical knowledge base, and (4) computational module. The first and the third components form the knowledge base. Some non-numerical information like the conditional probabilities are obtained from numerous variations in machining conditions and different types of sensors. The best judgment, however, should be then used to estimate or subjectively generate the numerical information.

Even though there have been few examples of application of expert systems in machining and tool condition monitoring, an expert system can be of significant benefit in tool condition monitoring as in the selection of process parameters. During machining monitoring, sensors, feature extraction, and decision algorithms could be properly selected by an expert system. This expert system should contain different models for various cutting conditions and different tool-workpiece pairs. It also should have the ability of learning to update the knowledge base and to improve its capability.

2.6 NEURAL NETWORKS

Artificial neural network models have been studied for many years in the hope of achieving human-like performance in the field of speech and image recognition. These models are composed of many nonlinear computational elements operating in parallel and arranged in patterns similar to biological neural networks. Such systems possess capabilities for fast learning and efficient pattern recognition. The applications of neural network in machining process and tool condition monitoring have been reported since 1987. The main contributions were made by Rangwala and Dornfeld (1987, 1990), Chryssolouris and Domroese (1988), Dornfeld (1990), Elanayar *et al* (1990), Liu and Ko (1990), Emel (1991), Tansel *et al* (1993a, 1993b), Pramod and Bose (1993), Govekar and Grabec (1994), Ko and Cho (1994), and many others.

Rangwala and Dornfeld (1987, 1990) demonstrated the feasibility of using neural networks to integrate information from acoustic emission and force sensors to monitor flank wear during turning. Networks are used as learning and pattern recognition devices. They were able to filter out noise in the sensor data and this enhances their ability for successful pattern association tasks over a range of machining conditions. It is shown that in cases where the sensor data is noisy and not very clustered, the classification performance benefits greatly through the use of multi-layered neural networks.

Chryssolouris and Domroese (1988) performed simulations in order to assess the learning capabilities of these networks. Based on the simulation results, they proposed the use of neural networks as the decision making component in an intelligent tool condition monitoring system.

Dornfeld (1990) developed and evaluated an on-line tool wear monitoring system for turning operations. By applying the multichannel AR time series models with the artificial neural network structure for learning the characteristics of the signals from multiple sensors, tool wear is effectively detected. It is stated that the various parameters in the neural network should be carefully chosen to ensure optimum performance and efficiency of the tool wear

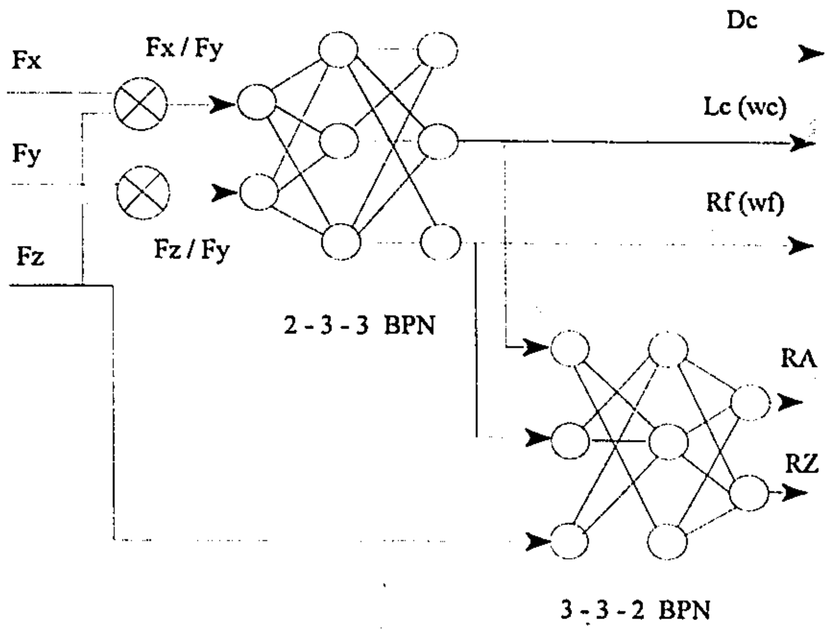


Figure 2. 6 Hierarchical Network Architecture for Machining Monitoring / Elanayar *et al*, 1990 /

detection system. For the purpose of tool wear detection in turning operations, relatively small sized networks work well. Ko and Cho (1994) also showed that networks with two hidden layers gave better performance than that with a single hidden layer. However, it would appear that three or more hidden layers could not improve the performance. They also came to the conclusion that a small network was preferable from the viewpoint of efficiency as well as fast learning.

Elanayar *et al* (1990) used a hierarchical network architecture to represent physical relations of the variables and reduce the network size. Two simple neural networks are connected to predict two critical machining conditions, tool wear and surface finish. The tool wear portion of the neural network is of a 2-3-3 type, while the surface finish portions uses a 3-3-2 network. This structure involves a two-stage back-propagation training procedure. Figure 2.6 shows the overall architecture. This configuration is also readily amenable to variations and modifications.

Different sensor signals were reported to be used in the signal integration by neural networks. Acoustic emission and the cutting force were used in turning by Rangwala and Dornfeld (1987, 1990), and Emel (1991). Cutting force/torque signals were used by Elanayar *et al*(1990), Govekar and Grabec (1994), and Ko and Cho (1994). Acceleration and thrust force were used by Liu and Ko (1990) and Tensel *et al* (1993a, 1993b) in tool wear monitoring for drilling and milling respectively.

In most of the work reported above, the most simple neural network structure is employed. It has three layers, one of which is the hidden layer. The number of outputs is selected as two or three (in most cases, worn and sharp tools, or initial wear and further wear). Feed forward neural networks are built and a back-propagation training algorithm is used. This simple structure makes it possible to apply neural networks for monitoring on the shop floor.

2.7 FUZZY CLASSIFICATION

Fuzzy concept was first introduced by Zadeh (1967, 1973) to deal with non-statistical uncertainties. Unlike probability theory, which describes the occurrence frequency of an uncertain event, the fuzzy set theory describes the impression of an uncertain event. In manufacturing, the fuzzy logic theory was mostly used in control applications. Recently, fuzzy patterns were introduced to describe the metal cutting states (Wang *et al*, 1985) and tool condition monitoring (Li and Wu, 1988; Du *et al*, 1992; Ko *et al*, 1992; Li *et al*, 1992; Chen, 1993; Ko and Cho, 1994; Li and Elbestawi, 1994, 1995a, 1995b). The use of the fuzzy set theory for process and tool condition monitoring offers the advantage of providing a systematic means for dealing with the inherent uncertainties in the metal cutting process, and particularly in describing the relationship between tool conditions and various process signatures.

Wang *et al* (1985) proposed the application of a fuzzy equivalent matrix method and the fuzzy interactive self-organizing data to the pattern recognition of metal cutting states. The fuzzy mathematics is applied to recognize the chip shapes (C-shape, coiled, tangled chip), the existence or non-existence of build-up edges, and vibrations of the cutting. Their results showed successful classification for types of chip, built-up edges, and vibrations.

Li and Wu (1988) introduced an approach for on-line monitoring of drill wear states by using a fuzzy C-mean algorithm. Experimental and simulation results have shown that drill wear conditions can be represented by four fuzzy grades: initial, small, normal, and severe. The purpose of fuzzy classification is to partition the features into the fuzzy grades. The tool replacement decision can be made upon the detection of the fuzzy grades "severe." The thrust force and torque are selected as the features relevant to drill wear states.

Du *et al* (1992) proposed the fuzzy linear equation method for tool condition monitoring. The fuzzy model is:

$$\mathbf{r} = \mathbf{Q} \circ \mathbf{p} \tag{2.4}$$

where, **r** represents the fuzzy degree of the input variables (monitoring indices), **p** represents the fuzzy degree of the classes (tool conditions), **Q** is the fuzzy relationship function, and the symbol "∘" is the fuzzy operator. This relationship is established based on the possibility distribution (frequency of occurrence) and the probability distribution (strength of support) of the learning samples. The proposed monitoring methodology was verified experimentally in turning under a range of cutting conditions. The details of this method will be discussed later. The results of classification represent a significant improvement over those obtained using other classical decision making strategies in pattern recognition methods.

Fuzzy pattern recognition for tool wear monitoring in diamond turning was performed by Ko *et al* (1992) and Ko and Cho (1994). The wear on tool edge is classified into two types: micro-chipping and gradual. Some features selected to partition the cluster of patterns are obtained from the adaptive auto-regressive time series modelling of dynamic cutting force signals. The optimal features which are sensitive to flank wear are selected using the fuzzy clustering mean scatter criterion.

Chen (1993) developed a fuzzy decision system for fault classification. Three fuzzy variables are defined to construct the fuzzy decision system: **C**, **R**, and **M**. **C** is based on the class fuzzy set. **R** is associated with the region set, which is defined by the distribution of indices. And, **M** corresponds to the two-dimensional fuzzy set representing the relationship between **R** and **C**. A membership function and a few weighing functions are used to determine **M**. The membership function is established based on information gained directly from data in a learning process. The weighting functions are determined objectively based on information measures carried by each index. Torque and thrust force are used for generating the indices for diagnoses in the tapping processes.

Li *et al* (1992) proposed a fuzzy decision tree algorithm for tool condition monitoring in turning. The strategy combines a decision-tree approach with the fuzzy set theory in developing a decision making strategy valid for a reasonable range of cutting conditions. Several sensors are used for the integration of sensory information including force, vibration, and cutting power signals. The process and tool conditions under the consideration are: three states of tool wear, tool breakage, and machining chatter. Unlike other fuzzy pattern recognition methods mentioned above, where fuzzy compatible matrices are used, this fuzzy decision tree employs a hierarchical structure to classify the tool conditions by several steps with the "pivot index" in the "maximum partition" strategy. These issues will be expanded upon later in this thesis.

One of the most recent approaches to automated machining and tool condition monitoring is the combination of fuzzy logic and neural networks. The author's contribution is the Multiple Principal Component (MPC) fuzzy neural network (Li and Elbestawi, 1994). Three major issues for developing the automated tool condition monitoring are supervised learning and classification (Li and Elbestawi, 1994), unsupervised learning (Li and Elbestawi, 1995a), and knowledge updating (Li and Elbestawi, 1995b). Experiments were carried out in turning and drilling. Good classification results were achieved in both cases. The development of this monitoring system is the major issue of this dissertation. The details of this approach by fuzzy neural network will be discussed in the following chapters.

Another effort in combining neural network and fuzzy logic by Mesina and Langari

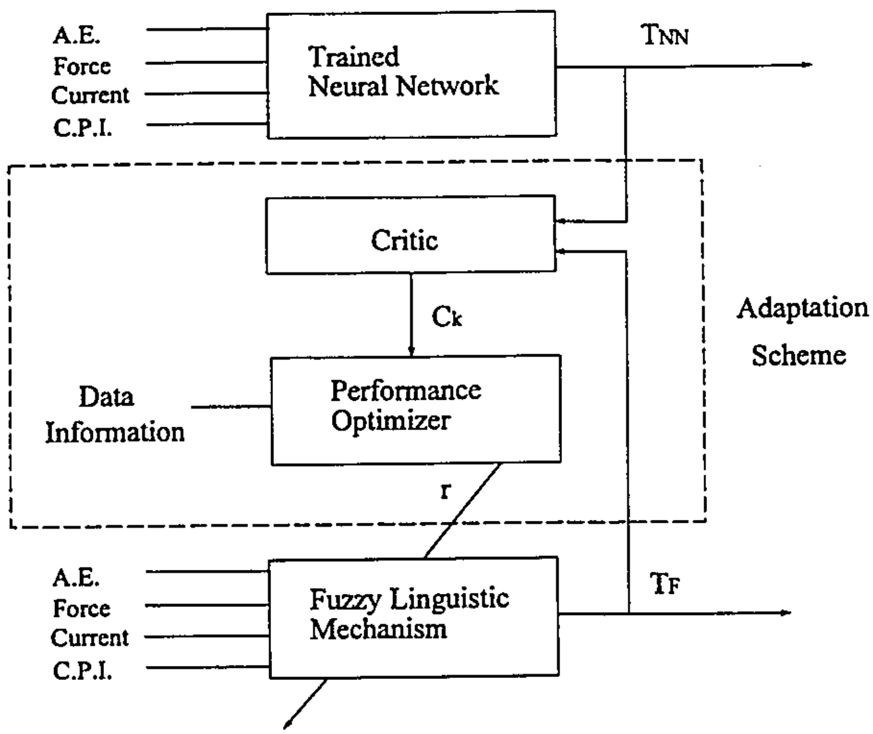


Figure 2.7 Architecture of the "Neuro-Fuzzy System" / Mesina and Langari, 1994 /

(1994) discusses a "neuro-fuzzy system" used to predict the tool conditions in a milling process. The architecture of the neuro-fuzzy system is shown in Figure 2.7. The input parameters are fed both to the trained neural network as well as the fuzzy linguistic mechanism. The difference between the output of the network and the fuzzy mechanism is taken as the prediction error and used to tune the membership functions. The error is used to refine the factor which contributed to this error. The error-based, density-driven adaptation strategy improves the performance of predicting the tool conditions.

2.8 RESEARCH ISSUES

From the above literature review, it is clear that the requirements of future automated machining process and tool condition monitoring system involve the following:

(1) "*Multi-Sensor System*" which uses more than one sensor for monitoring machining processes and tool conditions. It gives an extended survey of the sensitive features, as most process variables have an influence on one another.

(2) "*Automated Feature Extracting System*" which generates automatically, through learning procedures, the monitoring features. The signals sensed from multiple sensors are not simply fed to the system. They are analyzed, compacted, and selected by the system to generate the most sensitive features to the monitoring subjects. The extracted features are also further refined or reselected by the monitoring system.

(3) "*Learning and Decision Making System*" which builds up flexible and comprehensive monitoring strategies and generates automatically control parameters. The concentrated information from the learning procedure is stored in the system for classification purposes. The stored information can be modified by knowledge updating procedures. With increasing experience, the system will become more and more reliable and promote the monitoring/control functions. The learning and decision making strategies should be robust for shop floor applications and valid for a reasonable range of cutting conditions.

The objective of this dissertation is to develop and to verify experimentally a methodology for automated tool condition monitoring in machining. The major research issues related to this project are identified as follows:

(1) Selection/development of sensing system suitable for machining processes. Several mechanical parameters of interest include cutting forces, vibrations, and spindle motor current.

(2) Integration of information from multiple sensors (*i.e.* sensor fusion). Sensor fusion is applied in both feature selection and decision making.

(3) Development of a robust self-learning decision making strategy which is valid for the classification within a reasonable range of cutting conditions. This includes three major functions: supervised learning and classification, unsupervised learning, and knowledge updating.

(4) Applications of the proposed monitoring system to turning and drilling processes.

CHAPTER III

DECISION MAKING STRATEGIES FOR TOOL CONDITION MONITORING

3. 1 INTRODUCTION

The purpose of automated tool condition monitoring in machining is to relate the process signals to the tool conditions, and detect or predict the tool failure. Automated tool condition monitoring involves the act of identifying the characteristic changes of the machining process based on the evaluation of process signatures without interrupting normal operations. Basically, a monitoring process has three parts: *sensing*, *signal processing* and *decision making*.

Sensing is a process of obtaining the cutting process signals by sensors. Appropriate signals used for tool condition monitoring are force, torque, vibration, temperature, acoustic emission, electric current and so on. Various types of sensors and their applications have been developed and studied.

Signal processing and decision making may be considered as an integrated entity and called *monitoring methods* (Du *et al*, 1995). Monitoring methods are the major interest of all researchers working in this field. In general, monitoring methods can be divided into two categories (Du *et al*, 1995): model-based methods and feature-based methods. Both monitoring methods use sensor signals from the cutting process for the system input.

Metal cutting is a dynamic process. The sensor signals can be considered as the output of the dynamic system in a form of time series. Consequently, process and tool condition monitoring can be conducted based on system modelling and model evaluation. One of the most used models is linear time-invariant system, such as state space model, input-output transfer function model, Auto-Regressive (AR) model, Auto-Regressive and Moving Average (ARMA) model, and the Dynamic Data Systems (DDS) methodology. When a model is found, monitoring can be performed by detecting the changes of the model parameters and/or the changes of expected system responses. This kind of approach to automated tool condition monitoring in machining has been attempted by numerous researchers. The detailed discussion on the model-based monitoring is beyond the scope of this thesis.

Feature-based monitoring methods use suitable features of the sensor signals to identify the process and tool conditions. This is a mapping process to relate the tool conditions to the sensor features. Such a technique includes pattern recognitions, expert systems, neural networks, and fuzzy classifications. The feature-based methods consist of two phases: *learning* and *classification*. Learning, also called training, is the procedure of establishing the system structure and classification rules. The knowledge for decision making is obtained from the learning samples as well as from instructions. Updating and refinement of the stored knowledge are often required for improving the monitoring performance. This procedure is called knowledge updating or continuous learning. The system is retrained with new information available. Monitoring tasks are done in classification phase. The structure and the knowledge base built in the learning phase are used for the decision making in monitoring.

This chapter gives further discussions on some decision making strategies for tool condition monitoring. These strategies are pattern recognition, neural networks, and fuzzy classification.

3. 2 PATTERN RECOGNITION

3.2.1 The Pattern Recognition Problem

Pattern recognition refers to a process in which an input object is measured, analyzed, and classified by a machine as being more or less similar to a predefined prototype stored in memory. The goal of pattern recognition is to provide a machine with a kind of perceptual capability so that it can be used to automatically analyze and extract useful information from raw data. Figure 3.1 shows a conceptualized structure of pattern recognition system. It has

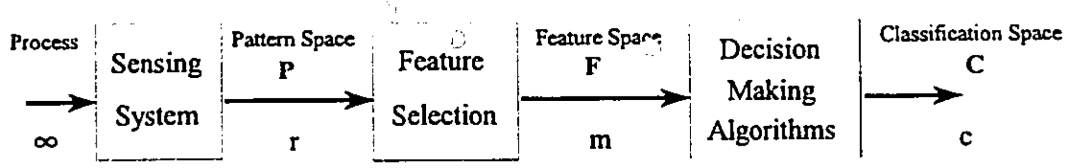


Figure 3.1 Conceptualized Pattern Recognition Problem

three stages: *sensing system, feature selection*, and *decision making algorithms*. In physical world, a process can be represented by a continuum of parameters and basically has infinite dimensions. A sensing system is designed to collect certain representative data of that process. Then, the process will be described by r scalar values. At this stage, a pattern space is created with the dimensions of r, where r is usually quite large. In order to focus on features of the monitored process, the dimensions of the space have to be reduced. This is executed by the feature selection (or feature extraction). The r-dimensional pattern space is converted to an m-dimensional feature space in which the discriminant capability of classification purposes is still maintained. Thus the feature space, in which classification rules can be computed in reasonable amounts of time, is proposed of dimension m being much smaller than r. In the last stage, decision making maps a variable in the feature space into the classification space, in which one of c classes has been selected. The decision making is carried out by the function of the decision algorithm. Classification space has c dimensions, obviously.

Thus, the pattern recognition problem can be described as a transformation from the pattern space P, to the feature space F and, finally, to the classification space C. The dimensions of these spaces are reduced one by one.

A variable can be represented by a point in the feature space. The classification problem is now simply one of finding separating surfaces in *m* dimensions which correctly separate the known prototypes. The feature variable should also afford some degree of confidence in correctly classifying unknown patterns. In order that such a task be successfully carried out, it becomes necessary to define similarity measures between points in the feature space. And, the use of such a metric presupposes that the feature space is a metric space. The necessary metric must then satisfy the following conditions with relation to these points **x**, **y**, and **z** in the space (Andrews, 1972):

(1) $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$;

(2) $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{y}, \mathbf{z}) + d(\mathbf{x}, \mathbf{z})$;

(3) $d(\mathbf{x}, \mathbf{y}) \geq 0$;

(4) $d(\mathbf{x}, \mathbf{y}) = 0$, iff $\mathbf{y} = \mathbf{x}$.

Where, $d(\mathbf{x}, \mathbf{y})$ is a function relating the two points. Such a metric is often referred to as a distance function of which there is a large variety. Each dimension may be a measure of unrelated parameters and must itself be normalized before being combined with other dimensions, as in a distance calculation. This normalization makes the dynamic ranges of various axes to be somewhat well behaved. Various normalization techniques exist (refer to Andrews, 1972 or James, 1985) and should be selected according to their respective applicability to a given pattern recognition problem.

3.2.2 Feature Selection

A pattern is assumed to have certain properties or attributes which distinguish it from other patterns. In observing a pattern, measurements are made which reflect, either directly or indirectly, these attributes. The pattern space is defined by sensor data which itself may be defined by convenience rather than classification discriminatory power. These measurements may carry a very small amount of information and result in high dimensionality in the pattern space. For convenient classifications, each measurement should be selected according to the physical meaning of the problem and yield significant information for classification purposes. In order to make the classification algorithm simple, we have to find some way to select or extract distinguishing features from the observed samples. This problem is called *feature selection* or *feature extraction*. It is an important part of pattern recognition.

Features are functions of the measurements which are intended to facilitate classification. Feature selection can be considered as a mapping from the r-dimensional pattern space to a lower dimensional space, for instance, m. This space is called feature space. Thus, the feature space must not only be defined by the inherent discriminatory power of data presented in the pattern space, but also be optimized for specific class problems. It is desirable that the dimension of the feature space, m, be much smaller than the dimension of the pattern space, r. A variable vector in the feature space is represented as

$$\mathrm{y} = [\, y_1, y_2, \ldots, y_m \,]^T. \tag{3.1}$$

The objective of feature selection in defining a feature space is to reduce the dimensionality of the pattern space yet maintain the discriminant capability for classification purposes. A variety of techniques have been developed. The real frontiers of pattern recognition research still lie ahead in developing a viable feature selection transformation that minimizes the redundant data gathering inherent in the definition of the pattern space (Andrews, 1972).

Feature selection is generally considered a process of transforming the original measurements into more effective features. This mapping can be either linear or non-linear. Generally, there are two analytical methods of feature selection: *canonical analysis* and *variable selection*.

A. Canonical Analysis

Let $\mathbf{x}$ be an r-dimensional variable from the pattern space. We may state the feature selection problem in a fairly general form, thus:

> *find a matrix $A(m \times r)$ such that a classifier using $y = Ax$ has the best error rate achievable* (James, 1985).

As the number of features is reduced to m, it is usual to add the condition that m is as small as possible without losing too much performance.

Consider a two-class case in two dimensions. The two classes can be represented by ellipsoids indicating the regions in which samples are likely to fall, as shown in Figure 3.2. An arbitrary line between the two classes is drawn. The difference between the projections of the two class means depends on the angle of the line. The size of the projected spreads on the line also depends on the angle.

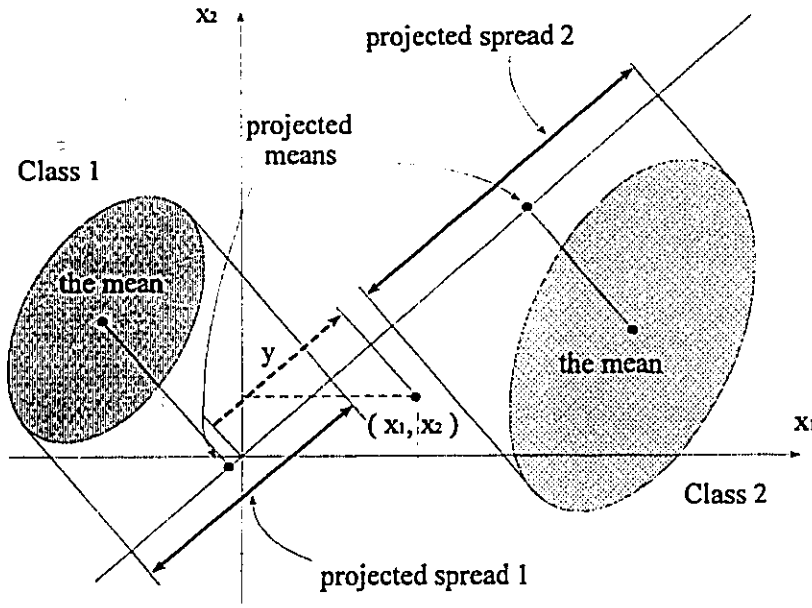


Figure 3.2 Canonical Conversion of the Feature Space

Using distance measured along the line as a new variable, y, then y is given by the simple expression:

$$y = \sum_{i=1}^{2} v_i x_i = v_1 x_1 + v_2 x_2 , \tag{3.2}$$

where, the v_is are constants which define the orientation of the line. The "Within group Sums of Squares (SSW)" can be introduced to measure the spread of cases with any of the classes, and the "Between group Sums of Squares (SSB)", to measure the spread of the class means.

Then, the best line to separate the two classes is the one which maximizes the ratio SSB/SSW.

More generally, for a c-class, r-dimensional problem, some other lines can also be found to maximize SSB/SSW. The standard extra condition is to require that the scores on a line be uncorrelated with the others. Each of these lines is known as an eigenvector and the value of the sums of squares associated with each is called an eigenvalues. If we have the normal equal covariance case, or if we are using linear discriminant functions, then we have

> *Linear discriminant functions constructed using only the eigenvectors with non-zero eigenvalues give the same results as linear discriminant functions constructed using the original measurements* (James, 1985).

That means all the information used by the linear discriminant functions is contained in just few eigenvectors.

In normal cases, we often want to choose the number of indices, m, being less than r and we will lose classifying power. It is also possible that another set of m linear combinations exists which will give a smaller error rate. It is true that there is no simple and efficient method of obtaining m linear combinations of the original variables that give the smallest error rate. The selection of the combination depends on the definition of the error as well as on calculation algorithms. Therefore, the feature extraction by canonical analysis can be presented as a search, among all possible singular transformations, for the best subspace which preserves class separability as much as possible in the lowest possible dimensional space.

B. Variable Selection

Canonical analysis is used for generating a feature space with linear combinations of all original variables in the pattern space. If we want to reduce the dimension of the pattern space, or to select the most information-bearing signals from the pattern space, we usually use another method: variable selection. Four different methods of selecting a "best" set of m variables can be identified as follows (James, 1985):

(1) *Complete subsets*: This consists simply of finding every possible subset of size m of the variables and calculating the measure of the best set on each. The solution is the set with the largest value of the measure. This method fails when m and n are too large for the subset being considered.

(2) *Stepwise forward*: First, we find the single variable which maximizes our measure of the best set. Then we find the second variable which, when paired with the previously selected variable, maximizes the measure of the best. Processed in same way, a third one is added to the previous two and so on until m variables are selected.

(3) *Stepwise backward*: We start searching with all the variables and discard the variable which results in the smallest lowering of our measure of the best. The search is stopped when m variables are left.

(4) *Full stepwise*: The full stepwise method works by examining, at each stage, the decrease in the measure of the best set produced by removing a variable. If the decrease is below a specified threshold, the variable is removed. If no variable meets this criterion, a variable is added by the usual forward stepwise method. This is the combination of methods (2) and (3).

Feature extraction is a huge objective in pattern recognition research. Numerous criteria have been studied, and different methods will give different results even with the same variables in pattern space. For further details about the various algorithms, readers should refer to the text books and the papers on pattern recognitions.

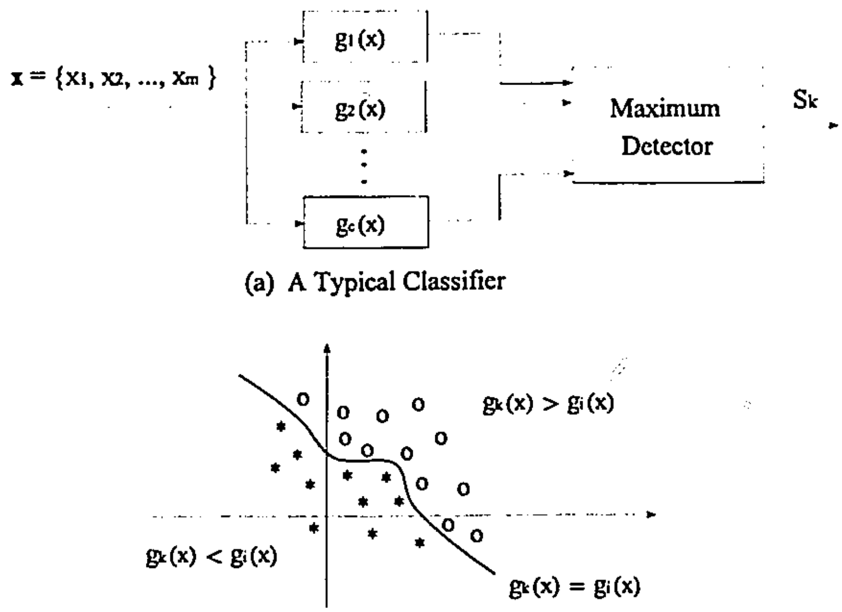
3.2.3 Classification

Classification refers to the association of a class with a particular feature vector. The distinction between classification and feature extraction is somewhat arbitrary, and considerable overlap exists. Given a good feature set, the classification can be simple. Even powerful classifier will fail if inadequate features are selected.

The problem of classification is to find a way of assigning a new object in the feature space to one of a number of possible classes. A classification rule must be defined to accomplish this assignment. The classification rule should be, in some sense, as all-encompassing as possible so that the "Total Error of Classification", or TEC, is minimal.

For the *c* pattern classes: $S_1, S_2, ..., S_c$, there is a function which measures each point in the feature space and assigns to that point a value as to its degree of membership in a given class. Such functions are called *discriminant functions* in the context of pattern recognition. Discriminant functions have the property that they divide the feature space into mutually exclusive regions, each region contributing to the domain of a class.

The discriminant function is defined such that for all points **x** within the region describing S_k, there exists a function $g_k(\mathbf{x})$ such that $g_k(\mathbf{x}) > g_i(\mathbf{x})$ for all $i \neq k$. Mathematically, it is expressed as follows:



(a) A Typical Classifier

(b) A Decision Surface

Figure 3.3 Classifier with Discriminant Functions

$$g_k(x) > g_i(x) \qquad \forall\ x \in S_k \ \wedge\ \forall\ i \neq k \tag{3.3}$$

Thus, within region S_k, the *k*th discriminant function will have the largest value. The surface separating region S_k and S_i is given by

$$g_k(x) - g_i(x) = 0 \tag{3.4}$$

which is equivalent to those points in the space which have equal discriminant function for both class S_k and S_i. There are $c(c-1)/2$ such separating surfaces in a c class problem. Figure 3.3 shows the discriminant function classifier and a possible separating surface in two-dimensional space.

A. Bayes Classifier

Bayes' rule is the essential base for statistical classification. As conditional probabilities can be used to summarize any information that we have about an event, they are central to the classification of an object based on any measurements that we have made. Bayes' rule states that:

assign the object to the group with the highest conditional probability.

Formally, if there are c classes, then Bayes' rule is to assign the object to class k where:

$$P(S_k \mid x) > P(S_i \mid x) , \quad \forall \quad i \neq k . \tag{3.5}$$

If, by any chance, there is more than one class with the largest conditional probability, then the tie can be broken by allocating the object at random to one of the groups concerned. The success of Bayes' rule is that it shows that the information about possible group membership is contained in the set of conditional probabilities.

However, quantities such as $P(S_k|\mathbf{x})$ are very difficult to find by the standard method of estimation. But, the conditional probability $P(\mathbf{x}|S_k)$, the probability of getting a particular set of measurements $\mathbf{x}$, given that the object comes from class k, is something that can be estimated simply by taking a sample of objects from class k. To get $P(S_k|\mathbf{x})$ from $P(\mathbf{x}|S_k)$, we can use Bayes' theorem:

$$P(S_k \mid x) = \frac{P(x \mid S_k)\, P(S_k)}{\sum\limits_{\forall i} P(x \mid S_i) P(S_i)} \, . \tag{3.6}$$

Putting Bayes' theorem into Bayes' rule gives the following result:

assign to class k if

$$P(x \mid S_k)\, P(S_k) \; > \; P(x \mid S_i)\, P(S_i) \, , \qquad \forall \;\; i \neq k \, . \tag{3.7}$$

To apply Bayes' rule, we have to know the value of $P(S_i)$ and $P(\mathbf{x}|S_i)$ for each class. Although $P(S_i)$ is easy enough to find, $P(\mathbf{x}|S_i)$ is a rather difficult problem. However, if within each class, the variables that make up the measurement vector **x**, have a multivariate normal distribution, then the form of $p(\mathbf{x}|S_i)$ is known. So that

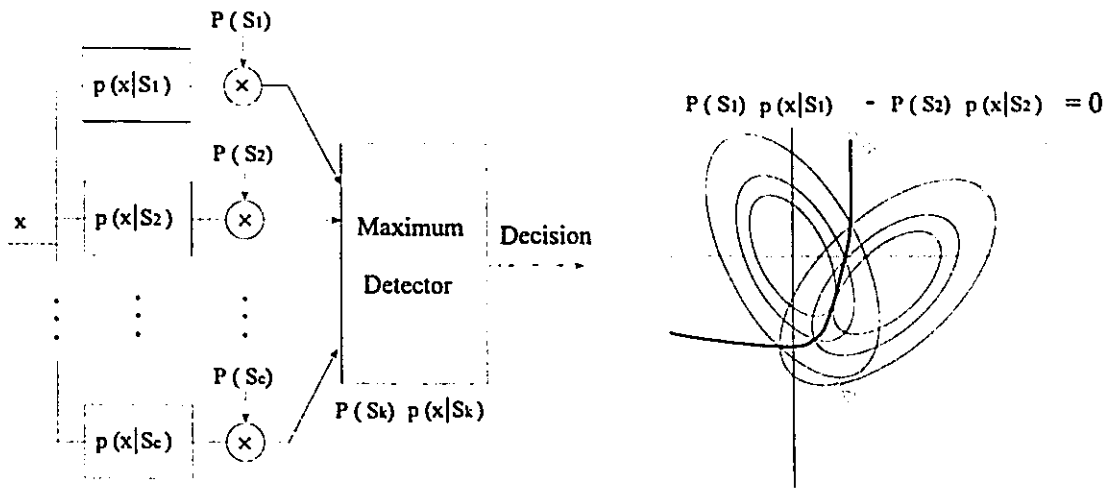
$$p(x \mid S_i) = \frac{1}{(2\pi)^{N/2} \mid \Sigma_i \mid^{1/2}} \exp\left[-\frac{1}{2}(x-\mu_i)^T \, \Sigma_i^{-1} (x-\mu_i) \right] . \tag{3.8}$$

In this case, estimating $P(\mathbf{x}|S_i)$ comes down to estimating two parameters for each class, $\boldsymbol{\mu}_i$, the group mean vector, and Σ_i, the class covariance matrix.

Using the normal form of $P(\mathbf{x}|S_i)$ in Bayes' rule gives:

assign x to S_k if

$$\begin{aligned} &\ln|\Sigma_k| + (x-\mu_k)^T \, \Sigma_k^{-1} (x-\mu_k) - \ln(P(S_k)) < \\ &\qquad \ln|\Sigma_i| + (x-\mu_i)^T \, \Sigma_i^{-1} (x-\mu_i) - \ln(P(S_i)) \, , \qquad \forall \; i \neq k \, . \end{aligned} \tag{3.9}$$



(a) Bayes Classifier (b) Bayes Decision Surface

Figure 3.4 Bayes Classification

Figure 3.4(a) presents a block diagram of pattern recognition machine for the Bayes classification.

Introducing the concept of discriminant function

$$g_i(x) = \ln|\Sigma_i| + (x-\mu_i)^T \Sigma_i^{-1} (X-\mu_i) , \tag{3.10}$$

we have Bayes' rule as :

assign to class k if

$$g_k(x) - \ln(P(S_k)) < g_i(x) - \ln(P(S_i)) , \qquad \forall \quad i \neq k . \tag{3.11}$$

Where, x is multivariate normal in each of the groups.

If the prior probabilities $P(S_1)$ and $P(S_2)$ for a two-dimensional two group problem are assumed to be equal, as shown in Figure 3.4(b), then the dividing line between the two regions is given by

$$P(x \mid S_1) - P(x \mid S_2) = 0 \tag{3.12}$$

It is known that the dividing surfaces are of quadratic form for a normal Bayes classifier.

B. Linear Classifier

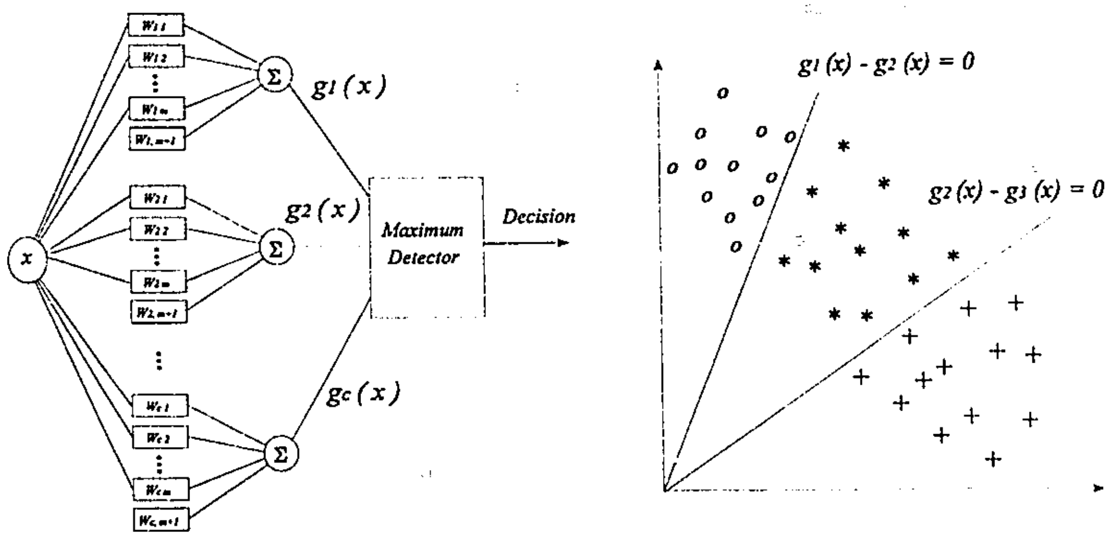
In a problem of classification with c classes, only $(c\text{-}1)$ discriminant functions are needed to separate those classes. The construction and adjusting of discriminant functions are referred to as "training" or "learning". If the training is based upon statistics, parametric and certain nonparametric techniques are used. Some classification theories rely on an approach completely independent of statistical knowledge or assumption, often referred as distribution free or nonparametric classification. If the training is based on an assumed functional form for the discriminant function, distribution free techniques are employed.

Linear classifier is the simplest assumed functional form for discriminant functions. Such a function can be represented in scalar and vector forms as

$$g_i(X) = w_1 x_1 + w_2 x_2 + \ldots + w_m x_m + w_{m+1} \tag{3.13}$$

or

$$g_i(x) = W_i^T x \tag{3.14}$$



(a) Linear Classifier

(b) Linear Decision Surfaces

Figure 3.5 Linear Classification

A scalar term w_{m+1} is added to allow a translation of all linear discriminant function to pass through the origin of the augmented space when desired. Figure 3.5 gives an example of this kind of classifiers.

One of the simplest classification algorithms using a linear discriminant function is the minimum distance classifier. Suppose that the average point of the prototypes defining a given class S_i be given by

$$<y_i> = \frac{1}{M_i} \sum_{m=1}^{M_k} y_m^{(i)} , \qquad (3.15)$$

and there exist c such points in m space. Let the Euclidian metric be assumed in the space and let the classifier assign a unknown point **x** to that space which has its average value $<y_i>$ closest to **x**. Thus, the decision rule becomes

assign to group k if

$$d(x, <y_k>) \; < \; d(x, <y_i>) \qquad \forall \quad i \neq k \tag{3.16}$$

The discriminant function can be obtained as :

$$g_i(x) \; = \; x^T <y_i> - \frac{1}{2} <y_i>^T <y_i> \; . \tag{3.17}$$

In the context of linear discriminants, the elements of $<y_i>$ become the linear weights and also the augmenting quantity. The decision surface is defined to be the plane between the perpendicular bisector separating points $<y_i>$ and $<y_j>$.

Another way of defining discriminant functions is given by the piecewise linear functions. The separating surface of piecewise linear machines no longer defines convex regions in the pattern space as the linear discriminant function does. The classic example for the piecewise linear machine is the minimum distance classifier with respect to prototypes. Thus, the distance of an unknown **x** from a class S_i may be :

$$d(x, S_i) \; = \; \min_{m=1,\dots,M_k} \; \{ \, d(x, y_m^{(i)}) \, \} \; , \tag{3.18}$$

The distance become the smallest distance between all the prototypes of S_i and the unknown **x**. The discriminant function corresponding to such an algorithm is

$$g_i(x) = \max_{m=1,\ldots,M_k} \{x^T y_m^{(i)} - \frac{1}{2} y_m^{(i)T} y_m^{(i)}\} . \tag{3.19}$$

If the second or higher orders are included in discriminant function, a quadratic surface or a polynomial surface is defined respectively.

C. Potential Function Method

In linear discrimination, a polynomial surface is always assumed. Instead, the so-called potential function method utilizes superposition such that a function is defined for each prototype over the entire pattern space with variable **x**. Such a function, known as a kernel in probability density function estimators, will be denoted as $\Psi(\mathbf{x}, \mathbf{y}_m^{(i)})$, where $\mathbf{y}_m^{(i)}$ is the mth prototype defining class S_i. The sum of these individually kernel and "potential" functions will then become the discriminant function:

$$g_i(x) = \frac{\sum_{m=1}^{M-k} \Psi(x, y_m^{(i)})}{M_i} . \tag{3.20}$$

Where, the $\Psi(\cdot)$ function may be different between classes or even between prototypes within a class. These functions should reflect a decreasing influence of a sample point, **x**, upon points in the pattern space, $\mathbf{y}_m^{(i)}$, as the distance between the two points, $d(\mathbf{x}, \mathbf{y}_m^{(i)})$, increases. The average of these $\Psi(\cdot)$ kernels or "potentials" of prototypes from a given class indicates the degree of membership of the point **x** in the class. Desirable characteristics of potential functions might be enumerated as follows.

(1) (x, y) should be maximized for x = y.

(2) (x, y) should be approximately zero if the distance from x to y is too far in the region of interest.

(3) (x, y) should be continuous and decrease approximately monotonically with distance d(x, y).

(4) If $(x_1, y) = (x_2, y)$ where y is a prototype, then the pattern represented by x_1 and x_2 should have approximately the same "degree of similarity" to y.

3.2.4 Pattern Recognitions for Tool Condition Monitoring

Pattern recognition technique has been applied to recognize the cutting states and to monitor the tool conditions in machining for decades. The most simple and popular algorithm is linear classifiers.

In applications of linear classifiers (Zhang *et al*, 1982; Marks and Elbestawi, 1988; Monostori, 1988; Liu and Wu, 1989; and the like), the linear model represented in Equation 3.13 was used. The features for classifying the cutting states included cutting speed, feed and the power spectrum in different frequency bands. The features used for the tool condition monitoring are usually feed rate, depth of cut, cutting force, cutting torque, sums of the magnitudes of spectral components at certain frequencies, and other signal features. Experiments showed that the number of features and the different combinations of features had great effects on the correct classification rates. An error-correction procedure was used to obtain the weight vectors. Arbitrary initial weight vectors were selected and adjustments were made whenever the classification scheme responded incorrectly to any pattern. The success rates of classification with these cases are 77% and greater.

Other pattern recognition algorithms for tool condition monitoring in machining included the class-mean scatter criterion, the class variance criterion, and Fisher's weighted criterion (Emel and Kannatey-Asibu, 1987, 1988). The class-mean scatter criterion maximizes class separation and minimizes within-class variance. The class variance criterion maximizes the difference between the within-class variance of each class. Fisher's weight criterion maximizes class separation and minimizes the within-class variance between each pair of classes. This methodology was applied in order to detect tool wear and breakage in turning operations using acoustic emission spectral information under fixed cutting conditions. The tool wear sensing results had performances ranging from 84 to 94%.

3. 3 NEURAL NETWORKS

3.3.1 Artificial Neural Networks

Artificial neural networks are an attempt to emulate the computational architecture of the human brain in electric hardware. Neural networks have great potential in areas of intelligent functions such as learning and pattern recognition where many hypotheses are pursued in parallel, where high computation rates are required, and where the current best system are far from equalling human performance. However, the architecture of human brains is exceedingly complex and not well understood at present, so that the current neural network architectures resemble the brain only at very coarse level. Despite such rudimentary representatives, they are very useful in the development of computer architectures based on a high degree of parallelism and are able to learn by adapting the strength of the connections between the processors. The potential benefits of neural networks extend beyond the high computation rates provided by massive parallelism. Neural networks typically provide a great degree of robustness and fault tolerance.

The architecture of a neural network is specified by the net topology, neuron characteristics, and training (or learning) rules. Theses rules specify an initial set of parameters and indicate how the system parameters should be adapted during use to the improve performance. Both design procedures and training rules are the topics of much current research. In general, neural networks do not do well at precise, numerical computations (Caudill, 1989). Neural networks can be taught to obtain the knowledge about pattern recognition or other assignments.

Figure 3.6 presents block diagrams of traditional pattern classifiers and neural network classifiers (Lippmann, 1987). Both types of classifiers determine which of c classes is the most likely representative of an unknown input pattern containing m input elements. In machining tool condition monitoring, the inputs might be process signals such as cutting forces, vibrations and acoustic emissions, and the classes represent different tool conditions. Inputs and outputs of a traditional classifier are passed serially and internal computations are performed sequentially. In addition, parameters are typically estimated from training data and then held constant. Inputs and outputs to a neural network classifier are in parallel and internal computations are performed in parallel. Internal parameters or weights are typically

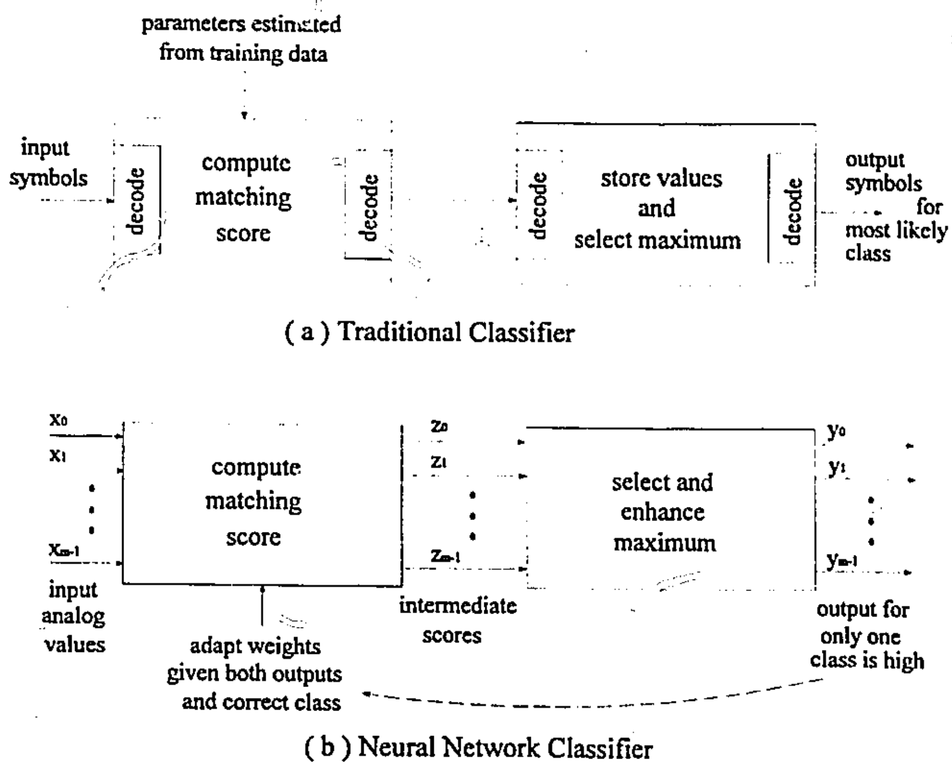


Figure 3.6 Block Diagrams of Traditional and Neural Network Classifiers / Lippmann, 1987 /

adapted or trained while using the output values and the labels specifying the correct class.

Traditional classifiers have two stages for classifications. The first computes matching scores for each class and the second selects the class with the maximum score. An algorithm computes a matching score for each of the c classes which indicates how closely the input matches the exemplar pattern for each class. This exemplar pattern is that pattern which is most representative of each class. In many situations, a probabilistic model is used to generate the input patterns from exemplars. A matching score is computed to represent the likelihood or probability that the input pattern was generated from each of the c possible exemplars. Matching scores are coded into symbolic representations and passed sequentially to the second stage of the classifier. There, they are decoded and the class with the maximum score is selected.

The input values to a neural network classifier are fed in parallel to the first stage *via* m input connections. The first stage computes matching scores and outputs these scores in parallel to the next stage over c analog output lines. Here, the maximum of these values is selected and enhanced. The second stage has one output for each of the c classes. After the classification is completed, only that output corresponding to the most likely class will be designated on "high"; other outputs will be "low". If the correct class is provided, then this information and the classifier outputs can be fed back to the first stage of the classifier to adapt parameters using a learning algorithm. The adaptation will make a correct response more likely for succeeding input patterns that are similar to the current pattern. Neural network classifiers are nonparametric, so the form of input distributions is not assumed and the parameters of distribution are not estimated.

A neural network is composed of many simple processing elements that typically do little more than taking a weighted sum of all its inputs. The simplest processing element or neuron sums m weighted inputs and passes the results through a non-linearity. Neurons used in neural networks are usually nonlinear and analog. Figure 3.7 illustrates three common types of nonlinearities: hard limiter, threshold logic element, and sigmoidal nonlinearity. More complex neurons may include temporal integration or other types of time dependencies and more complex mathematical operations than summation. A neural network does not execute a series of instructions; it responds in parallel to the inputs presented to it. The

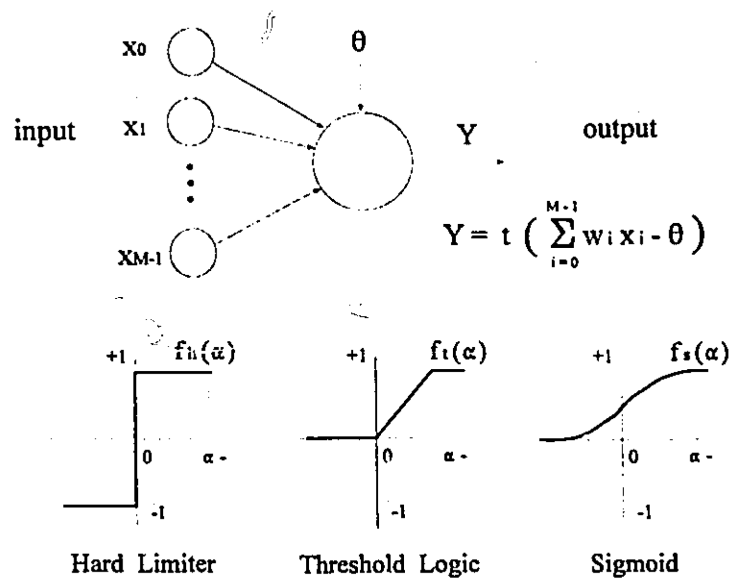


Figure 3.7 Nonlinerities at a Neuron / Lippmann, 1987 /

knowledge within a neural network is not stored in a particular location. It is stored both in the way the processing elements are connected and in the importance of each input to the processing elements. Such knowledge is more a function of the network's architecture or structure than the contents of a particular location.

Neural network classifiers work well for many real-world problems. These classifiers frequently provide reduced error rates when compared to more conventional Bayesian approaches (Lippmann, 1989). Furthermore, neural network classifiers differ in their ability to use unsupervised training data and in the ease with which internal operations can be understood and interpreted to determine what input features contribute to classification performance. These issues, more than error rate, tend to drive the selection of a classifier to a particular application.

3.3.2 Structure of Neural Networks

A neural network is a computing system made up of a number of simple, highly interconnected processing elements (neurons). The computing system processes information by its dynamic state response to external inputs (Caudill, 1989). By this definition, two main elements make up a neural network: *processing elements* and *interconnections*. The structure of a neural network is defined by the interconnection architecture between the processing elements, the rules that determine whether a processing element will fire, and the rules governing changes in the related importance of individual interconnections to a processing element's input. It is constructed and modified by the training of the network.

A neural network is usually divided into three parts: the input layer, the hidden layer, and the output layer. The input layer $F_A = (a_1, a_2, ..., a_m)$ has m processing elements, one for each m dimension of the input pattern A_h. On the output layer, F_c, each node represents a pattern class of conclusion. If the input and the output layers are directly connected, a neural network becomes a single-layered network that functions as a direct mapping of the input to the output. Adding the third layer to the neural network allows it to develop its own internal representation of the mapping. The network is not dependent on the intrinsic relationship built into the data, but can determine for itself which is important in representing the mapping (Caudill, 1989). The importance of this layer, therefore, is mainly due to its dynamics. The connections can be any functions or logic reasoning that is suitable to the problem and may be rebuilt through its previous experiences. The middle layer is usually called a hidden layer because of its flexibility. Figure 3.8 shows a typical structure of neural networks with one hidden layer.

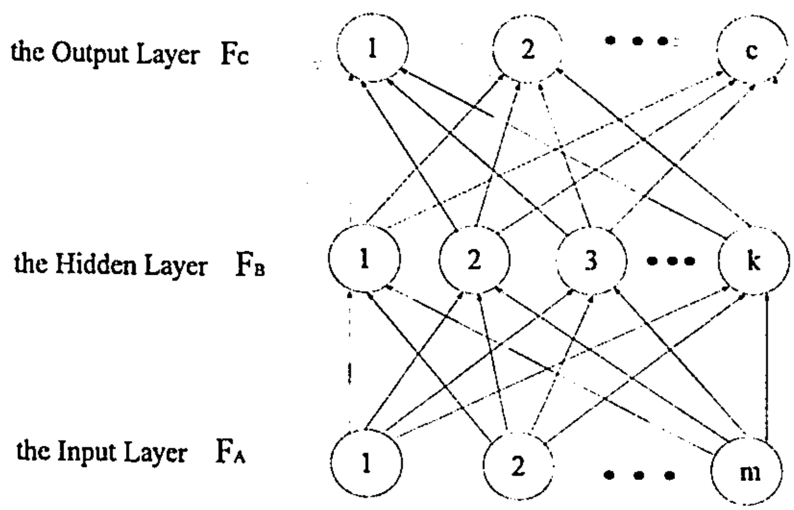


Figure 3.8 Artificial Neural Network

The hidden layer plays a very important role in constructing neural networks. If it is eliminated, as mentioned before, we have a direct mapping function. A switching function is a good example of a direct mapping function. By using weighted linear functions to connect the input elements with the output elements, we obtain a linear classifier. If nonlinear functions (say, sigmoidal) are used in the hidden layer, we obtain a typical neural network that is commonly used in the field of manufacturing process monitoring. Furthermore, a fuzzy neural network is created by introducing fuzzy logic functions into the hidden layer for interconnections. In some cases, not all of the interconnections between neurons in the network are necessary when the calculations are performed. Therefore, a partially connected neural network can be formed by keeping only those connections necessary. The effects of the hidden layer on a neural network are depicted in Figure 3.9.

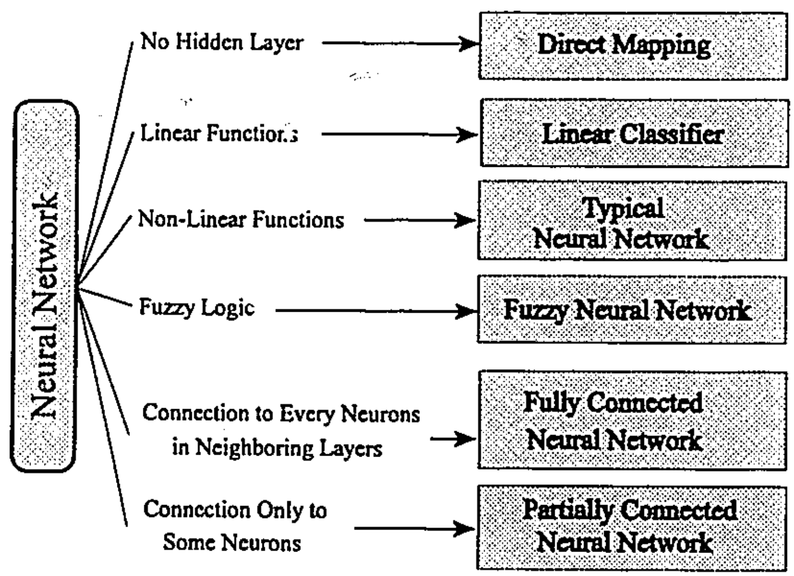


Figure 3.9 Effect of the Hidden Layer on Neural Networks

3.3.3 Learning in Neural Networks —— Back-Propagation

The knowledge of a neural network is obtained from learning processes. Learning implies that the processing element somehow changes its input/output behaviour in response to the environment. Learning from experience provides a vital organism with the means to adapt to and survive in a changing environment. For pattern classifiers, the learning produces reliable, enhanced and flexible performances.

Learning in neural networks can be supervised or unsupervised. Supervised learning means the network has some reliable inputs presented during training to tell it what the correct answer should be. The network then has a means to determine whether or not its output is correct and knows how to apply its particular learning rules to adjust its weights.

Unsupervised learning means the network has no such knowledge of the correct answer and, thus, cannot know exactly what the correct response should be before the learning. Supervised classification has been commonly applied to the tool condition monitoring in machining.

One particular learning rule which is the most commonly used learning algorithm is the Delta rule or Least Mean Squared (LMS) training law. In learning , the rule will tell how to change the weights depending on whether or not the output of classification was correct. Figure 3.10 shows a processing element modified by the Delta learning rule. First, we have to modify the processing element so it can monitor its own output. Then, we enable it to compare its output to the desired output signal, I_0, and compute the error value, E, for this input pattern. E is computed by subtracting the actual output, y, from the desired response I_0: $E = I_0 - y$. Finally, we calculate how to change the weights by using the Delta rule:

$$W_{new} - W_{old} = \frac{\beta\ E\ X}{|X|^2}\ , \tag{3.21}$$

where, X and W are the input and weight vectors respectively, $|X|$ is the length or magnitude of the input pattern vector and β is a learning constant. Note that the Delta rule is a vector equation. The error E and constant β are scalar values, and the other elements are vectors. The LMS rule attempts to insure that the aggregate statistical LMS error is minimized in the network. In this case, the error in the weights of the processing element is based on an ideal value for the weights. We compute the current error, or how much it deviates from this ideal value, for the weights for this input. We then adjust the weights by adding this Delta vector

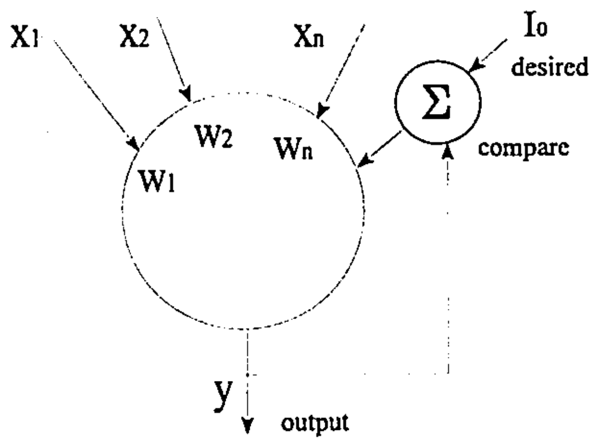


Figure 3. 10 Processing Element Modifying Itself by Learning / Caudill, 1989 /

to our current weight vector. This process has a simple geometric interpretation. It can be shown mathematically that the aggregate mean squared error is a function of the weight vector. It is shown in Figure 3.11.

Delta rule is a gradient-descent learning rule. The learning constant β is a measure of the speed of convergence of the weight vector to the minimum error position. A back-propagation neural network is built up of processing elements by using the Delta rule.

The back-propagation algorithm uses a gradient search technique to minimize a cost function equal to the mean square difference between the desired and the actual output. The neural network is trained by initially selecting small random weights and internal thresholds, and then presenting all training data repeatedly. Weights are adjusted after every trial using side information specifying the correct class until weights converge and the cost function is reduced to an acceptable level. An essential component of the algorithm is the iterative

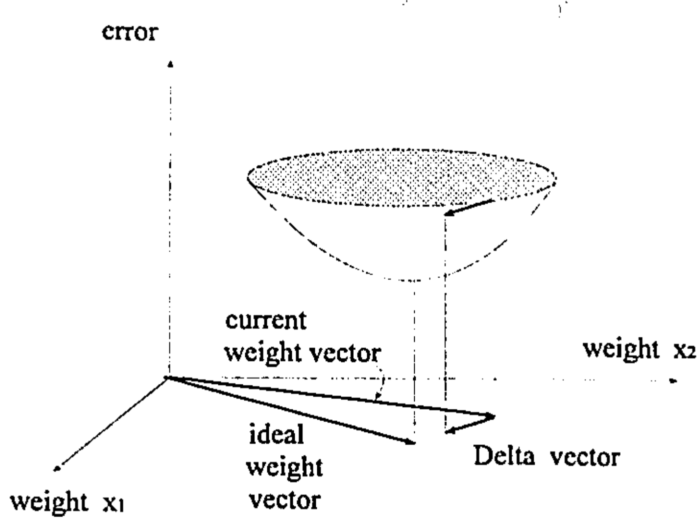


Figure 3.11 Weight Vector Changed by the Delta Rule / Caudill, 1989 /

method that propagates the error terms required to adapt weights back from the neurons in the output layer to neurons in a lower layer.

Since back-propagation refers to a general learning rule, not a specific architecture, the details of a building network vary with different practical problems. Nonlinear functions are usually used for relating the input and the output of a neuron. In many applications, a sigmoidal function is chosen. The input to a processing element, or neuron, is a weighted sum of the outputs from previous layer. It is given as

$$I = f(\Sigma\ w_i x_i)\ , \tag{3.22}$$

where x_is are the outputs of the previous layer and w_is are the weights. The output of a given neuron is a sigmoidal function:

$$f(I) = \frac{1}{1 + e^{-(I - T)}} , \tag{3.23}$$

where, I is the summed input of considered neuron and T is a simple threshold. Following through the network, this output, together with the outputs of other neurons in the same layer, is treated as the input to neurons in the next layer. The output of the network is compared with the desired value and the error is fed back to adjust the weights by applying the Delta rule (refer to Equation 3.21). If a network has more than three layers, we can then back-propagate this layer's error to the previous layer, compute weight changes the same way, and so on.

This multi-layered network with one or more hidden layers allows decision surface of arbitrary complexity and, therefore, can perform a more sophisticated separation of the feature space (compared to the linear classifier).

3.3.4 Neural Networks for Tool Condition Monitoring

The applications of neural networks into machining process and tool condition monitoring have been massively reported since 1980's. Major contributors include Rangwala and Dornfeld (1987, 1990), Chryssolouris and Domroese (1988), Elanayar *et al* (1990), Liu and Ko (1990), and other likes.

The most often used neural networks for tool condition monitoring in machining are the feed-forward networks trained by back-propagation algorithms. In most cases, a simple structure of neural network with three layers is used. The networks perform class association tasks in which the monitoring indices are presented to neurons in the input layer, and the tool conditions are assigned to neurons in the output layer. This simple structure makes it possible to apply neural networks for tool condition monitoring on the shop floor.

Rangwala and Dornfeld (1987, 1990) used acoustic emission signals and cutting forces as the inputs to the neural network. The output layer had one neuron representing the fresh or worn tool. It was stated that the various parameters in the neural network should be carefully chosen to ensure the optimum performance and efficiency of the tool wear detection system. For the purpose of tool wear detection in turning operations, relatively small size networks worked well. The results showed that neural networks possessed the ability for learning and noise suppression. High success rates (about 95 percent) were obtained for recognizing tool wear under a range of process conditions.

A similar neural network was used for drill wear recognition by Liu and Ko (1990). The results of classification were compared with those of linear discriminant function. The neural networks gave better results for recognition of drill wear states. The experiments also showed that the success rates of classification depended on the number of neurons.

Elanayar *et al* (1990) used two simple connected neural networks for machining condition monitoring. This was previously introduced in Chapter II.

Apart from the back-propagation neural networks, an adaptive resonance theory (ART2)-type neural network was also applied into the detection of tool failure in end milling (Tansel *et al*, 1993). It was reported that ART2-type neural network had a fast and continuous learning capability. The details of this neural network are not given in this thesis.

3. 4 FUZZY CLASSIFICATION

3.4.1 Basic Theory of Fuzzy Set and Fuzzy Partition

The problem of classification is to establish a function that describes the relationship between input patterns and output classes. In fact, uncertainty often exists when this relation function is built up. The main sources of uncertainty that bear on classification problem are (Bezdek, 1981): inaccurate measurements, random occurrences, and vague description. Accordingly, three different types of mathematical models are considered: deterministic, stochastic, and fuzzy. Fuzzy set theory, introduced by Zadeh (1965), is a means to deal with problems of nonstatistical uncertainty.

Sets (classes in classification problem are sets) can be represented using the object property method. If a set is well defined, its object properties provide a complete description of how to qualify for the membership. One way to describe a set is with a membership function. Let H be the set of real numbers greater than or equal to 6. We can associate "belong to H" with the number 1 and "not in H" with the number 0. Accordingly, a membership function, $u_H(r)$, may be defined:

$$u_H(r) = \left\{ \begin{array}{lll} 1 \textit{ when } & r \in H, & i.e.\ r \geq 6 \\ 0 \textit{ when } & r \notin H, & i.e.\ r < 6 \end{array} \right\}. \tag{3.24}$$

The description is illustrated graphically in Figure 3.12(a). Note that abrupt jump from 0 to 1 in the group of $u_H(r)$ at $r = 6$. This is a prominent feature of all hard sets: their boundaries are always "sharp."

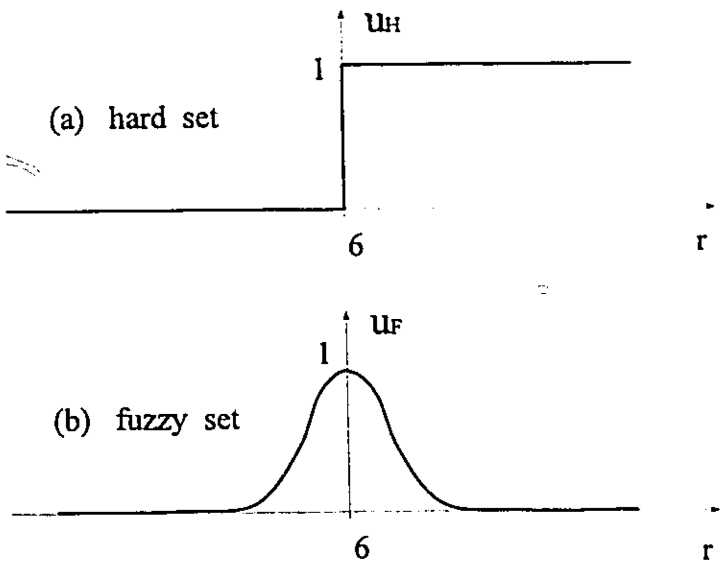


Figure 3.12 Membership Functions of Hard Set and Fuzzy Set

A fuzzy set is defined by extending the range of membership functions from the two point set {0,1} to the unit interval [0,1]. If X is any set, the u_F is a fuzzy subset of X in case u_F maps X to [0,1]. The membership function $u_F(r)$ describes the degree to which the object r belongs to the set, where $u_F(r) = 0$ represents no membership, and $u_F(r) = 1$ represents full membership. For instance, the set F containing numbers "close to 6" is a typical fuzzy set. A number of mathematical definitions can be given to this fuzzy set. One of possible mathematical expressions for this fuzzy set may be:

$$u_F(r) = e^{-(r-6)^2} . \tag{3.25}$$

The graph of this expression is shown in Figure 3.12(b).

For $r \in X$, the value $u_F(r)$ is called the grade of membership of r in F. It measures the extent to which r possesses the imprecisely defined object properties which characterize F. We notice that fuzzy set membership increases as the distance between the point and the fuzzy centre (*i.e.*, $| r-6 |$ in this example) decreases. This comparison cannot be done with a hard set. We also observe that the values of fuzzy membership functions not only order their arguments, but also have magnitudes which allow to build continuous decision thresholds. The fuzzy memberships represent similarities of the objects to imprecisely defined properties. The membership values are not affected by observations.

The use of fuzzy sets for pattern classification has been examined by many researchers. Fuzzy partitions produce soft boundaries between classes. This is a better description to most classification problems.

If X is a finite set, say $X = \{x_1, x_2, ..., x_n\}$, a collection of hard subsets $\{A_1, A_2, ..., A_c\}$ of X is a hard c-partition of X if

$$A_i \neq \emptyset , \quad \forall i ;$$

$$\bigcup_{i=1}^{c} A_i = X ; \tag{3.26}$$

$$A_j \cap A_i = \emptyset , \quad \forall j \neq i; \quad i, j = 1,2,...,c .$$

Now let u_i be the membership function of A_i, so that, for $i = 1, 2, ..., c$,

$$u_i = \left\{ \begin{array}{ll} 1; & x \in A_i \\ 0; & otherwise \end{array} \right\} . \tag{3.27}$$

Then $\{A_1, A_2, ..., A_c\}$ is equivalent to $\{u_1, u_2, ..., u_c\}$. The condition of hard c-partition is equivalent to

$$u_i \neq \emptyset , \quad \forall\, i ;$$
$$\sum_{i=1}^{c} u_i = 1 . \tag{3.28}$$

Thus, we can call either $\{A_i\}$ or $\{u_i\}$ a hard c-partition of X. The definition is:

$X = \{x_1, x_2, ..., x_n\}$ is any finite set; V_{cn} is the set of real $c \times n$ matrices;

c is an integer, $2 \leq c \leq n$. Hard c-partition space for X is the set:

$$M_c = \{ U \in V_{cn} \mid u_{ij} \in \{0,1\} \ \forall i, j; \quad \sum_{i=1}^{c} u_{ij} = 1 \quad \forall j; \quad 0 < \sum_{j=1}^{n} u_{ij} < n \quad \forall i \} . \tag{3.29}$$

Unlike the above functions which map X into $\{0,1\}$, if one or more of these functions maps X into $[0,1]$, we can define a fuzzy c-partition of X to be any set of membership functions on X into $[0,1]$, which also satisfies Equation 3.28. The definition given by Ruspini (1981) is as follows:

X is any finite set; V_{cn} is the set of real $c \times n$ matrices, c is an integer,

$2 \leq c \leq n$. Fuzzy c-partition space for X is the set

$$M_{fc} = \{ U \in V_{cn} \mid u_{ij} \in [0,1] \ \forall i, j; \quad \sum_{i=1}^{c} u_{ij} = 1 \quad \forall j; \quad 0 < \sum_{j=1}^{n} u_{ij} < n \quad \forall i \} . \tag{3.30}$$

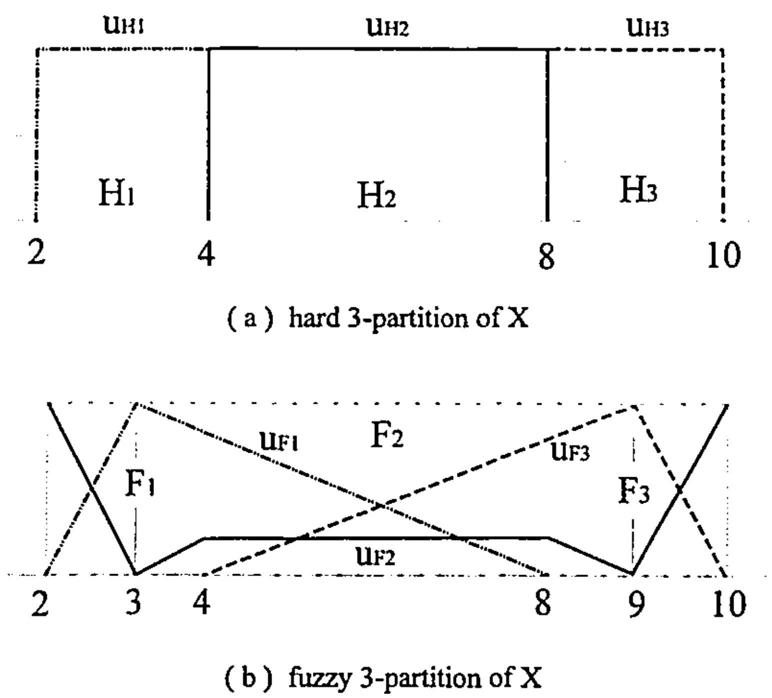


Figure 3.13 Divisions by Hard and Fuzzy Partitions

As mentioned above, the c-partition can be denoted by a $c \times n$ matrix U. Row i is the membership of every x_j in subset i and column j is the membership of x_j in each of the c subsets. The value of u_{ij} is either $\{0,1\}$ or $[0,1]$ depending upon whether it is a hard or a fuzzy c-partition.

To demonstrate hard and fuzzy c-partitions, Figure 3.13 provides an example. X is a set containing the real number between 2 and 10:

$$X = [2, 10] = [2, 4) \cup [4, 8] \cup (8, 10] = \bigcup_{i=1}^{3} C_i . \tag{3.31}$$

The three hard sets are

$$
\begin{aligned}
H_1 &= \{ x \mid 2 \le x < 4 \}, \\
H_2 &= \{ x \mid 4 \le x \le 8 \}, \\
H_3 &= \{ x \mid 8 < x \le 10 \};
\end{aligned}
\tag{3.32}
$$

and the corresponding membership functions are

$$
\begin{aligned}
u_{H1}(x) &= \left\{ \begin{array}{ll} 1; & 2 \le x < 4 \\ 0; & 4 \le x \le 10 \end{array} \right\}, \\
u_{H2}(x) &= \left\{ \begin{array}{ll} 0; & 2 \le x < 4 \\ 1; & 4 \le x \le 8 \\ 0; & 8 < x \le 10 \end{array} \right\}, \\
u_{H3}(x) &= \left\{ \begin{array}{ll} 0; & 2 \le x \le 8 \\ 1; & 8 < x \le 10 \end{array} \right\}.
\end{aligned}
\tag{3.33}
$$

The three fuzzy sets are

$$
\begin{aligned}
F_1 &= \text{"numbers close to 3 in X"}, \\
F_3 &= \text{"numbers close to 9 in X"}, \\
F_2 &= \text{"other numbers in X"};
\end{aligned}
\tag{3.34}
$$

and their possible membership functions are

$$u_{F1}(x) = \begin{cases} (x-2); & 2 \le x < 3 \\ (8-x)/5; & 3 \le x < 8 \\ 0; & 8 \le x \le 10 \end{cases},$$

$$u_{F3}(x) = \begin{cases} 0; & 2 \le x \le 4 \\ (x-4)/5; & 4 < x \le 9 \\ (10-x); & 9 < x \le 10 \end{cases}, \qquad (3.35)$$

$$u_{F2}(x) = 1 - u_{F1}(x) - u_{F3}(x); \quad 2 \le x \le 10.$$

It is evident that hard c-partitions subdivide the total membership with hard boundaries with vertical unit jumps, whereas fuzzy boundaries provide a means for continuous, partial membership allocation.

3.4.2 Criteria for Fuzzy c-Partition

As mentioned previously, three models are used for classification problems. They are deterministic, stochastic, and fuzzy. Objective functional criteria are usually used for fuzzy classification. For each class, an objective function measures the similarity of the samples in the same class. Most used criteria include density functionals, likelihood functionals and least-square functionals.

A. Density Functional Method

This method was proposed by Ruspini (1981). Let d_{jk} denote the "distance" in the real p-dimensional vector space, $\mathbb{R}^p$, between x_j and x_k; we assume that, for all x_j and x_k in $\mathbb{R}^p$, this function satisfies

$$\left\{ \begin{array}{c} d_{ik} \doteq d(x_j, x_k) \geq 0 \\ d_{ik} = 0 \Leftrightarrow x_j = x_k \\ d_{jk} = d_{kj} \end{array} \right\} \tag{3.36}$$

Functions that satisfy the above equations are called measures of dissimilarity.

One of Ruspini's classification criteria incorporating the distances $\{d_{jk}\}$ and the fuzzy c-partition of X is described as follows:

Let J_R: $M_{fco} \rightarrow \mathbb{R}^+$ be defined as

$$J_R(U) = \sum_{j=1}^{n} \sum_{k=1}^{n} \{ [\sum_{i=1}^{c} (u_{ij} - u_{ik})^2] - d_{jk}^2 \}^2 \tag{3.37}$$

where σ is a real constant and the number of classes c, $2 \leq c < n$, is fixed a priori. J_R is the classification criterion, d is the measure of dissimilarity. Ruspini interprets J_R as a measure of class quality based on local density, because J_R will be small when the terms in the functions are individually small. This, in turn, will occur when close points have nearly equal fuzzy class membership in the c u_i's in U.

Optimal fuzzy c-partitioning of X is taken as the local minima of J_R. Ruspini's algorithm is based on several general results concerning the constrained optimization problem:

$$\underset{U \in M_{fc}}{minimize} \quad \{J_R(U)\} \ . \tag{3.38}$$

Ruspini's algorithms are the first well-defined fuzzy partitioning methods with a substantial mathematical basis. However, they are hard to interpret and difficult to implement because the computational efficiency is poor and generalizations to more than $c = 2$ classes have met with little success.

B. Likelihood Functional

This method was developed by Woodbury and Clive (1974). The functional employed involves products of fuzzy memberships and discrete probabilities, resulting in a composite fuzzy-statistical criterion. Specifically, let $K_j = \{1, 2, ..., m_j\}$ for $j = 1, 2, ..., p$. It is assumed that the data set X is a subset of the Cartesian product of the K_j's over j:

$$X \subset (K_1 \times K_2 \times \dots \times K_p) = K \subset \mathbb{R}^p \ , \tag{3.39}$$

where $|K_j| = m_j$, and as usual, $|X| = n$, the number of observations in sample X. Generally, $x_{kj} \in K_j$ is the jth indicant or feature for individual x_k.

Given that

n = No. of observations in X;

n_{kj} = No. of times feature j of $\mathbf{x}_k$ appears;

n_{kj} = No. of times $t \in K_j$ appears for $\mathbf{x}_k$;

$p_{ijk} = Prob$('pure' v_i manifests outcome $t \in K_j$ in feature j),

and, $P_{wc} = \{P \in \mathbb{R}^{cpm} \mid p_{ijt} \geq 0,\ \sum_{t=1}^{m_j} p_{ijt} = 1 \quad \forall i,j,t\}$,

the objective function of Woodbury and Clive is:

Let J_{wc}: $M_{fc} \times P_{wc} \rightarrow \mathbb{R}$ be

$$J_{wc}(U,p) = \sum_{k=1}^{n} \sum_{j=1}^{p} \left(\sum_{t=1}^{m_j} n_{kjt} \{\log\left(\sum_{i=1}^{c} u_{ik} p_{kjt}\right) - \log\left[\left(\sum_{i=1}^{c} u_{ik}\left(\sum_{t=1}^{m_i} p_{kji}\right)\right]\right\} \right) \tag{3.40}$$

J_{wc} is a "fuzzy-statistical" criterion, in that it combines fuzzy memberships of observed data with probabilities of pure prototypical features. Optimal c-partitions of X are part of optimal pairs $(\hat{U}, \hat{p})$ which solve the following:

$$\underset{M_{fc} \times P_{wc}}{maximize} \ \{ J_{wc}(U, p) \} . \tag{3.41}$$

The measure of similarity in this algorithm is somewhat obscured by the combination of memberships and probabilities. Interpreting J_{wc} as a maximum likelihood criterion depends upon the probabilities, p_{kjt}, that data vector $\mathbf{x}_k$ will manifest the outcome $t \in K_j$ in feature j. The algorithm is appropriate only for data sets in $\mathbb{R}^p$, each of whose feature is a categorical variable.

C. Least-Square Functional

This method was introduced by Bezdek *et al* (1981). The criterion generalizes the within-group sum of square errors function J_w. This algorithm, also called fuzzy *c*-mean algorithm, uses interactive optimization to approximate the minima of an objective function using a particular inner product norm as a similarity measure on $\mathbb{R}^p \times \mathbb{R}^p$. The distinction between family members is the results of the application of a weighting exponent, *m*, to the membership values used in the definition of the functional.

Let $u \in M_{fc}$ be a fuzzy *c*-partition of X, and v be the *c*-tuple $(v_1, v_2, ..., v_c), v_i \in \mathbb{R}^p$. The fuzzy *c*-mean functional J_m: $M_{fc} \times \mathbb{R}^{cp} \rightarrow \mathbb{R}^+$ is defined as:

$$J_m(U, v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m (a_{ik})^2 . \tag{3.42}$$

Where, $U \in M_{fc}$ is a fuzzy *c*-partition of X; $v = \{v_1, v_2, ..., v_c\} \in \mathbb{R}^{cp}$, with $v_i \in \mathbb{R}^p$, is the class centre or prototype of class *i*, $1 \leq i \leq c$, and

$$d_{ik}^2 = \| x_k - v_i \|^2 , \tag{3.43}$$

where $\|\cdot\|$ is inner product norm metric, and $m \in [1, \infty)$.

The measure of dissimilarity is $d_{ik} = \| x_k - v_i \|$ the distance between each data point x_k and a fuzzy prototype v_i. The squared distance is weighted by the *m*th power of the membership of data **x** in class *i*. Since each term of J_m is proportional to $(d_{ik})^2$, J_m is a squared error criterion, and its minimization produces fuzzy classification (matrix U) that is optimal in a generalized least squared error sense:

$$\underset{M_{fc} \times \mathbb{R}^{cp}}{\text{minimize}} \quad \{J_m(U, v)\} \tag{3.44}$$

The fuzzy c-mean functional extends the classical WGSS (the Within-Group Sum-of-Squared-error) criterion J_w, which is a very popular and well-studied basis for hard classification. From the mathematical point of view, J_m is intimately related to the Hilbert space structure of $\mathbb{R}^p$ and is thus tied to a profound mathematical structure.

3.4.3 Fuzzy Neural Networks

Fuzzy classification and neural networks are two powerful and convenient tools for pattern classifications. The combination of fuzzy logic with neural networks is interesting, since these two approaches generally deal with the design of "intelligent" systems from different angles. Neural networks enable one to deal with large amounts of sensor data simultaneously using simple processing elements, and fuzzy logic provides a structural framework that utilizes these simply processed results. Two possible ways to merge these these two technologies are: (i) fuzzification of conventional neural network architectures — interconnections with fuzzy relationship or fuzzified processing elements, and (ii) the use of neural networks as tools in fuzzy models.

An example of recent research on fuzzy neural networks is the Fuzzy Min-Max Classification Neural Network (Simpson, 1992). Hyperboxes defined by pairs of min-max points, and their corresponding membership functions are used to create fuzzy subsets of the

n-dimensional pattern space. The min-max hyperbox $B_j = \{ V_j, W_j \}$ in $\mathbb{R}^3$ is shown in Figure 3.14(a). The min and max points are all that are required to define the hyperbox. A membership function is associated with the hyperbox and determines the degree to which any point $\mathbf{x} \in \mathbb{R}^3$ is contained within the box. A collection of these boxes forms a pattern class. The membership function for each hyperbox fuzzy set describes the degree to which a input pattern fits within the hyperbox:

$$b_j(A_h) = \frac{1}{2n} \sum_{i=1}^{n} [\max(0, 1 - \max(0, \gamma \min(1, a_{hi} - w_{ij}))) + \max(0, 1 - \max(0, \gamma \min(1, v_{ji} - a_{hi})))] . \tag{3.45}$$

Where $A_h = (a_{h1}, a_{h2}, ..., a_{hn}) \in I^n$ is the hth input pattern, $V_j = (v_{j1}, v_{j2}, ..., v_{jn})$ is the min point for B_j, $W_j = (w_{j1}, w_{j2}, ..., w_{jn})$ is the max point for B_j, and γ is the sensitivity parameter that regulates how fast the membership values decrease as the distance between A_h and B_j increases. An example of fuzzy min-max hyperboxes along the boundary of a two-class problem is illustrated in Figure 3.14(b).

In the fuzzy min-max neural network, each neuron in the hidden layer represents a fuzzy set of hyperbox where the input to output connections are the min-max points. The transfer functions are the hyperbox membership functions defined by Equation 3.45. The connections are adjusted using a proposed learning algorithm (Simpson, 1992). The outputs at the output layer represent the degree to which an input pattern fits within corresponding classes. If a soft decision is required, the outputs are utilized directly. If a hard decision is required, the output node with the highest value is located.

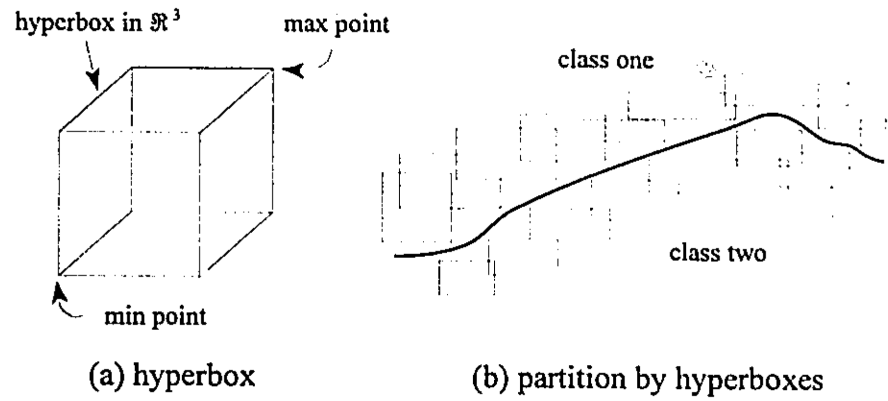


Figure 3.14 The Min-Max Fuzzy Hyperbox for Classification / Simpson, 1992 /

3.4.4 Fuzzy Classification for Tool Condition Monitoring

Metal cutting is a very complex process. Cutting conditions (cutting speed, feed, depth of cut) and tool conditions (chipping, wear, chatter) will greatly affect process parameters such as forces, torques, and the power. The purpose of tool condition monitoring is to identify the tool working status from the information acquired from the cutting process, which are mostly force and vibration signals.

Conventional classification methods make an effort to correlate precisely the tool conditions with the monitoring indices. However, the data substructure for tool condition monitoring in machining is always mixed. A fuzzy expression of operating parameter-tool condition relation is much more suitable in practice. The description that the tool is "slightly" or "severely" worn has more sense for on-line tool condition monitoring. The membership grade can be used to measure the uncertainty of tool conditions.

A. Linear Fuzzy Equation

The basic idea of this approach is to use a simple fuzzy model for tool condition monitoring. In this model (Du *et al*, 1992), the tool conditions are the input and the monitoring indices are the output. The relationship between the input and the output is described by a linear fuzzy equation:

$$r = Q \circ p . \tag{3.46}$$

Where, **r** represents the fuzzy degree of the monitoring indices (**x**), **p** represents the fuzzy degree of the tool conditions (**h**), **Q** is the fuzzy relationship function, and the symbol "∘" is the fuzzy operator.

Given a process that has c different tool conditions (*i.e.*, $\mathbf{h} = \{h_1, h_2, ..., h_c\}$) and m monitoring indices (*i.e.*, $\mathbf{x} = \{x_1, x_2, ..., x_m\}$ and $h(\mathbf{x}) \in \{h_1, h_2, ..., h_c\}$); then, **r** is an m-dimensional vector, **p** is a c-dimensional vector, and **Q** is an $m \times c$ matrix that describes the fuzzy relationship between the tool conditions and the monitoring indices. Equation 3.46 implies that the changes in the magnitude of the monitoring indices are correlated to the changes in tool conditions by a linear fuzzy equation.

Similar to other decision making methods, the linear fuzzy equation approach consists of two phases: learning and classification. The learning process is carried out to establish the fuzzy relationship based on the experimental samples obtained from cutting tests. This method determines the fuzzy relationship function **Q** based on the *possibility distribution* (frequency of occurrence) and the *probability distribution* (strength of support) of the learning samples.

Suppose that there are n learning samples, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ obtained from the experiments. Each sample consists of m elements: $\mathbf{x}_k = \{x(k, i), i = 1, 2, \ldots, m\}, k = 1, 2, \ldots, n$. Using i to label the monitoring indices, j to label the tool condition, k to label the learning samples, and Equation 3.46 can be rewritten in a matrix form:

$$\begin{bmatrix} r_1 \\ r_2 \\ \cdots \\ r_m \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1c} \\ q_{21} & q_{22} & \cdots & q_{2c} \\ \cdots & \cdots & \cdots & \cdots \\ q_{m1} & q_{m2} & \cdots & q_{mc} \end{bmatrix} \circ \begin{bmatrix} p_1 \\ p_2 \\ \cdots \\ p_c \end{bmatrix} \tag{3.47}$$

or

$$r_i = q_{i1} \otimes p_1 \oplus q_{i2} \otimes p_2 \oplus \ldots \oplus q_{ic} \otimes p_c, \quad i = 1, 2, \ldots, m \tag{3.48}$$

where, $\otimes$ is the fuzzy multiplication operator and $\oplus$ is the fuzzy addition operator.

Since the fuzzy operator is a linear operator, the ith monitoring index will be affected only by $q_{ij}, j = 1, 2, \ldots, c$. The element q_{ij} is the fuzzy relationship function that relates the ith monitoring index to the jth tool condition, and it can be described by a set. For example, $S_i = \{x(1, i), x(2, i), \ldots, x(n, i)\}$ is the set which contains the ith monitoring index of all the learning samples. Furthermore, we will assign that

$$\begin{aligned} x_{i,\max} &= \max \{ x(1,i), x(2,i), \ldots, x(n,i) \}, \\ x_{i,\min} &= \min \{ x(1,i), x(2,i), \ldots, x(n,i) \}. \end{aligned} \tag{3.49}$$

Divide the interval between $x_{i,max}$ and $x_{i,min}$ into L evenly distributed sub-intervals. Each interval , denoted by $v\ (i,\ k)$, $k = 1, 2, ..., L$, is defined as follows:

$$v(i,k) = [x_{i,\min} + (k-1)\,\Delta x,\ x_{i,\min} + k\,\Delta x]\ , \tag{3.50}$$

where $\Delta x = (x_{i,max} - x_{i,min}) / L$. Then, q_{ij} can be represented by a fuzzy set with L-elements:

$$q_{ij} = \{\ v(i,k)\ :\ q(i,j,k),\quad k = 1,2,...,L\ \}. \tag{3.51}$$

Here, the fuzzy degree, $q(i,j,k)$, is determined by the possibility and the probability distributions of the learning samples.

The possibility distribution, denoted by $f(i,j,k) = C_{ijk} / C_{ik}$, represents how the ith monitoring index is distributed in the kth interval when the jth tool condition occurs compared with the other tool conditions. C_{ijk} is the number of samples (index i) that belong to tool condition j and are located inside the kth sub-interval, and C_{ik} is the number of samples (index i) located inside the kth sub-interval.

The strength of support $S(i,j,k) = C_{ijk} / C_{ij}$ defines how the ith index is distributed in the kth interval when the jth tool condition occurs. Where, C_{ij} is the number of samples (index i) that belong to class j.

Combining $f(i,j,k)$ and $S(i,j,k)$, the fuzzy relationship function is

$$q(i,j,k) = \frac{f(i,j,k) + S(i,j,k)}{2}\ , \tag{3.52}$$

and $q(i,j,k)$s are the elements of the matrix **Q** in Equation 3.46.

When the value of the fuzzy relationship function is determined, the classification can be made by the "summation" solution defined as follows:

$$p_j = \sum_{i=1}^{m} min \{ \bar{q}_{ij} , r_i \} , \tag{3.53}$$

where, $\bar{q}_{ij} = q(i,j,k)$, $i = 1, 2, ..., m$, and $j = 1, 2, ..., c$.

B. Fuzzy Decision Tree

The problem of decision making deals with finding some criteria that relate the feature space to the classification space, then to map a vector in feature space into the classification space. Usually, matrices are used for this mapping, like the fuzzy relationship function **Q** in Equation 3.46 obtained from known **r** and **p**. We will refer to this type of decision making as matrix-type decision making. An alternative approach for relating the feature and the classification spaces is to use a decision tree.

A tree structure means that the data are organized so that items of information are linked by branches. A tree is a finite set of one or more nodes such that: (i) there is a specially designated node called the root; (ii) the remaining nodes are partitioned into $n \geq 0$ disjointed sets T_1, T_2, ..., T_n where each of these sets is a tree (Horowily, 1976).

The tree structure can be used for decision making. At each node, input data are partitioned into two or more groups containing the data from the same category. These categories may be of one class or a combination of different classes. Further searches are performed through all the branches until a final result is obtained at a leaf of the tree where only one class is assigned.

The fuzzy decision tree method for tool condition monitoring combines a decision tree with the fuzzy classification (Li *et al*, 1992). In contrast to matrix-type decision making, tree-type decision making divides the feature space into several subspaces with fewer dimensions. It gradually increases the precision of the decision. As an example of partitioning the feature space made by a decision tree for a two-class problem (Class A and Class B), a portion of the samples from one class (A or B) can be isolated from the others initially by an index I_i. Then, another portion of the samples from one class is partitioned by index I_j, and so on. The whole space is partitioned in steps by I_i ($i = 1, 2, ..., m$) into the regions which contain only samples from the same class. Because fuzzy classification is introduced, fuzzy decision tree generates soft boundaries for neighbouring classes at each node and the uncertainty measure of the decision is also provided. In general, a node in a fuzzy decision tree is a statement such as "if C_i happens, d_j is true which is measured by the membership grade μ_{ij}." It can be represented by the following form:

$$C_i\text{: } d_j \text{ is } \mu_{ij}. \tag{3.54}$$

where, C_i, $i = 1, 2, ..., M$ (M is the number of nodes in the tree), is a condition statement such as "if the membership function of the input **x**, $u_k(\mathbf{x})$, is larger than a particular threshold value;" d_j, $j = 1, 2, ..., c$ (the number of classes), is the conclusions at the ith node; and μ_{ij} is the membership grade for this decision. Note that d_j may be either one of, or any combination of, the decision sets in the classification space. Classification using the fuzzy decision tree may be performed in several steps. For example, a tool may be classified as "either sharp or worn," then further classification specifies the final result as "slightly worn."

In the learning phase, a fuzzy decision tree is constructed based on available learning samples. The proposed method constructs the fuzzy decision tree by partitioning the learning samples using the recursive procedure below:

```
p := 1;
X_p := X;
repeat
    (A_p, B_p) := maximum_partition;
    set_node_pointer;
    f_p := d_j / n_j;
    c_p := means(A_p, B_p);
    p := p+1;
    X_p := B_p-1;
until X_p belongs_to_the_same_condition;
```

The key operation of the procedure is the "maximum partition:" $X_p = A_p + B_p$. X_p is the learning data set used at node p, and A_p is the set which contains the samples that belong to the same tool condition (say h_j). This partition separates the maximum number of samples in A_p. To do this, all the monitoring indices are examined against all the tool conditions. That is, for all monitoring indices I_i, $i = 1, 2, ..., m$, we seek all intervals that contain the samples from the same tool condition. Accordingly, the partition which separates the maximum number of samples is chosen as the maximum partition. Note that the maximum partition may not completely separate the samples of the same tool condition from the other samples since

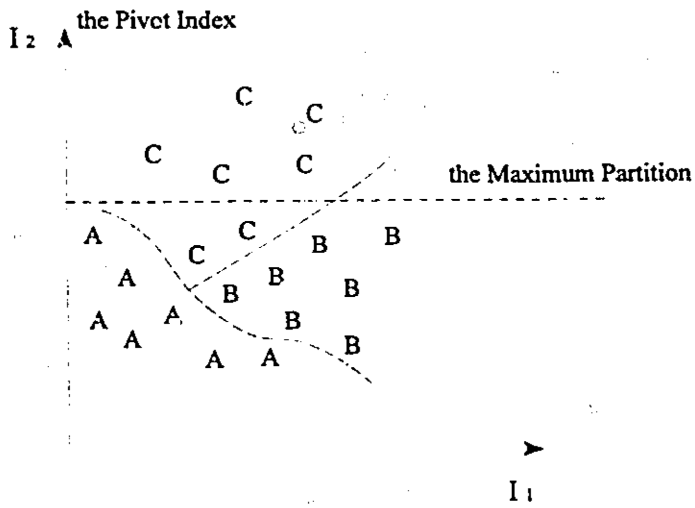


Figure 3.15 Example of the "Maximum Partition"

samples from different tool conditions are typically overlapped. The monitoring index associated with the partition is defined as the *pivot index* and is used as part of the information at a node of the decision tree.

Figure 3.15 gives an example of the maximum partition of a 2-index, 3-class problem. By studying the distribution of the samples from classes A, B and C, we notice that I_2 gives the maximum partition of the learning set for class C with respect to the other classes. Here, I_2 is chosen as the pivot index at this node for the maximum partition.

The maximum partition generates a left son A_p, which is a leaf of the tree indicating one tool condition, and a right son $B_p = X_p - A_p$, which is either a leaf or a sub-tree. The father, left and right sons of the node are represented by the node pointers. Also, the distribution of the learning samples is used to measure the strength support of the partition.

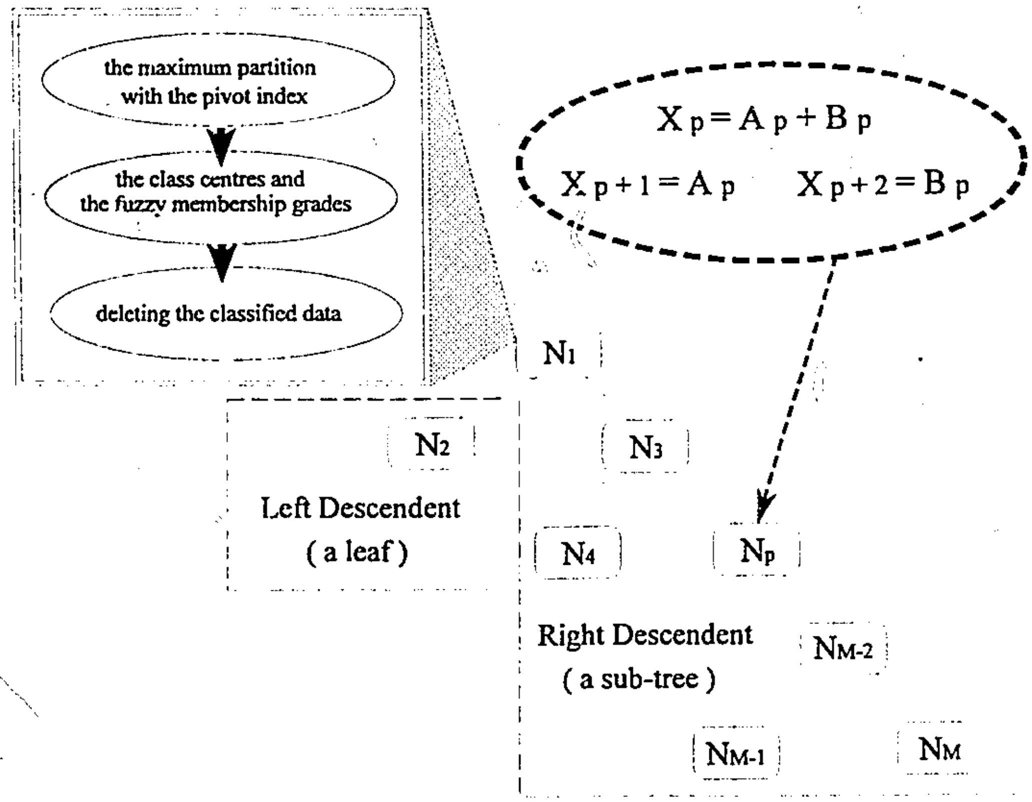


Figure 3.16 Building the Fuzzy Decision Tree through Learning

Suppose that at node *p*, index *i* is used to separate some of the samples of class *j* from the other samples. Then, the uncertainty measure f_p is defined by:

$$f_p = \frac{d_j}{n_j} \tag{3.55}$$

where, n_j is the total number of samples that belong to the *j*th tool condition and d_j is the number of samples in A_p. The fuzzy centre of the class considered is designed to form a soft boundary between A_p and B_p. It is calculated as the statistical mean of the samples.

The above operation is then repeated for the new $X_{p+1} = B_p$ until all the samples in X_{p+1} belong to the same tool condition. As a result, at each step of the operation, a new node is generated and added to the decision tree. Accordingly, a binary fuzzy decision tree is built. This procedure is illustrated in Figure 3.16. In the given tree, leaves N_2, N_4, ..., N_{M-1}, N_M contain the samples which belong to the same class respectively. Nodes N_3, ..., N_p, ..., N_{M-2} contain the samples from B_p ($p = 1, ..., (M-1)/2$) generated by the maximum partitions.

For classification, a recursive procedure called "tree traversal" is used to identify the class to which a given sample belongs to. At node p, the calculation of the membership grade of a sample to class i is performed by the following equation (fuzzy c-mean method, Bezdek, 1987):

$$U_{ip} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{ip}}{d_{jp}}\right)^{\frac{2}{M-1}}} \tag{3.56}$$

C. Other Fuzzy Classification Methods

Monitoring drill states has also been done by fuzzy c-means algorithm (Li and Wu, 1988). The fuzzy classification is reported to be more adequate than crisp classification methods for drill wear detection. Experimental and simulation results show that the drill wear conditions could be represented by four grades: initial, small, normal, and severe. The detection of a large membership grade on the wear state "severe" was proposed as an indicator for drill replacement.

Other applications of fuzzy classification to the tool condition monitoring in machining included tool wear monitoring in diamond turning by fuzzy *c*-means method (Ko and Cho, 1994), the fuzzy decision system used for tapping process diagnosis (Chen, 1993), and so on.

3.4.5 Improvements Needed for Fuzzy Tool Condition Monitoring

The fuzzy set theory has been used to develop decision making strategies for automated tool condition monitoring in machining. Two algorithms, the linear fuzzy equation and the fuzzy decision tree, were developed and verified by cutting tests in turning. The results have shown superior performances of the fuzzy classifications in these tests when compared to other classic pattern recognition methods such as the *k*-mean, Fisher's method, and the nearest neighbour method. These two approaches also result in better classification results than the fuzzy *c*-mean method (Du *et al*, 1992 and Li *at al*, 1992).

The linear fuzzy equation method uses a matrix to describe the relationship between the monitoring indices and the tool conditions. Because decision trees are more flexible than matrix approaches, the fuzzy decision tree method out-performs the linear fuzzy equation method. The disadvantage of using matrix-type decision making is also seen in considering the time spent for the learning and the classification. It takes much more time to do matrix calculations.

In constructing the fuzzy decision tree for tool condition monitoring in machining, the maximum partition algorithm generates the node holding the samples from only one tool condition to create a leaf. The other samples are put into another node which is usually a sub-tree. Because of this structure, the classification paths to the leaves in lower levels are much longer compared those close to the root. A single-class-partition in the maximum partition also makes a large number of tree nodes. Thus, the classification time is increased.

3.5 SUMMARY

Decision making strategies are one of the major issues in the development of automated machining process and tool condition monitoring. A decision making in monitoring is based on the relationship between the process/tool conditions and the feature-bearing signals (monitoring indices). This relationship can be described in a number of ways such as models, patterns, expert systems, neural networks, and fuzzy systems.

Among the large number of decision making methods that have been developed, statistical pattern recognition, neural networks and fuzzy classification are very interesting aspects in the development of automated/intelligent tool condition monitoring in machining. They have been applied successfully to many cases of monitoring tasks in turning, milling, drillling and other metal cutting processes.

Automated tool condition monitoring in machining includes two parts. They are feature extraction and decision making. Feature extraction involves the experiences and the knowledge of the metal cutting process. The principle of the metal cutting mechanics has to be understood for the selection of the measured process signals. Canonical analysis and variable selection are two major methods for the feature extraction in the signal processing.

In classification, Bayes' rule is essential for statistical decision making. The function used for relating the monitoring indices and the tool conditions can be linear, polynomial, or in other forms.

Neural networks and fuzzy logic are powerful tools for automated/intelligent tool condition monitoring in machining. Neural networks provide the possibility of dealing with large amount of sensor data simultaneously using simple processing elements, and fuzzy logic gives a structural framework that utilizes these simple processed results for the uncertainty. Back-propagation is the most widely used method for training neural networks. The back-propagation neural networks are applied to tool condition monitoring by several examples. It is determined that the simple neural networks with one or two hidden layers will work better for machining process and tool condition monitoring.

The applications of fuzzy logic in machining process and tool condition monitoring include fuzzy *c*-mean algorithms, the fuzzy liner equation, and the fuzzy decision tree. Fuzzy classifications provide soft boundaries between the classes, which give better descriptions to the class overlaps. The measurements of uncertainty can be calculated by the fuzzy membership grades through a membership function. Fuzzy expressions of the relations between the operating parameter and the tool conditions are much more suitable for metal cutting processes because of the inter-effects between cutting parameters and the overlaps in the tool conditions.

Linear fuzzy equation and the fuzzy decision tree are two new contributions to the decision making algorithms in tool condition monitoring. Some improvements are still needed for further developments of automated machining tool condition monitoring system.

CHAPTER IV

THE MULTIPLE PRINCIPAL COMPONENT FUZZY NEURAL NETWORKS FOR TOOL CONDITION MONITORING

4.1 INTRODUCTION

The tasks of an automated tool condition monitoring system involve the ability to recognize tool conditions by analyzing measured cutting process parameters such as forces and vibrations. This ability is based on the accumulation of useful information from related laws of physics and operators' experiences. In building automated/intelligent tool condition monitoring systems, some basic functions have to be considered:

(1) Fusion of multiple sensors;

(2) Learning or training strategies for the monitoring system;

(3) Knowledge updating techniques; and

(4) Description of the imprecision in tool conditions for various cutting conditions.

With the increasing needs for effective and robust automated machining process and tool condition monitoring, a significant amount of research work has been performed to find decision making strategies. One of the most recent approaches to machine intelligence is "soft computation" (Zadeh, 1993). This approach deals with approximation and dispositionality in classification problems. The principal constituents of soft computation include fuzzy logic for imprecision in the acquired data, neural networks for learning, and probability reasoning for uncertainty. These three components are usually overlapped. The "soft computation" is easily implemented by fuzzy neural networks.

In this chapter, a new approach for automated tool condition monitoring in machining under varying cutting conditions by fuzzy neural networks is proposed. The system is named *the Multiple Principal Component (MPC) Fuzzy Neural Network*. It is based on three major components of soft computation, which were mentioned previously. Principal component analyses are conducted in multiple directions for feature extraction and optimum class partitions. Fuzzy neural networks are constructed with fuzzy classification at the neurons and the fuzzy interconnections. The MPC fuzzy neural networks are built through training with the learning data obtained from cutting tests performed within a reasonable range of cutting conditions. The strategies for supervised learning, unsupervised learning, and knowledge updating of the MPC fuzzy neural network are developed. The three major subjects, along with sensor fusion, are supervised classification, unsupervised classification, and knowledge updating of the system.

4. 2 STRUCTURE OF THE MPC FUZZY NEURAL NETWORKS

4.2.1 Partial Least Square Methods for Sensor Fusion

In general, the signals coming from only one sensor are typically insufficient to give enough information for machining process and tool condition monitoring. The use of several sensors at different locations simultaneously is proposed for data acquisition. Signals from different sources are integrated to give the maximum information needed for monitoring and control tasks. *Sensor Fusion* generally covers all the issues of linking sensors of different types together into one underlying system architecture (McClelland, 1988). The most significant advantage of sensor fusion is its enriched information for feature extraction and decision making strategies. It provides more reliable data for the decision making process with low uncertainty which may be created by the inherent randomness or noise in the sensor signals.

As discussed previously, one of the classification criteria is the "principal component analysis" (James, 1985). Assume, for a moment, that there are two fairly compact and distinct classes. The combined covariance matrix describes the shape of the total sample distribution. Specifically, the eigenvector corresponding to the maximum eigenvalue gives the direction

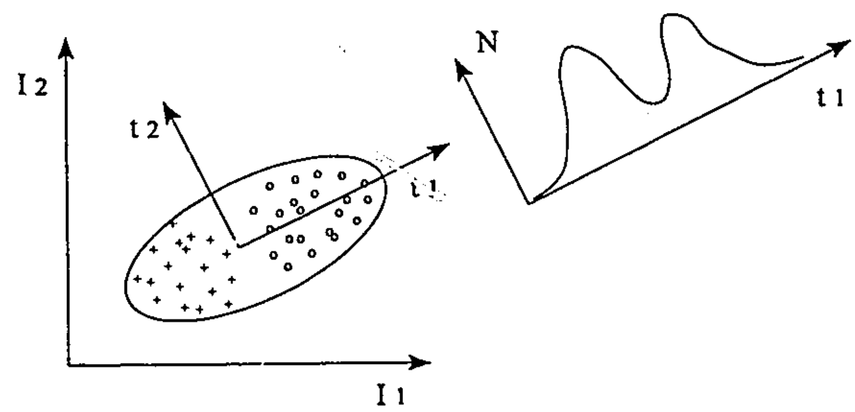


Figure 4.1 Principal Component Analysis

of the maximum variance. As suggested in Figure 4.1 for a two-dimensional space, this direction may indicate a discriminant for the two classes. If we take t_1 as the maximum separating direction and choose a threshold, θ, as the possible discriminant, we can specify the classes A and B with the following:

$$\begin{aligned} x &\in \textit{Class } A, && \textit{if } t_1^T x < \theta \,; \\ x &\in \textit{Class } B, && \textit{otherwise} \,. \end{aligned} \tag{4.1}$$

The principal component analysis takes into account the directions in which the measurement vectors, xs, have the largest covariance with the class (tool condition) vector, and ensures that this direction (eigenvector) is used for the classification. Unfortunately, principal component analysis is unable to deal with complicated problems, such as those involving $c > 2$ classes or cases with non-spherical distribution in the principal direction. The sharp boundary (threshold, θ) distinguishing two classes in the principal component direction is also not very precise in describing a practical problem such as tool condition classification.

The Partial Least Square (PLS) methods can be used for principal component analysis and feature extraction. It provides a systematic means for integrating the information from multiple sensors. The PLS methods have been used in many applied sciences (for example, refer to Höskuldsson, 1988). Accordingly, only a brief recapitulation is given here.

We seek to estimate c outputs from n variables. It should be noted that c is also the number of classes considered (*i.e.*, dimension of classification space). If $\mathbf{x}_i$s, $i = 1, 2, ..., N$, are the measurement vectors, and **K** is a matrix, then:

$$y_i = K\, x_i \ , \tag{4.2}$$

where, $\mathbf{y}_i$ is the c-dimensional tool condition vector of the ith sample, and is obtained during the training by using 1 for the element corresponding to the known class and 0s elsewhere.

Using N samples of **y** and **x**, we have:

$$Y = X\, K^T \ , \tag{4.3}$$

where **Y** is $N \times c$ and **X** is $N \times n$ matrices respectively. The ordinary least squares solution for K is:

$$K_{LS} = Y^T X\, [X^T X]^{-1} \tag{4.4}$$

To reduce the dimension of **X**, the PLS method can be used to transform the measurement vectors to their principal components. The PLS method calculates m largest eigenvalues from $\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}$, and gives a new expression for x with linear combination of those m corresponding eigenvectors

$$x = t_1 p_1 + t_2 p_2 + \ldots + t_m p_m \, , \tag{4.5}$$

where $\mathbf{p}_i$ is the eigenvector corresponding to the ith largest eigenvalue. Constructing an $n \times m$ matrix **P** whose ith column is $\mathbf{p}_i$, we have the new feature vector **t** (since **P** is column orthogonal):

$$t = P^T x \, . \tag{4.6}$$

Compared to the original measurement vector **x**, the dimension of **t** is significantly reduced. Consequently, **y** can be derived as:

$$y = K x = K P t \, . \tag{4.7}$$

Let $\mathbf{K}_t = \mathbf{K}\,\mathbf{P}$, then **y** is given:

$$y = K_t \, t \, . \tag{4.8}$$

And, the least square solution for $\mathbf{K}_t$ is:

$$(K_t)_{PLS} = Y^T \, T \, [T^T \, T]^1 \, , \tag{4.9}$$

where, **T** is $N \times m$ matrix whose rows are $\mathbf{t}^T$.

Using the PLS methods, the largest eigenvalues of X^TYY^TX are easily computed. The principal component analysis is complemented. This approach takes into account the multiple directions in the measurement vector, x, which have the largest covariance with the class (machining process and tool conditions) vector, y, and ensures that these directions (eigenvectors, or principal components) are used.

4.2.2 Neural Networks and Knowledge Learning

Learning refers to the processes which build the monitoring system in a given structure with information from the learning data. In addition, some logic rules are also created, which determine the data processing and govern the relationship between the processing elements. During the learning phase, a limited amount of learning data is used to adjust the parameters of the monitoring system. The trained monitoring system uses the stored knowledge gained from the learning to classify the data in classification of the tool conditions.

If the sampled data for training the system are labelled with the class to which a sample belongs, the decision making is performed with *a priori* knowledge. This is simply called *pattern classification*, or *supervised classification*, and it is a common problem in automated tool condition monitoring in machining. When the training samples are collected, the tool conditions related to each training sample are provided to give the necessary information.

Unlike pattern classification, which is performed with *a priori* knowledge (labelled samples), *pattern clustering*, or *unsupervised classification*, deals with the pattern recognition with unlabelled samples. The pattern clusters are formed according to some predefined similarities, *i.e.* a cluster is defined as a set of samples which are similar to each other.

Knowledge updating, or *self-learning*, refers to processes in which the structure and the parameters of a monitoring system are modified according to the new information about the classification. This ability is essential for an automated tool condition monitoring system. Classification results should be checked on-line to ensure the system gives correct results. If the results are not correct, the system should be retrained or modified. Also, the new information about the cutting process may be available and it can be used for improving the monitoring performance.

Neural networks are computing systems made up of a number of simple, highly interconnected processing elements. They provide the capability of self-learning for the system. Using neural networks, simple classification algorithms can be used and the system parameters are easily modified. One major characteristic of building neural networks is the training time. Training times are typically longer when complex decision regions are required and when networks have more hidden layers. As with other classifiers, the training time is reduced and the performance improved if the size of a network is tailored to be just large enough to solve a problem but not so large that too many parameters need to be estimated with limited training data. Besides the size of a neural network, the interconnections between the neurons also affect the training time. Partial interconnections will reduce the training time as well as the time for the classification.

4.2.3 Fuzzy Classification and Uncertainties in Tool Condition Monitoring

During machining, cutting conditions (*e.g.*, cutting speed, feed, depth of cut) as well as tool conditions (*e.g.*, tool wear) significantly affect the process parameters such as cutting forces and vibrations, which are usually used as the input signals to a monitoring system. Deterministic models which attempt to describe the relationship between the tool conditions and the various measured parameters are typically valid for a limited range of cutting conditions. The fuzzy classification can be used to describe the uncertainties and the overlapped relationship of the tool conditions and the monitoring indices. It is more suitable to say that the tool wear is "small" or "large" in practice. Briefly, the fuzzy expression of a tool condition, A, is a fuzzy concept. It is:

$$A = \{ x \mid \mu_A(x) \} , \tag{4.10}$$

where x is the value of A, and $\mu_A(x)$ is a fuzzy measure, also known as the membership function. $\mu_A(x)$ is a monotonous function, and $0 \leq \mu_A(x) \leq 1$. The function increases with respect to the decrease of the uncertainty of A. If B is also a fuzzy set and is more uncertain than A, then:

$$\mu_A(x) > \mu_B(x) . \tag{4.11}$$

This might be interpreted as "the membership grade of small tool wear is greater than that of large tool wear." The conclusion about a tool condition comes with the measurement. The

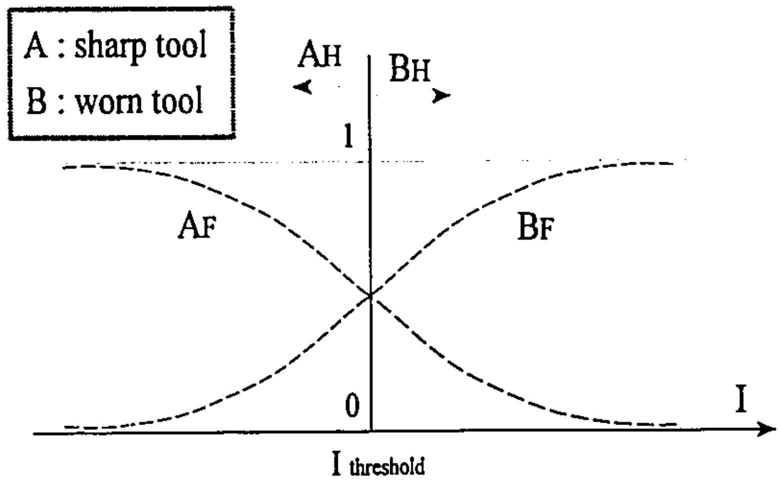


Figure 4.2 Soft Boundaries in Fuzzy Classification

fuzzy representation of the tool conditions in machining has its advantages. The concept of fuzzy decision making in machining tool condition monitoring is illustrated in Figure 4.2. Where, A_H and B_H are categories classified by the hard decision, while A_F and B_F are classified by the fuzzy decision. $I_{threshold}$ is used for hard decision making and forms a sharp edge in the boundary. Fuzzy decision making partitions the classification space with the continuous "soft boundaries." The use of fuzzy classification for the tool monitoring in machining is particularly attractive in situations where a few different tool conditions result in almost identical changes in sensor outputs.

4.2.4 Construction of the MPC Fuzzy Neural Networks

The Multiple Principal Component Fuzzy Neural Networks are constructed based on the idea of "soft computation." Neural networks, fuzzy logic and statistical reasoning are employed for the construction of the system. The neural network makes up the mainframe of the system. Simple classification procedures can be implemented at individual processing elements (neurons). The interconnections between neurons in the network communicate the information between neurons and make it possible to solve complex classification problems. The fuzzy classification at neurons deals with uncertainties in the classification. The measurement of those uncertainties at one neuron is also taken as a part of the input information to the other related neurons. Statistical reasoning is used in the learning procedure for the feature extraction and the partition strategies.

For conventional neural networks, each of the processing elements (at the input, output, and hidden layers) is always connected to every single processing element in the neighbouring layers. In general, a processing element could be connected only to those "closely related" neurons in nearby layers and even directly to those in other distant layers. This is a special case for neural networks. We will denote them as partially connected neural networks and they provide "shortcuts" to the decision making.

Decision tree classifiers are hyperplane classifiers that have been developed extensively in the literature (Lippmann, 1989). They can be regarded as a type of partially connected neural networks since each node in the tree is connected to only its "father" and "sons." They require comparatively less computation for the classification. They can be implemented using parallelism from decision region by performing simple, easily understood operations on the input features. They can use continuous-valued inputs or discrete symbolic inputs. Their size can be easily adjusted to match their complexity based on the amount of training data provided. At each node in a tree, the input data are partitioned into two or more groups containing the data from the same category. These categories may be one class or a combination of different classes. Further searches are performed through all the branches until the final result is obtained at a leaf of the tree where only one class is assigned. In contrast to matrix-type decision making, which gives the final decision in one step after computations in the hidden layer, tree-type decision making divides the feature space into several sub-spaces with many fewer dimensions, and gradually increases the precision of a decision.

The training procedures to build decision trees do not minimize a global cost function directly but gradually build a tree by minimizing the local cost function at each stage of the training. The training data are sorted or ordered separately along each input dimension and a cost function is computed for all possible splits of the training data. This speeds up the training because only as many features as input dimensions are involved and the local cost functions are simple. A comparison between a binary tree and a back-propagation classifier showed small differences in error rates but greatly reduced training times with the binary tree classifier (Fisher and McKusick, 1988).

The problem of decision making deals with finding some criteria to relate the feature space to the classification space, then to map a variable in the feature space into the classification space. Typically, parameters (weights) in a matrix form are used for this mapping, such as the estimation of weights in a linear classifier or in a back-propagation neural network. In more sophisticated implementations, multi-layered neural networks are employed, which consist of nonlinear connections between the inputs and the outputs.

As an alternative to conventional neural networks, we propose here a partially connected, fuzzy neural network approach for automated tool condition monitoring in machining. Different from matrix-type decision making, a tree structure is used for reducing unnecessary connections between elements in the input and the output layer. The fuzzy classifications are used in the neural networks to provide a comprehensive solution for certain complex problems. The neural network that utilizes fuzzy classification is shown in Figure 4.3. The input layer, $F_A = (a_1, a_2, ..., a_m)$, has m processing elements, one for each of the m dimensions of the input pattern $\mathbf{x}_k$. The hidden layer of the network, F_B, consists of the neurons that use the fuzzy classification to separately address the subsets of the original data set while invoking necessary information from other neurons. The probability distribution and the membership function are used for interconnections within the hidden layer and the connections to the output layer. The neurons of the output layer, F_C, represent the degrees to which the input pattern $\mathbf{x}_k$ fits within the each class. There are two possible ways that the outputs of F_C class nodes can be utilized. For a soft decision, outputs are defined with the fuzzy grades. For a hard decision, the F_C nodes with the highest membership degree is located (a "winner-take-all" response). The connections within the hidden layer are not from one element to every one in the neighbouring layers. The structure depends on the training data and is created through the training process. These partial connections result in the simpler and faster training and classification.

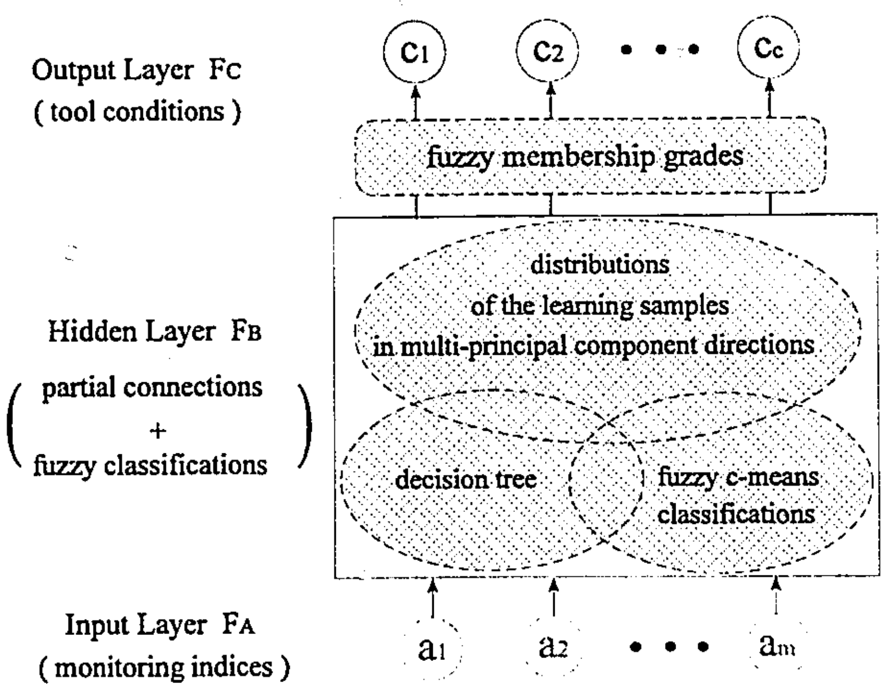


Figure 4.3 The Multiple Principal Component (MPC) Fuzzy Neural Network

Similar to other decision making strategies, the proposed fuzzy neural networks for automated tool condition monitoring have also the learning and the classification procedures. The learning is used to construct the fuzzy neural networks with fuzzy classifications and the decision tree, from available learning samples, while the classification is used to estimate the most likely decision for the given samples.

4.3 SUPERVISED CLASSIFICATION OF THE MPC FUZZY NEURAL NETWORKS

4.3.1 Learning

In the learning procedure, the hidden layer considered as a fuzzy decision tree is constructed based on the available learning samples. Suppose that, by appropriate experiments, N learning samples, $\mathbf{x}_1$, $\mathbf{x}_2$, ... $\mathbf{x}_N$, were obtained at the known tool conditions. They form a set of learning data. We denote X to be the set of all the learning samples, so $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$. A learning sample $\mathbf{x}_i$ ($i = 1, 2, ..., N$, the number of learning samples) is an m-dimensional vector, where m is the number of monitoring indices. It can be written as $\mathbf{x}_i = [x_{i1}, x_{i2}, ..., x_{im}]^T$, where x_{ik} denotes the value of the kth monitoring index of the ith learning sample. If I_i is used to denote the ith monitoring index in all the learning samples, we have m sets of monitoring index data (*i.e.*, $I_k = \{x_{1k}, x_{2k}, ..., x_{Nk}\}$, and $k = 1, 2, ..., m$). Note that since a monitoring index is assumed to be continuous, I_i can be represented by the interval $I_i = [\ I_{i\text{-min}}, I_{i\text{-max}}]$ (refer to Figure 4.2).

The proposed method constructs a fuzzy neural network by partitioning the learning samples using the recursive procedure below:

(1) Set up neuron label p = 1 and the training set $X_p = X$;

(2) At neuron p,

i. using "the maximum partition" to partition the training set X_p so that $X_p = A_p + B_p$;

ii. assign $X_{p+1} = A_p$ and $X_{p+2} = B_p$;

(3) Let $p = p + 1$ and go back to (2) unless X_p contains only the samples that belong to the same class.

For the "the maximum partition," $X_p = A_p + B_p$, A_p is the set which contains the samples that belong to certain tool conditions (say, h_j or $h_i + h_j$) and B_p is the set of all the remaining learning samples. The partition separates the maximum number of samples in A_p. To do this, all the monitoring indices are examined against all the tool conditions. That is, for all I_i, $i = 1, 2, \ldots m$, we seek all intervals that contain the samples from the same tool condition. The "same tool condition" here means either one or two tool conditions. Accordingly, the one partition which separates the maximum number of samples is chosen as the maximum partition. Note that the maximum partition may not completely separate the samples of the same tool condition from the other samples since samples from different tool conditions are typically overlapped. The monitoring index associated with the partition is defined as "the pivot index" and is used as a major input of the neurons in the hidden layer.

The maximum partition at neuron *p* generates two new neurons: one with the learning set A_p and the other with the learning set $B_p = X_p - A_p$. The distribution of the learning samples is used to measure the supporting strength of the partition. The fuzzy centres of the classes considered on a neuron are designed to form a soft boundary for the partition. They are calculated as statistical means of the learning samples. All of the information are taken as the inputs to the neuron for classification purposes.

The above operation is then repeated for the new $X_{p+1} = A_p$ until all the samples in X_{p+1} belong to a unique tool condition. As a result, as each the operation is carried out, new neurons are formed and added to the hidden layer. Accordingly, interconnections between neurons are built within the networks. This procedure is illustrated in Figure 4.4.

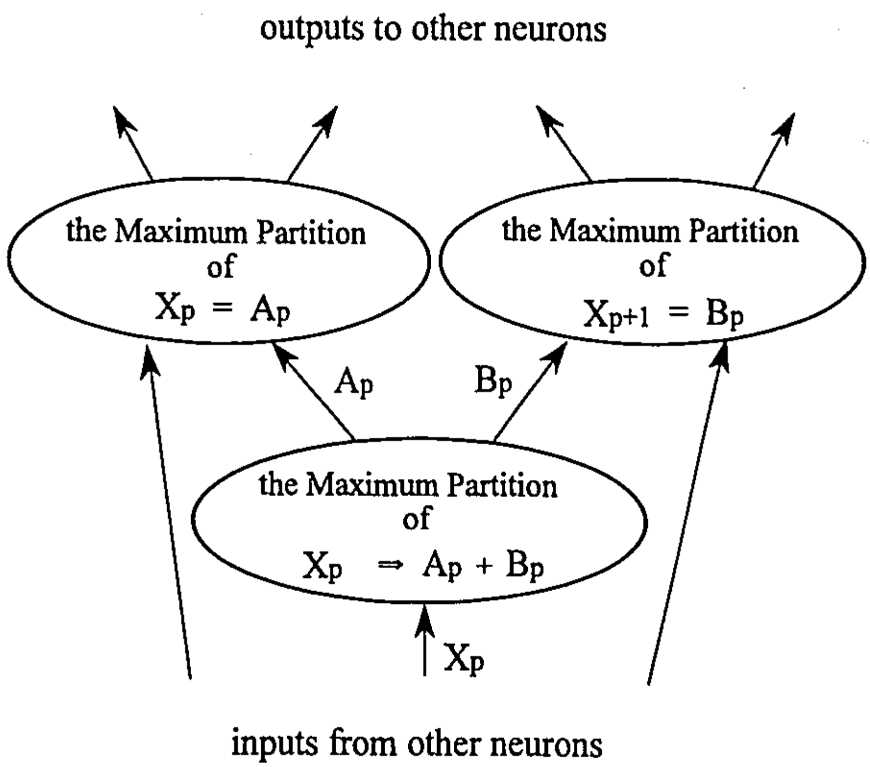


Figure 4.4 Learning of the MPC Fuzzy Neural Networks

4.3.2 Classification

The classification procedure deals with identifying the condition to which a given sample belongs. Suppose that a given samples is $\mathbf{x} = \{x_{ij}, i = 1, 2, ..., N, j = 1, 2, ..., m\}$, then the classification is performed by searching a path for the decision through the network. This path passes through a few neurons in the network and will lead to the final decision of the classification. At each neuron in the hidden layer, the search is directed by the fuzzy membership grade of the sample x at that neuron. The membership function u_{ip} that measures how much a given sample belongs to class i at neuron p is given as follows (Bezdek, 1987):

$$u_{ip} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{ip}}{d_{jp}}\right)^{\frac{2}{M-1}}}, \tag{4.12}$$

where, d_{ip} or d_{jp} is the inner product norm metric of the sample to a class centre, M is a predefined parameter, and $1 < M < \infty$. According to $\mathbf{u}_p(\mathbf{x}) = [u_{1p}, u_{2p}, .., u_{sp}]^T$ (s is the number of classes at neuron p), the direction of the next step for the search is then determined. The next neuron takes the information from the neuron under consideration as well as other neurons in the hidden and/or the input layer. The recursive procedure continues until a neuron leads to the output layer and gives the final decision. In the fuzzy neural networks, calculations can be performed simultaneously at several neurons as long as the required inputs for calculations are given. The results are given in the output layer, and the fuzzy membership grades are also provided. The proposed classification procedure for the MPC fuzzy neural networks is shown in Figure 4.5.

In general, the decision at a neuron in the hidden layer is represented by both tool conditions and their membership grades such as "d_j is μ_{ij}," where d_j, $j = 1, 2, ..., s$ (s is the number of classes at the ith neuron), is the conclusion at the ith neuron, and μ_{ij} is the membership grade for this decision. In addition, d_j may be a cluster containing either one or any combination of the decision sets in the classification space (the tool conditions). Each neuron in the hidden layer gets its inputs from the input layer as well as from some other

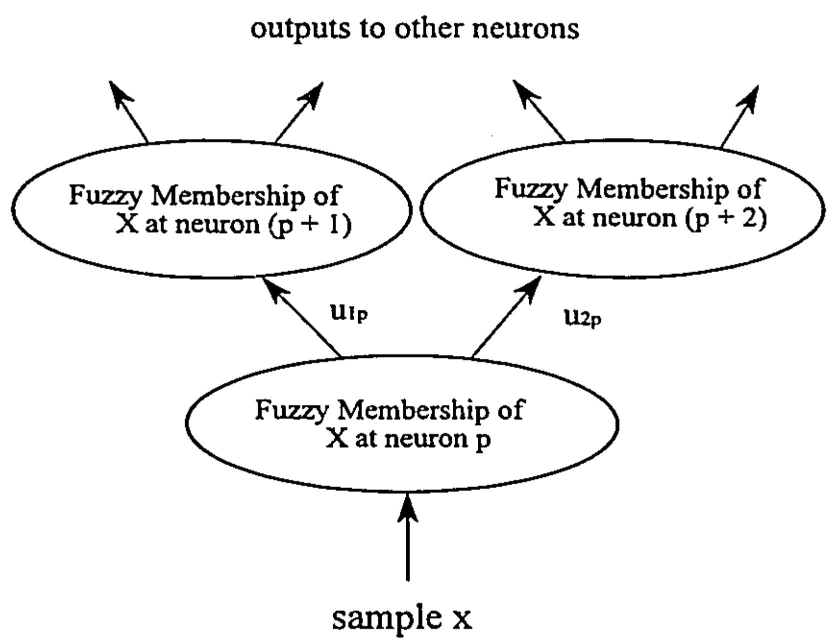


Figure 4.5 Classification of the MPC Fuzzy Neural Networks

related neurons in the hidden layer. The outputs of a neuron are sent to either the output layer or to other related neurons in the hidden layer. The classification made through this fuzzy neural network may be performed in several steps. For example, a tool may be classified to be "either sharp or worn" by some neurons in the hidden layer. A further classification specifies the final result in the output layer as "slightly worn" with a membership grade that measures the uncertainty of tool wear estimation.

4.4 UNSUPERVISED CLASSIFICATION OF THE MPC FUZZY NEURAL NETWORKS

4.4.1 Learning with the Unlabelled Samples

The knowledge (rules in the network) for tool condition monitoring is obtained by training (learning) processes. The classification based on similarity measures and discriminants are dependent upon the availability of labelled samples, or the patterns representative of each class. During the training, class labels are usually provided, along with the monitoring features, to give the necessary information for creating the rules that minimizes the misclassification. This procedure is known as pattern classification.

There are many cases where classification must be performed without *a priori* knowledge. Unlike pattern classification, which is performed with *a priori* knowledge (labelled samples), pattern clustering deals with the pattern recognition with unlabelled samples. The pattern clusters are formed according to some predefined similarities. A cluster may be defined as a set of samples which are similar to each other. Depending on the similarity criterion or measure, different clustering results are obtained. There are many possible clustering criteria that can be used (Bezdek, 1981), including distance, angle, curvature, symmetry, connectivity, and intensity.

A clustering problem is not well defined unless the results of clustering can exhibit certain properties. The choice of the properties (the definition of a cluster) is a fundamental issue in the clustering problem. For selecting a suitable measure of clusters, the mathematical properties mentioned above could be used. Moreover, the variety of "structures" is without bound. Bezdek (1981) argued that (i) no clustering criterion or measure of similarity will be universally applicable, and (ii) selection of a particular criterion is at least partially subjective, and always open to question. He categorized clustering systems into deterministic, stochastic, and fuzzy. Accordingly, three types of clustering criterion are presented: hierarchical, graph theoretical, and objective functional.

The problem of clustering for automated tool condition monitoring arises with the development of automated/intelligent monitoring systems. Metal cutting is a rather complicated manufacturing process. With the continuous introduction of new types of part-tool materials, and more aggressive cutting conditions, there is often no *a priori* knowledge about the tool failure. An automated/intelligent tool condition monitoring system should be able to update its knowledge by unsupervised learning so that it can recognize and classify new phenomena of process and tool conditions.

In this section, the MPC fuzzy neural networks are introduced for clustering in tool condition monitoring within a range of cutting conditions. Like the classification, the clustering approach uses the three basic constituents of soft computation: fuzzy logic, neural network, and probability reasoning.

4.4.2 Clustering

A. Clustering by Principal Component Analysis

Unlike pattern classification, where the class labels are provided with the training samples and the classifier seeks to find the decision boundaries between classes that minimize misclassification, pattern clustering is the process to find naturally occurring groups or clusters without labelling the training samples. In many cases, the distance is assumed to be the similarity measure. Alternatively, a cluster may be defined as a region in the feature space containing a high density of the training samples. With this definition, peaks in the sample estimates of density functions, for example, are associated with a cluster.

One of the conventional criteria for pattern clustering is the principal component analysis (James, 1985). To discuss this criterion, we will assume, for a moment, that there are two fairly compact and distinct clusters as shown in Figure 4.1. The combined covariance matrix describes the shape of the total sample distribution. Specifically, the eigenvector corresponding to the maximum eigenvalue is the direction of the maximum variance. If we take $\mathbf{t}_1$ as the maximum separation direction, we can project the samples on $\mathbf{t}_1$, construct a frequency *vs* $\mathbf{t}_1^T\mathbf{x}$ histogram and expect to obtain a distribution similar to those shown in Figure 4.1. Choosing a threshold θ as the minimum, we can specify the clusters with

$$\mathbf{x} \in \text{class A}, \qquad \text{if } \mathbf{t}_1^T\mathbf{x} < \theta. \tag{4.13}$$

As mentioned previously, the principal component analysis takes into account the directions in the measurement vector $\mathbf{x}$ which have the largest covariance with the class (machining process and tool conditions) vector, and ensures that these directions (eigenvectors) are used for partitions.

After the principal component analysis, a local minimum can be obtained as a "reference of boundary." There are three possible results of such separation as shown in Figure 4.6. The first one is the ideal case in which the two classes are perfectly separated by

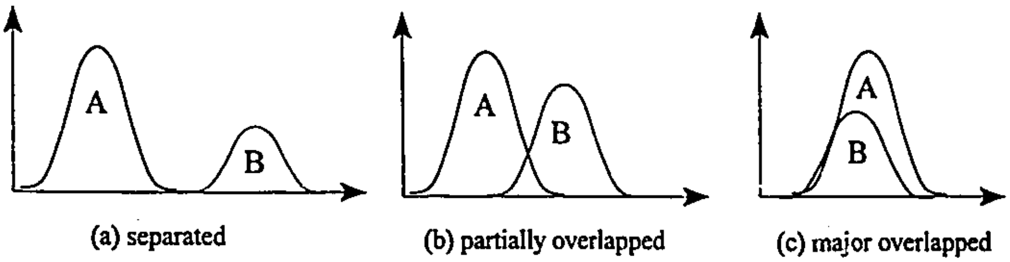


Figure 4.6 Possible Distributions of the Learning Data

a boundary line. In the second case, the two classes are overlapped, and the boundary (the minimum) is actually a mixture of the samples from both classes. Misclassification is hereditary. The criterion for pattern classification is to minimize such misclassification. For the third case, one class is included in the other and the principal component analysis fails.

As we noted earlier, the principal component analysis gives a possible way for grouping unlabelled samples. If we choose a peak on the frequency histogram as a class, the boundary is usually formed at a valley, the local minimum, of the curve. This is the conventional principal component analysis in pattern clustering. We try to improve the separation by introducing a fuzzy classification method. Then, the boundary between two classes is "soft" and similar to that shown in Figure 4.2. As a membership function is defined for clustering, the clustering results can be measured by the fuzzy membership grades for each partition. The clustering decision at this stage is made with

$$\mathbf{x} \in \text{class } i \text{ in the grade of } u_i \tag{4.14}$$

B. Clustering by the MPC Fuzzy Neural Networks

The principal component analysis is one of the most simple and effective clustering schemes. Unfortunately, it is unable to deal with complicated problems, such as those involving $C > 2$ clusters or cases with non-spherical distributions in the principal direction (the largest eigenvalue eigenvector's direction). A strategy for clustering by the MPC fuzzy neural networks is proposed, which constructs a combination of certain simple procedures. The principal component analysis is done in multiple directions at neurons of the network while the fuzzy classification is applied to each neuron. The fuzzy neural networks are used for exploiting the parallel nature of the classification and, for providing the fuzzy classifications for certain complex problems. The partial connections are used for reducing unnecessary calculations in both the learning and the classification.

To cluster a set of original measured samples, we will first partition them into several sub-clusters. Then the same scheme is applied to each sub-cluster to get further detailed partitions. Different from other conventional hierarchical clustering algorithms, which combine each individual samples to form a new cluster and then try to combine similar clusters for generating a new one, this proposed scheme uses an "up-side down" tree to seek a possible separation of the original data set for generating new clusters. The processing work is performed at every neuron of the network.

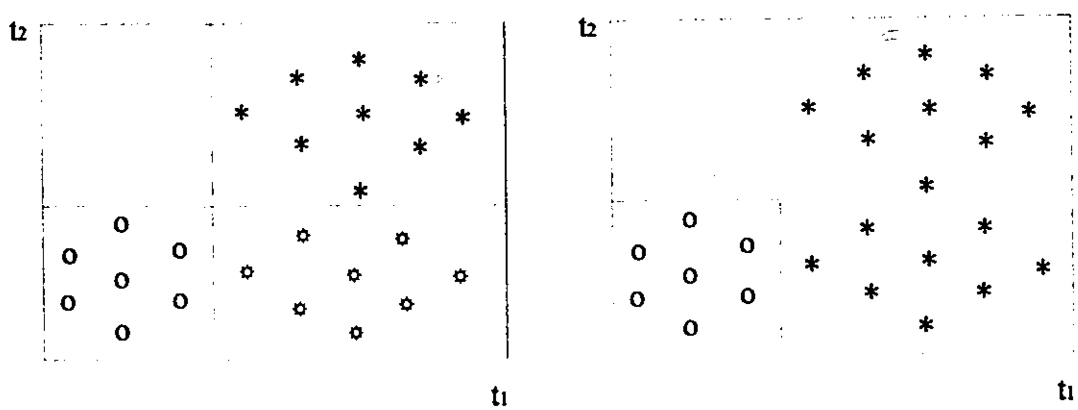
The proposed clustering method combines the principal analysis, the fuzzy classification with a decision tree for defining the structure of the hidden layer. The principal component analysis on the input data is performed at each neuron. To solve the problem of non-spherical distributions, several directions of the eigenvectors are considered. For the two-dimensional case shown as an example in Figure 4.7, the significant separations in both directions are required and four regions of separation can be obtained. The membership grades of the input samples for each region are calculated by using the above-mentioned fuzzy membership function (Equation 4.12). Accordingly, the samples are labelled with the membership grades for each of those four regions. Then, an attempt is made to merge those four regions into two clusters for the simplicity by the following procedure:

As we know, a total of seven possible combinations exist for this example. They are $\{C_1,\ C_2 \cup C_3 \cup C_4\}$, $\{C_2,\ C_1 \cup C_3 \cup C_4\}$, $\{C_3,\ C_1 \cup C_2 \cup C_4\}$, $\{C_4,\ C_1 \cup C_2 \cup C_3\}$, $\{C_1 \cup C_2,\ C_3 \cup C_4\}$, $\{C_1 \cup C_3,\ C_2 \cup C_4\}$, $\{C_1 \cup C_4,\ C_2 \cup C_3\}$. To choose the most convenient one, a fuzzy discriminant function is defined:

$$u\,(C_{new\text{-}1},\ \ C_{new\text{-}2}) = \min \{\ dist\,(C_{new\text{-}1}, C_i),\ dist\,(C_{new\text{-}2}, C_i),\ \ \forall\ i\ \} \qquad (4.15)$$

Where, $C_{new\text{-}1}$ and $C_{new\text{-}2}$ are the new-formed clusters, $dist(C_i\,, C_j)$ is the distance between cluster C_i and C_j which is measured from one centre to the other. By minimizing the above discriminant function u ($C_{new\text{-}1}$, $C_{new\text{-}2}$), a new pair of clusters is obtained as $C_{old} = C_{new\text{-}1} \cup C_{new\text{-}2}$. The same procedure is carried out for the new cluster(s) until the final clustering result is satisfactory.

The clustering process of the MPC fuzzy neural networks can be implemented as follows:



(a) clustering after principal component analysis

(b) result of cluster combination

Figure 4.7 Clustering by the Principal Component Analysis and the Cluster Combination

(1) C_{old} = all samples;

(2) the principal component analysis on C_{old} is performed in two directions to create four regions for the new clusters;

(3) the calculation of the membership grades of all the samples in C_{old} for all C_{new}s;

(4) the selection of the most convenient pair of combinations by minimizing the fuzzy discriminant function to generate two new sub-clusters so that $C_{old} = C_{new-1} \cup C_{new-2}$;

(5) Assign $C_{old} = C_{new}$ and go back to (2) until the final clustering is done.

The same procedure can be used for a clustering problem with more than two dimensions, where the number of the possible sub-clusters may vary. Moreover, the C_{new}

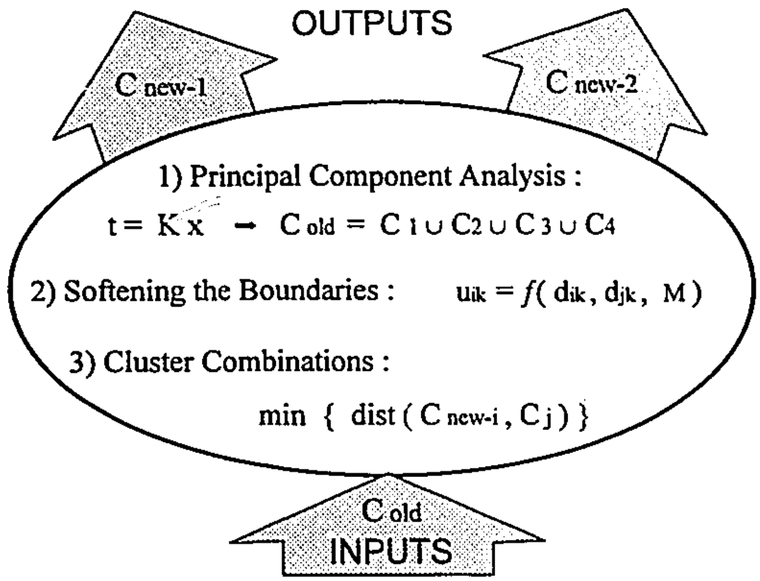


Figure 4.8 Clustering at a Neuron of the MPC Fuzzy Neural Networks

may be a set containing either one class or any combination of the decision sets in the classification space. Each neuron in the hidden layer gets its inputs from the input layer as well as from some other related neurons in the hidden layer. The outputs of a neuron are sent to either the output layer or to other neurons in the hidden layer. Like the classification with the MPC fuzzy neural networks, the clustering tool conditions using thess fuzzy neural networks may be performed in several steps. For example, a tool may be classified to be "either sharp or worn" by some neurons in the hidden layer. Then, further clustering specifies the final result in the output layer as "slightly worn" with a certain degree that measures the imprecision of tool wear estimation. The clustering process at a neuron in the MPC fuzzy neural networks is illustrated in Figure 4.8.

Table 4.1 Success Rates of Clustering Fisher's Iris Data with Different Methods

Clustering Methods		Class 1	Class 2	Class 3	Total
the MPC Fuzzy Neural Networks		**100 %**	**100 %**	**90 %**	**96.7 %**
the Fuzzy Min-Max Neural Networks (Simpson, 1993)	Test A	100 %	100 %	60 %	86.7 %
	Test B	100 %	94 %	70 %	88.0 %
	Test C	98 %	96 %	78 %	90.7 %
	Test D	100 %	88 %	90 %	92.7 %
the Nearest Neighbour Method (James, 1985)					96.0 %

4.4.3 Comparison to Other Clustering Methods

The proposed clustering algorithm of the MPC fuzzy neural networks was tested using data from the available literature. Fisher's Iris Data (refer to James, 1985) is the first testing data set selected because of its familiarity to the pattern recognition research community which allows a measure of relative performance. The Iris Data consist of 150 four-dimensional feature vectors (patterns) in three separate classes, 50 for each class. Fisher's Iris Data was successfully clustered by the proposed fuzzy neural networks. The samples from class 1 are first separated from the others at a neuron. The remaining samples are further divided into two new clusters. Table 4.1 gives the detailed results of clustering Fisher's Iris Data by the proposed MPC fuzzy neural networks as well as other well known methods. The results of clustering by the MPC fuzzy neural networks are shown in Figure 4.9, where (a) is the result of the first clustering step (the first neuron in the hidden layer) and (b) is the result of the second clustering step.

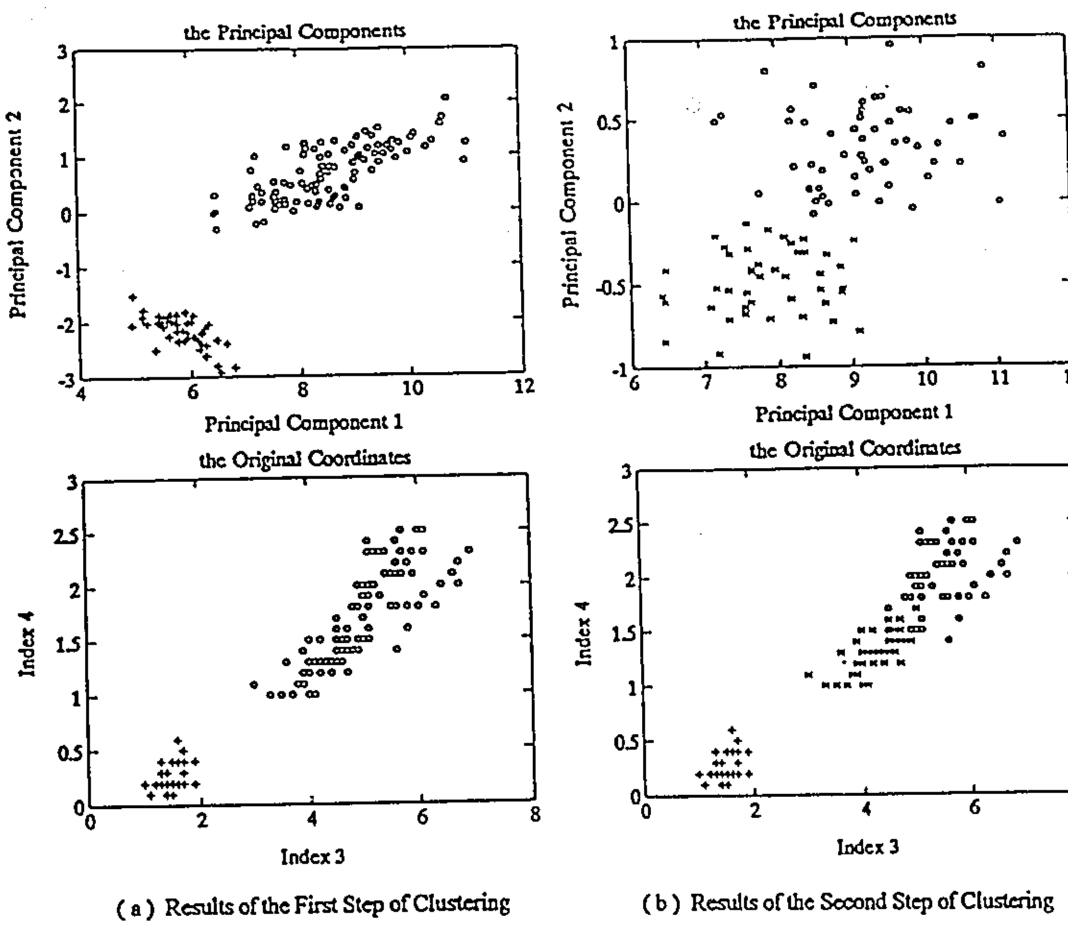
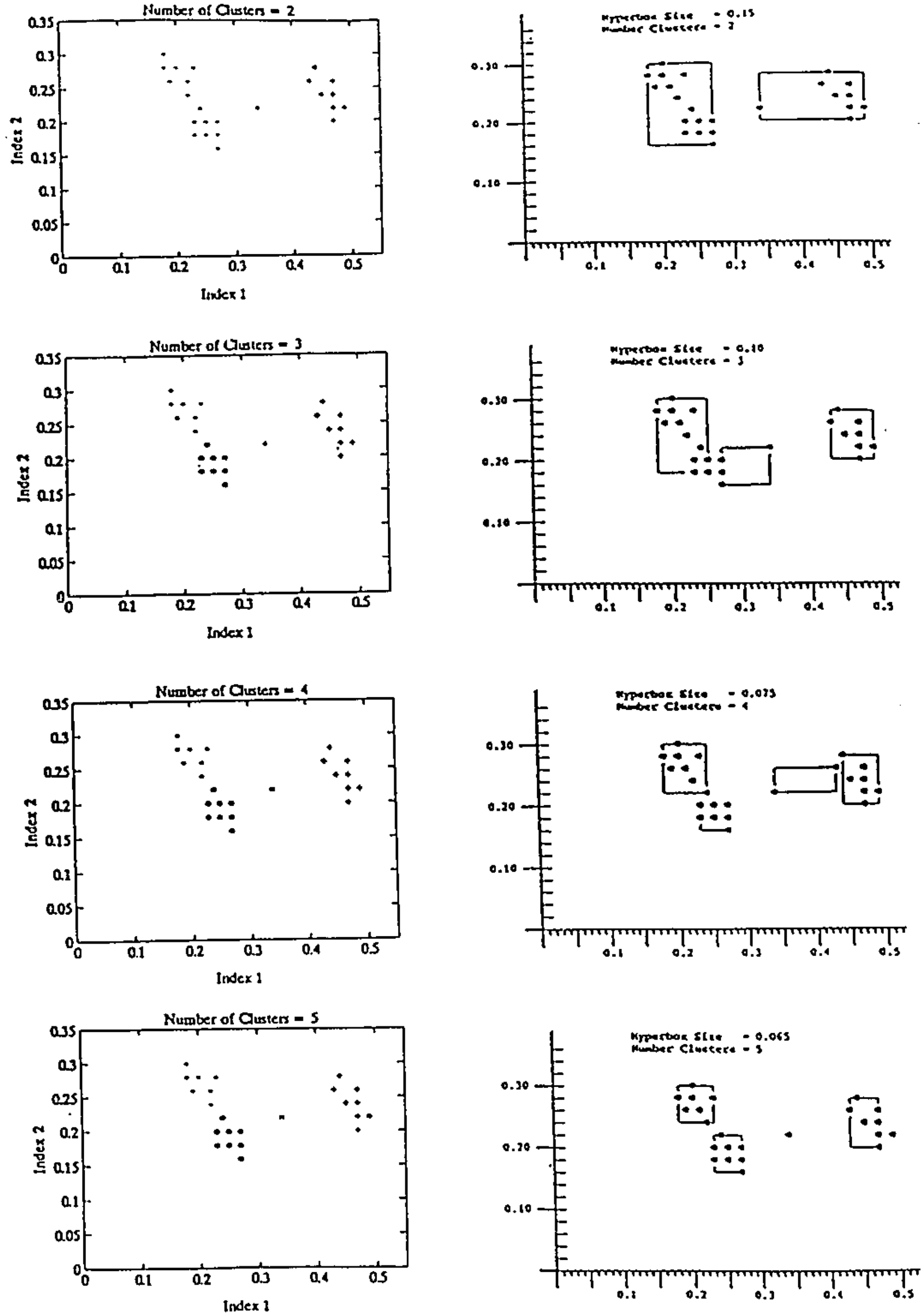


Figure 4.9 Clustering Results for Fisher's Iris Data

The other set of testing data is Simpson's (1993). The data consist of 24 two-dimensional points. Figure 4.10 shows the clustering performance of the proposed fuzzy neural networks, where Simpson's results from the Min-Max fuzzy neural networks are also given for comparison. The results of clustering such data vary depending on the predefined number of clusters and the parameters in the networks. Even though there are slight differences in clustering the same data into the same number of clusters, the results by both neural networks showed great similarity.



(a) the MPC Fuzzy Neural Networks

(b) the Min-Max Fuzzy Neural Networks

Figure 4.10 Clustering Results for Simpson's Data

4.5 KNOWLEDGE UPDATING OF THE MPC FUZZY NEURAL NETWORK

4.5.1 Retraining with the New Information

As mentioned previously, a tool condition monitoring system should be able to update its knowledge using the new information related to the cutting process. On-line adaptation is a key problem in the design of fuzzy neural networks for tool condition monitoring in machining. Many popular neural networks and traditional pattern classification techniques utilize off-line adaptation. Each time the new information is added to the classification system, it requires a complete retraining of the system with both the old and the new learning data. As such, off-line adaptation requires more computer memory and leads to longer training times. On the other hand, a good knowledge updating algorithm has to be a simple process that utilizes the maximum information from the new learning data to modify the existing structure and the parameters, and without destroying the stored class information.

The knowledge updating algorithm for the MPC fuzzy neural networks is developed with the assumption that the system has been trained and keeps only the necessary information (from the old learning data) for classification purposes. The original learning data which trained the system is not saved. The idea is that the trained system modifies itself by obtaining updating information only from the new learning data. The main structure of the old system is not destroyed. By the updating process, some class information stored at neurons in the system are renewed. The interconnections might be slightly changed. New neurons may be inserted into the network. Behaviour of the retrained system is evaluated by its performance

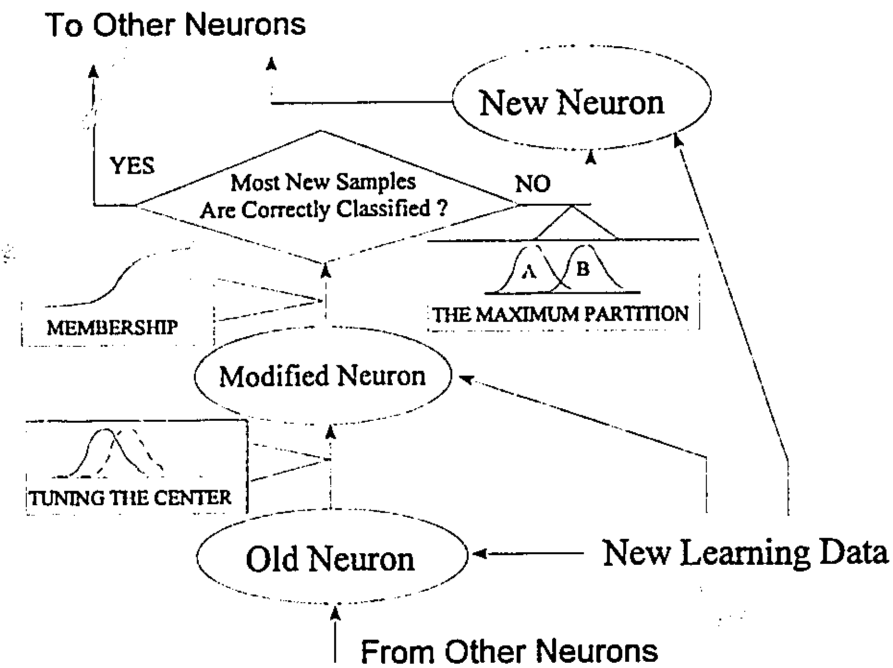


Figure 4.11 Knowledge Updating of the MPC Fuzzy Neural Networks

in classifications after the knowledge updating process.

Updating class information without using old data has advantages. Once the system is trained, the learning data are discarded. Thus, the memory for information storage is greatly reduced. Modification of the system using only the new learning data takes much less time than training a new system with both old and new data sets. The difference in training times becomes more significant when the knowledge updating is performed several times.

The proposed knowledge updating algorithm deals with two issues: *Tuning Old Neurons* and *Adding New Neurons*. Such a retraining scheme is depicted in Figure 4.11.

4.5.2 Tuning Old Neurons

During the learning procedure of the MPC fuzzy neural networks, the maximum partition is applied in multiple principal component directions. The learned class information at the neurons is mainly obtained based on the distribution of the learning samples. One neuron in the network typically contains the following class information:

(1) *The pivot index*, the monitoring index used at this neuron for the maximum partition;

(2) *Class clusters*, which are partitioned by the maximum partition at this neuron (in this study, two class clusters are generated at each neuron and each cluster may contain up to five tool conditions);

(3) *Class centres*, the class central positions represented by the mean values of the learning samples;

(4) *Number of the learning samples*, which are used in the learning at this neuron;

(5) *Class boundary*, the discriminant function that distinguishes two classes (this function is given in Equation 4.12 and the parameter M defines the gradient of the sigmoidal curves); and

(6) *Connections*, which indicate which neurons in the neighbouring layers are directly related to this neuron.

In the knowledge updating process, the new learning samples are first introduced to the system for checking the distribution in the principal component directions. If the majority of the samples in the retraining set fall around the class centres at an old neuron, then the class information at that neuron will still be reliable. Thus, the pivot index and the class clusters are kept the same. The class centres and the class boundary may be tuned to enhance the accuracy of classification. Assume that n samples were used for the original learning and there are m new learning samples available for the knowledge updating. The estimate of old class centre is $\bar{O}$. The new estimate of the class centre, $\bar{N}$, is then given by

$$\overline{N} = \frac{n}{n+m}\,\overline{O} + \frac{1}{n+m}\sum_{i=1}^{m} x_i \tag{4.16}$$

where the summation is taken for all the learning samples, x_i, in the retraining set. The new learning samples are used for tuning the class centres. The fuzzy membership function defined in Equation 4.12 could be modified by tuning M to fit the requirement of the new boundaries. The tuning process at an existing neuron does not change its connections to the other related neurons.

The tuning of the class information at old neurons is similar to the adjustment of the expected value (the sample mean) and the feature variance from the new learning data for a Bayes classifier. The class centre and the sample mean define the location of the samples in the class. The boundary gradient and the feature variance describe the density of samples around the class centre. All these parameters are affected by both old and new learning data. The performance of a Bayes classifier depends greatly on the probability density function. The new learning samples will adjust the hard boundary position (threshold) from the new estimates of the data. In the MPC fuzzy neural networks, the boundary between classes are fitted by tuning M in the membership function according to the distribution of the learning data. Distribution estimates of the learning data are not required.

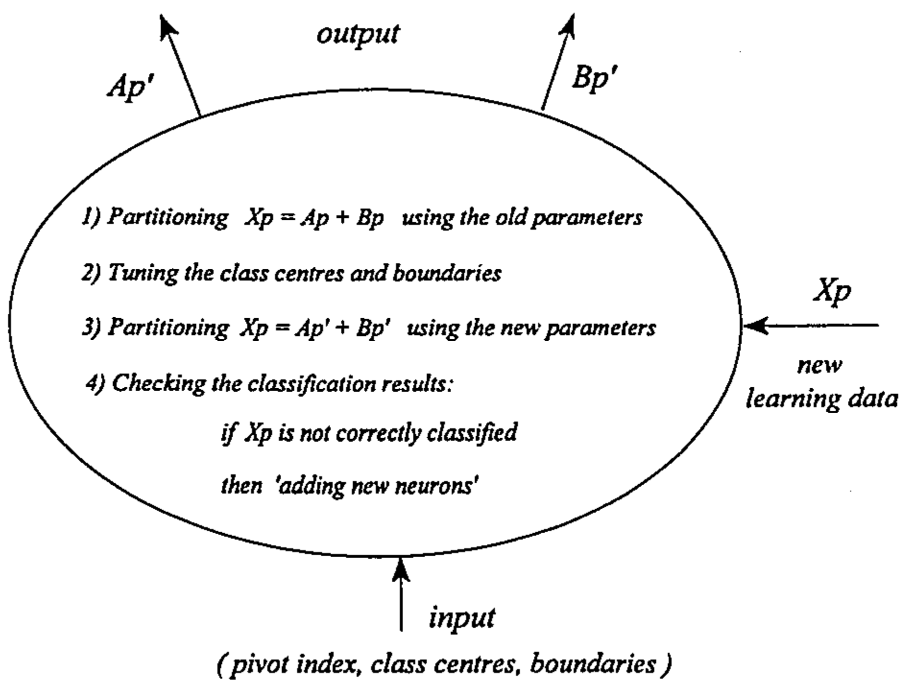


Figure 4.12 Tuning Old Neurons in the MPC Fuzzy Neural Networks

The procedure of tuning old neurons for the MPC fuzzy neural networks is summarized as following:

(1) Set neuron label $p = 1$, and the retraining set $X_p = X$, where X is the whole set of the learning samples for the knowledge updating;

(2) Partition X_p into two sets so that $X_p = A_p + B_p$, where A_p and B_p, respectively, are the sets which contain the samples belonging to the two class clusters defined at neuron p;

(3) Tuning the class centres at neuron p using Equation 4.16;

(4) Fixing the class boundary by adjusting parameter M in Equation 4.12 and examining the classification results of the learning samples in X_p;

(5) Set the retraining sets $X_{p+1} = A_p$ and $X_{p+2} = B_p$ for next neurons;

(6) Let $p = p + 1$, and go to step (2) until

(i) the majority of the samples in X_p are correctly classified with the new class information and neuron p defines only one-class cluster, or

(ii) carrying out the "adding new neurons" process at neuron p.

This procedure is depicted in Figure 4.12. The parameters at the neuron are modified, but the connections are not changed.

4.5.3 Adding New Neurons

Tuning of old neurons with new information is carried out based on the assumption that the learning samples for knowledge updating have the same distribution as the old learning samples in the principal component directions. If this does not apply to the new learning data, we can add new neurons into the network.

As discussed previously, the class centre and the class boundary at a neuron can be modified by a tuning process. First, we can use the new centre and the new boundary to check the classification of the new learning samples at this neuron. If the majority of the new learning samples at this neuron cannot be classified correctly, we have to seek new class information to solve the problem. Among those of the old class information, the connections can be changed for inserting new neurons. For example, an existing neuron, S, classifies the samples into two classes, A and B. It generates two separate neurons, Q and R, indicating class A and class B respectively. The output of neuron S is directly connected to the single-class-defined neuron (separating a single class; here, it is B), R in the output layer. Suppose that most of the learning samples belonging to class A in the retraining set are incorrectly classified into class B at neuron S. The updated class information by the tuning is not working for the classification of these data. To solve this problem, we cut the connection of neuron R to the output layer and implement the maximum partition as defined in the learning procedure on the retraining samples. The maximum partition on the retraining set (containing both class A and class B) at neuron R generates two new neurons in the network, say R1 and R2. After this operation, neuron R is no longer a single-class-defined neuron. New class information including the pivot index, the class clusters, the class centres, and the class boundary is obtained. The old output of R connection to the output layer is replaced by the new connections to neurons R1 and R2.

R1 and R2 are the new neurons inserted in the network. They could be single-class-defined neurons and connected to the output layer. If they are not single-class-defined, the same learning procedure is applied until one single class is separated at a neuron. New neurons are usually inserted between neurons which are close to the output layer.

The procedure of adding new neurons for the MPC fuzzy neural networks is summarized as following:

(1) Classify the learning samples with the tuned class information and partition the retraining set A into two sets: A = A1 + A2. A1 contains the correctly classified samples. A2 contains the incorrectly classified samples.

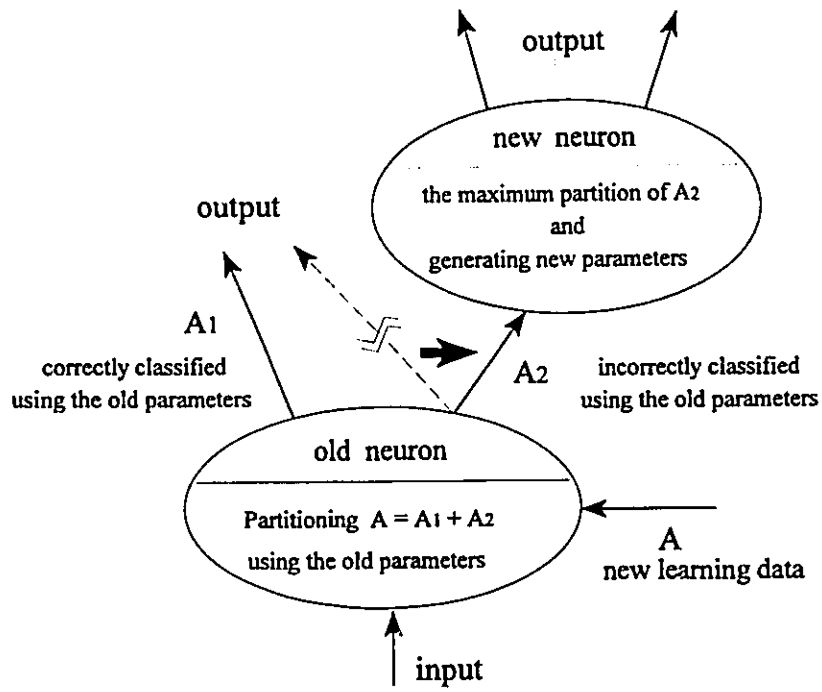


Figure 4.13 Adding New Neurons in the MPC Fuzzy Neural Network

(2) Break up the old connection to the output layer of the neuron. Generate two neurons and put connections to these new neurons.

(3) Learn new class information by using the maximum partition on set A2 and the old class information. Replace the old information with the new class information at this neuron and store the necessary class information in the two new neurons.

(4) Go back to step (1) at the new neurons until single-class-defined neurons are generated. For single-class-defined neurons, put the connection to the output layer.

This procedure is depicted in Figure 4.13. The new neurons are added to the network and the connections are changed.

4. 6 EVALUATION OF THE MPC FUZZY NEURAL NETWORKS

There are several properties that a pattern classifier should possess (Simpson, 1992). They are on-line adaptation, nonlinear separability, dealing with overlapping classes, short training time, soft and hard decisions, verification and validation, tuning parameters, and nonparametric classification. The proposed MPC fuzzy neural networks for automated tool condition monitoring have been developed in consideration of these requirements. The training and classification algorithms are based on the theories of neural networks, fuzzy logic, and probability reasoning. In the feature extraction, metal cutting mechanics are primarily studied for a distinctive feature space, and the principal component analyses are applied in multiple directions.

With the application of fuzzy classification, the neural networks are effective in dealing with nonlinear separable and/or class-overlapping classification problems, which are common in the case of tool condition monitoring in machining, especially for the monitoring with varying cutting conditions. The partial interconnections within the fuzzy neural networks make the training time very short compared to that of fully connected networks such as the back-propagation neural networks. The calculations necessary for the classification are also significantly reduced since not all the neurons in the hidden layer are used while a sample is being processed. Soft and hard decisions are optional for different applications. The maximum partition algorithm is based on the distributions of the learning samples and no parameter estimations are needed.

The proposed method functions similarly, in the partition of training samples, to the linear fuzzy equation algorithm proposed by Du, Elbestawi and Li (1992). The linear fuzzy equation method uses a matrix to describe the relationship between the monitoring indices and the tool conditions. The proposed MPC fuzzy neural networks use a tree structure similar to that in the fuzzy decision tree described in the work of Li, Elbestawi and Du (1992). Because the decision tree is more flexible than a matrix approach, it has better performance in the case of tool condition monitoring in machining. In constructing the fuzzy decision tree, the maximum partition generates nodes holding the samples from only one tool condition. The other samples are put into other nodes. This means each partition leads to a final decision at a leaf node of the tree. The maximum partition in the MPC fuzzy neural networks chooses a better partition so that a new-born neuron can hold samples from different tool conditions. A neuron can lead to other neurons in the hidden layer as well as neurons in the output layer. The consequence of this structure is simplicity in the interconnection and the short routines in the classification. Experimental tests by using the same set of data showed that the MPC fuzzy neural networks gave a better success rate than the fuzzy decision tree algorithm (Li *et al*, 1994).

In the consideration of on-line adaptation (on-line learning) and the tuning parameters, a classifier should have as few parameters to tune in the system as possible. Both the back-propagation neural networks and the proposed MPC fuzzy neural networks have very few tuning parameters. The structure of the MPC fuzzy neural network is, however, easily modified with new learning samples. Unlike the back-propagation neural networks that require a complete retraining of the system with both the old and the new learning data, the MPC fuzzy neural networks need only to change partially their neurons and the connections when the new learning information is added. Both supervised and unsupervised classification algorithms are easily implemented with the available learning samples.

The proposed classification algorithm of the MPC fuzzy neural networks is based on the three components of "soft computation." In the learning procedure, partitions of the learning samples are carried out by examining the distribution of the learning samples in multiple principal component directions. Like other statistical reasoning methods, the "maximum partition" relies greatly on the distribution of the data. If the data are scattered in the feature space, a cluster centre is impossible to be recognized. So the basis of the classification is that the learning patterns are distributed around their centre(s). This is also believed to be true to all kinds of pattern classifiers. The major difference in the proposed algorithm with other statistical reasoning is that parameter estimation is not required.

To insure a good distribution of the learning data, the training samples have to be representative of the whole span of the feature space. In tool condition monitoring, all applicable tool and cutting conditions have to be considered during the training phase. On the other hand, if a poor distribution is encountered, then a modified feature extraction procedure has to be implemented. Information about new phenomena can be added to the monitoring system by knowledge updating.

4.7 SUMMARY

A new approach to the development of automated tool condition monitoring in machining is proposed. The system consists of sensor fusion, neural networks, fuzzy logic and statistical reasoning. These simple procedures attack different problems within a complex classification problem. The combination of these techniques creates the Multiple Principal Component (MPC) fuzzy neural networks. The proposed decision making algorithms utilize the flexible structure of partially connected neural networks and the uncertainty measurement of fuzzy logic. The maximum partition is implemented in multiple directions based on the distributions of the learning samples.

The major subjects in the development of the MPC fuzzy neural networks are supervised classification, unsupervised classification and knowledge updating. Supervised classification deals with training the system with the labelled learning samples. The principal component analyses are implemented in multiple directions for feature extraction as well as for pattern partitions. The fuzzy neural networks are built through the learning procedures. Classification is done by examining the fuzzy membership grades of the samples at neurons in the network. The search directions are guided by the membership grades, and the final conclusion is given at the neurons in the output layer. Unsupervised classification deals with training the system without labelled learning samples. The pattern partitions are based on the principal component analyses in multiple directions and the fuzzy membership function which sorts the optimum pattern combinations. Knowledge updating algorithm involves tuning the old neuron parameters and adding new neurons to the existing network.

CHAPTER V

EXPERIMENTAL TESTS ON THE MPC FUZZY NEURAL NETWORKS FOR TOOL CONDITION MONITORING

5. 1 INTRODUCTION

The proposed MPC fuzzy neural networks for automated tool condition monitoring in machining were tested by experiments in turning and drilling.

Several sensors were used for measuring the process signals. Force, torque, vibration and spindle motor power signals were fused using the principal component analysis to produce highly sensitive features. The features generated by sensor fusion were taken as the inputs to the fuzzy neural networks for automated tool condition monitoring. The tool conditions considered in the monitoring tests included sharp tool, tool breakage, chipping and different states of tool wear. The classified tool conditions were obviously the outputs of the fuzzy neural networks.

The experiments were conducted for testing the performance of the MPC fuzzy neural networks in three subjects: supervised classification, unsupervised classification and knowledge updating. The learning and the testing samples were obtained from the cutting tests conducted within a reasonable range of cutting conditions.

The experimental setups and results are discussed in this chapter. Tests in turning and drilling will be addressed separately.

5. 2 TURNING EXPERIMENTS

5.2.1 Definition of the Tool Conditions

The experiments were conducted with five different tool conditions, namely *Sharp* tool (SHP), *Breakage* (BRK) and three states of tool wear which are defined as *Slight* (SLW), *Medium* (MDW), and *Severe* (SVW) *Wear*. The tool wear criterion considered was the flank wear (refer to Appendix). For the first tool condition, sharp tools or tools with flank or crater wear smaller than 0.1 mm and with no chipping were used. Tool breakage was identified by a chipping area on the tool larger than 0.04 mm^2. Tool wear states were defined

Table 5.1 Definition of the Tool Conditions in Turning Tests

Tool Conditions	Sharp Tool	Breakage	Slight Wear	Medium Wear	Severe Wear
Tool Features	wear < 0.1 mm	chipping > 0.04 mm^2	0.1 mm < wear < 0.16 mm	0.16 mm < wear < 0.3 mm	0.3 mm < wear < 0.6 mm

as follows: slight wear — with average flank wear between 0.1 mm and 0.16 mm, medium wear — with average flank wear between 0.16 mm and 0.3 mm, and severe wear — with average flank wear greater than 0.3 mm. These conditions are summarized in Table 5.1.

5.2.2 Experimental Setup

Cutting tests were performed using an NC lathe (Standard Modern model N/C 17). A schematic diagram of the experimental setup is shown in Figure 5.1. During the experiments, six process signals were measured which include three components of cutting forces, F_x, F_y, and F_z, two vibration signals, A_x and A_y, and the cutting power, *Pw*. The cutting forces were measured by using a three-dimensional dynamometer mounted under the tool holder. The vibrations were monitored by using two accelerometers located on the tool holder in both the axial and the radial directions. The cutting power, *Pw,* was measured using the armature current of the spindle motor along with the cutting speed.

The force and the acceleration signals were first passed through charge amplifiers (Kistler 5804) and low pass filters (cut-off frequency of 1 kHz), and then were sampled at 2 kHz. The power signal was also sampled at the same frequency. The workpieces were

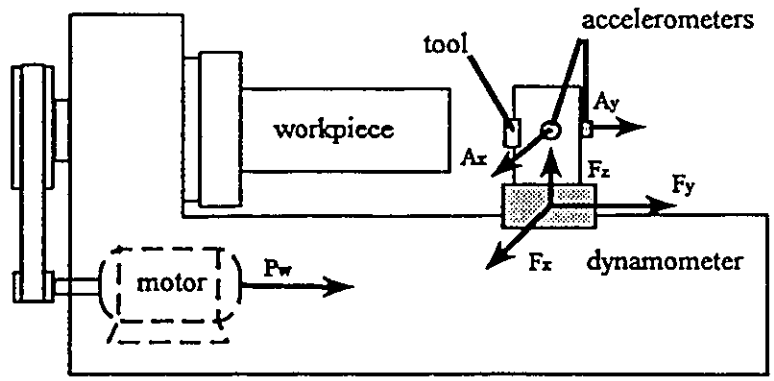


Figure 5.1 Sensor Setup for Turning Tests

AISI 1014 steel bars and the cutting tools were carbide inserts (grades K21and K6).

Various cutting tests were conducted at 52 different cutting conditions. The variations of the cutting conditions include cutting speeds ranging from 96 to 322 m/min, feeds ranging from 0.024 to 0.246 mm/rev, and depths of cut ranging from 1.2 to 3.5 mm.

5.2.3 Signal Conditioning and Feature Selection

A schematic representation of the signal processing system used in this study is shown in Figure 5.2. A total of 97 features were used to generate a measurement vector **x**. Of these features, nine were obtained from time domain records (length 2048) of the force, vibration and power signals. From each time domain record of the forces, both the mean and the variance were calculated. The variances of the acceleration signals were also calculated. From the power records, the root-mean-square values were determined. In addition, 88

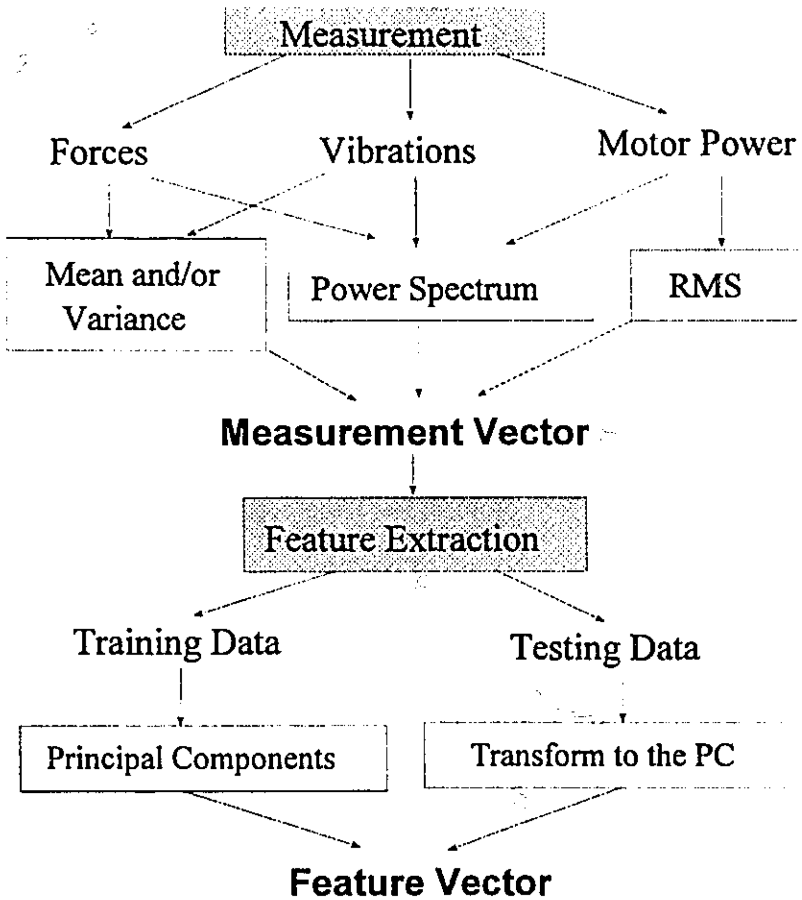


Figure 5.2 Signal Processing

features were obtained from the power spectra of the force, vibration and power signals. These frequency domain features represented the signal powers in a few specified frequency bands, calculated by the summation of the signal power at each frequency in the bands. For both the force and the vibration signals, the width of frequency bands was chosen to be 48 Hz, while for the cutting power signals the band width was 97 Hz. The frequency range of analysis for all the signals was 0 to 768 Hz.

As mentioned above, 97 features were selected to define the measurement vector. The dimension of this vector is too large for the classification. These features also may not give the best directions of separation in the classification space. Feature extraction is implemented to reduce the pattern space and simplify the decision surfaces. We define x as an n-dimensional vector in the pattern space, also known as a measurement vector, which is obtained by the signal processing. We seek to extract m features (the monitoring indices, and $m << n$) from vector x to form the feature space. Cover (1965) provided the guidelines for determining the optimum number of features, m, and the number of samples, N, to be used for the training and the classification test:

$$N \geq 2\ (m + 1)\ . \qquad (5.1)$$

In our experiments, by the principal component analysis, the number of the major principal components (the most significant eigenvalues) was determined to be six, The dimension m of the feature vector was, therefore, set to be six. Based on Cover's equation, it was decided that 15 samples per class were necessary for the learning. The experimental data were randomly selected within the considered rage of cutting condition. Each set contains 75 samples from five tool conditions. For each class, cutting conditions used in generating the training samples were chosen at random to cover the entire range considered in this study.

5.2.4 Experimental Results

A. Tests for the Supervised Classification

The MPC fuzzy neural networks (are simply denoted as FNN in this section) for automated tool condition monitoring in machining were trained using the procedure mentioned earlier (refer to Section 4.3). Four sets of experimental data (namely A, B, C, and D) were generated from the cutting tests. Both sets A and B were generated under the same cutting conditions but from different records. Sets C and D were generated under different cutting conditions. However, all the cutting tests for sets A, B, C, and D were performed under conditions which fall within the range given in Section 5.2.2. Each set of data contained 75 samples belonging to the five tool conditions considered in this study (*i.e.*, 15 samples for each tool condition). The principal component analyses were implemented for the feature extraction. The learning samples were transformed into the six-dimensional feature space (in the maximum component directions) before they were used for training the systems. The classification data were also transformed into the same features space. Table 5.2 summarizes the results obtained from those tests. For comparison, the results obtained using well-known feed-forward neural networks trained by the back-propagation (BPNN) with the same sets of data are also given.

As shown in the table, any decision making method performs better when using the same samples for both the learning and the classification. In general, the results show that using the proposed MPC fuzzy neural network for tool condition monitoring in machining, along with the integration of multi-sensor information, has resulted in the higher success rates than those obtained by using the BPNN. An important advantage of the proposed method

Table 5.2 Results of Supervised Classification with Different Tests

Test	Training Data	Classification Data	Classifications	Success Rate of FNN	Success Rate of BPNN
# 1	Group A	Group A	Self-Classification	94.7 %	89.3 %
# 2		Group B	Different Record Same Cutting Condition	89.3 %	80.0 %
# 3	Group B	Group B	Self-Classification	94.7 %	84.0 %
# 4		Group A	Different Record Same Cutting Condition	84.0 %	70.7 %
# 5	Group C	Group C	Self-Classification	96.0 %	86.7 %
# 6		Group D	Different Cutting Condition	80.0 %	69.3 %

is its good classification performance in the tool condition monitoring within a reasonable range of cutting conditions, even though the classification samples used are obtained under different cutting conditions than those of the training samples.

Figure 5.3 gives an example of the fuzzy neural networks generated in tests #1 and #2. The classification of a sample at a neuron is carried out based on the information such as the pivot index, the fuzzy centres, the membership function, and the associated classes. Then, the sample is either sent to other neurons for further classification, or put to the output layer where the final decision is given. The interconnections within the neurons are clearly seen in the network. At each neuron, only parts of the classes, the monitoring features are considered. The neurons in the network are partially connected.

Figure 5.4 shows the detailed results of classifying the three stages of tool wear in test # 2 by two different neural networks. In detecting tool wear, the fuzzy neural networks give

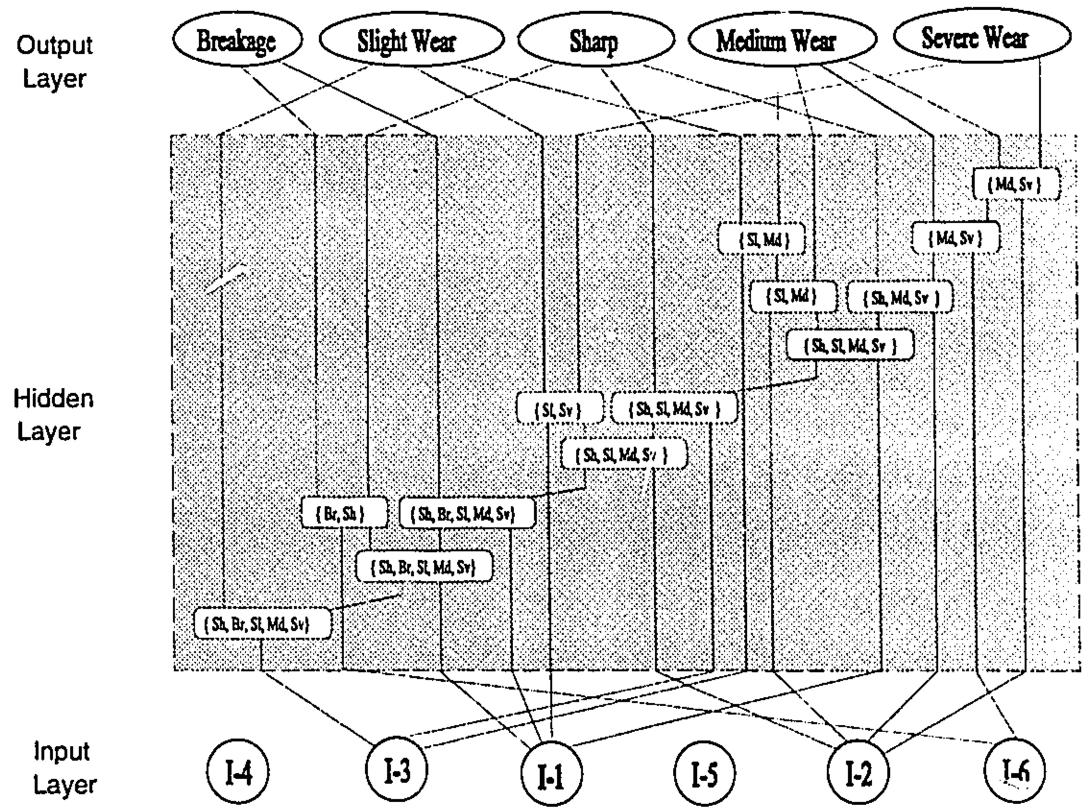


Figure 5.3 Example of the MPC Fuzzy Neural Networks (Test #1 & #2)

better separations between the three wear stages than the back-propagation trained neural networks. Finally, Figure 5.5 is the summary of the classification tests in detecting each tool condition. At each test, there were 15 samples from every tool condition. The bars in Figure 5.5 indicate the numbers of correctly classified samples in each tool condition. The difference in classifying the sharp tool and the tool breakage by two neural networks is not significant. In classifying the tool wear, the MPC fuzzy neural networks give significantly better estimations. The results are detailed in Table 5.3.

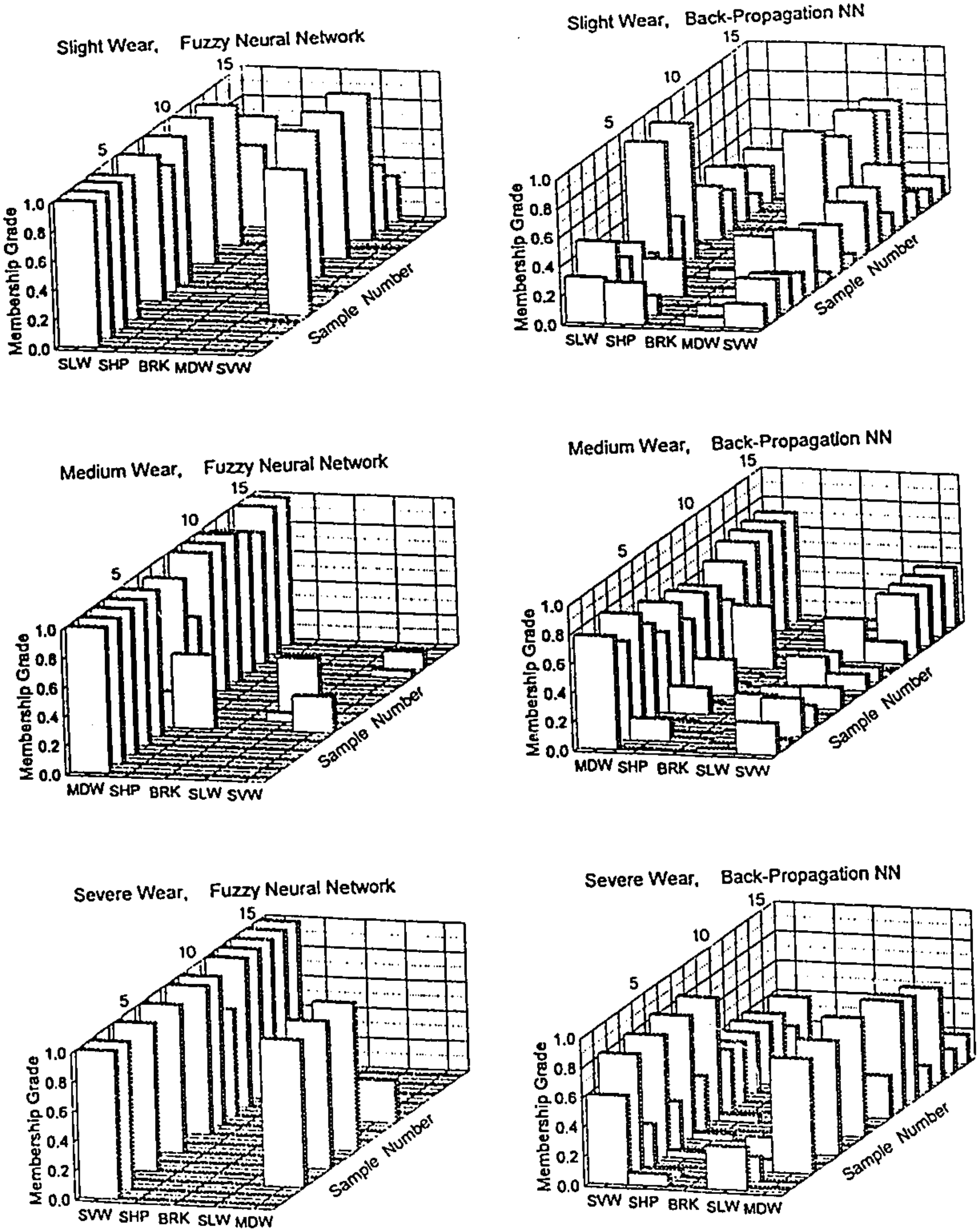


Figure 5.4 Classification Results of the Tool Conditions by Two Neural Networks

Table 5.3 Detailed Results of Classification by Using Two Neural Networks

Test	Network	Sharp		Breakage		Slight W		Medium W		Severe W		Total	
# 2	FNN	15	100 %	15	100 %	11	73.3 %	14	93.3 %	12	80.0 %	67	89.3 %
	BPNN	14	93.3 %	15	100 %	7	46.7 %	15	100 %	9	60.0 %	60	80.0 %
# 4	FNN	13	86.7 %	15	100 %	13	86.7 %	9	60.0%	13	86.7%	63	84.0 %
	BPNN	13	86.7 %	15	100 %	7	46.7 %	7	46.7%	11	73.3%	53	70.7 %
# 6	FNN	15	100 %	13	86.7 %	9	60.0 %	12	80.0 %	11	73.3 %	60	80.0 %
	BPNN	14	93.3 %	13	86.7 %	5	33.3 %	10	66.7 %	10	66.7 %	52	69.3 %

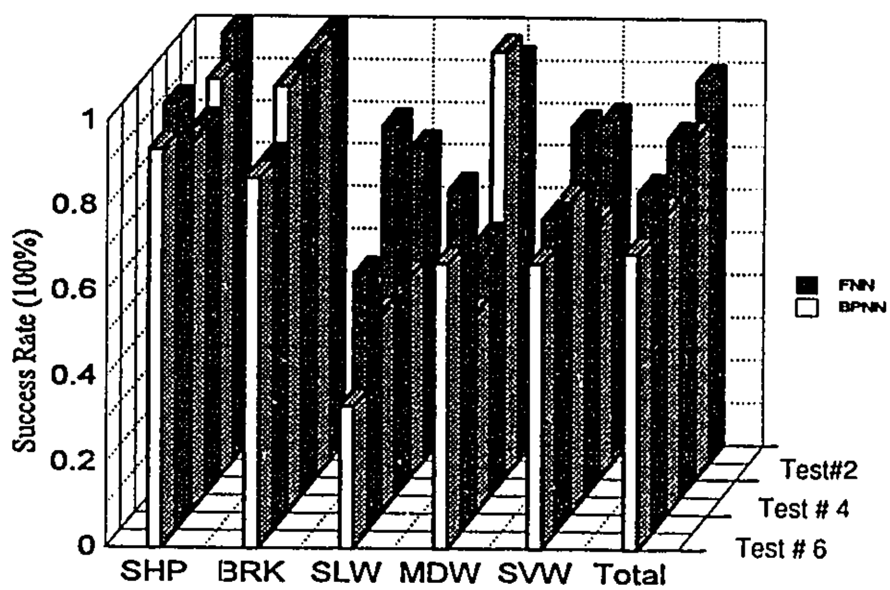
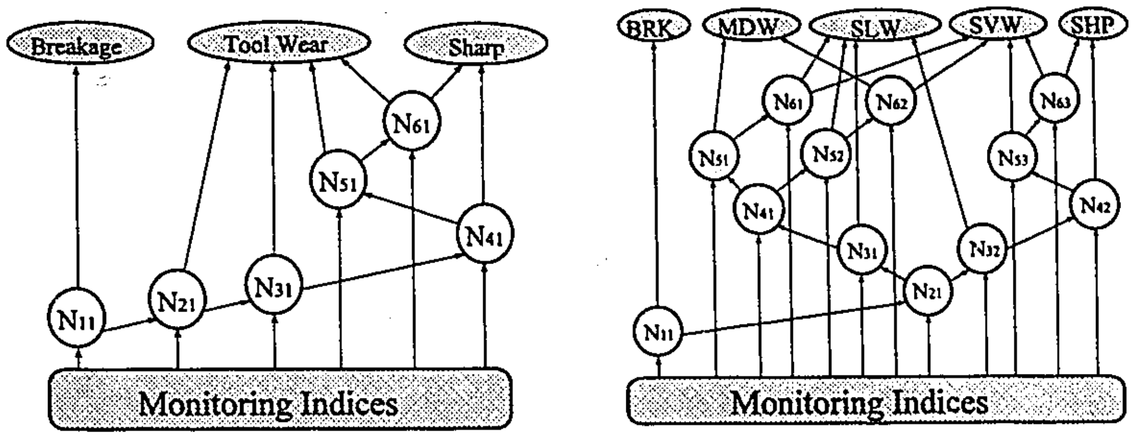


Figure 5.5 Comparison of the Classification Results by the FNN and the BPN

Another advantage of the proposed MPC fuzzy neural networks over the back-propagation neural networks is the short training time. For example, in tests # 1 and #2, the training of a back-propagation neural network required 11 minutes and 21 seconds of CPU time on a PC-486/33 computer. It took approximately one second to train the MPC fuzzy neural network with the same learning data on the same computer. This is a critical issue when the system is used for the on-line monitoring where self-improvement and self-adjustment are needed.

B. Tests for the Unsupervised Classification

The proposed MPC fuzzy clustering neural networks for tool condition monitoring were tested with the experimental data obtained within a range of cutting conditions. Two sets of experimental data, named A and B, were randomly selected. Each set of the data contained 75 samples obtained at the five tool conditions defined in this study (*i.e.*, 15 samples for each condition). The samples were transformed into their principal component directions before being used as the inputs to the networks. This transformation was done to reduce the dimension of the original data vectors. To investigate the effect of cluster numbers on the clustering results, five-class tests and three-class tests were designed. The five-class tests included all the tool conditions mentioned earlier. In the three-class tests, the three stages of tool wear were combined into a single class, so the tool conditions were simply sharp tool, tool breakage, and tool wear. The results showed that the larger the number of clusters is, the more levels in the hidden layer are required, and the more complicated the corresponding network is (refer to Figure 5.6).



(a) A Three-Class Test

(b) A Five-Class Test

Figure 5.6 Examples of the MPC Fuzzy Neural Networks for Clustering

Figure 5.6 shows the fuzzy clustering neural networks which were built during the clustering of the data in set A. Each neuron in the networks receives information from the input layer as well as from other neurons at the preceding level in the hidden layer. The outputs of a neuron are sent to either the output layer and/or other related neurons at the next level. These connections are jointed with the fuzzy membership grades.

The results of clustering the tool conditions in the turning experiments by the MPC fuzzy neural networks are shown in Table 5.4. In the three-class tests, the success rates are approximately 94 % while those of the five-class tests are about 80 % . These results proved the capability of the proposed MPC fuzzy neural networks for dealing with multi-class, overlapped clustering problems such as the clustering of the tool conditions in machining.

Table 5.4 Clustering Results of the Tool Conditions by the Fuzzy Neural Networks

	Five Class Tests						Three Class Tests			
tool condition *	SHP	BRK	SLW	MDW	SVW	Total	Sharp	Break	Wear	Total
Group A	14 / 15	14 / 15	12 / 15	14 / 15	13 / 15	67 / 75	14 / 15	14 / 15	45 / 45	73 / 75
success rate (%)	93.3	93.3	80	93.3	86.7	**89.3**	93.3	93.3	100	**97.3**
Group B	12 / 15	14 / 15	14 / 15	8 / 15	12 / 15	60 / 75	12 / 15	14 / 15	45 / 45	71 / 75
success rate (%)	80	93.3	93.3	53.3	80	**80**	80	93.3	100	**94.7**
A and B	25 / 30	27 / 30	24 / 30	21 / 30	29 / 30	126 / 150	25 / 30	27 / 30	90 / 90	142 / 150
success rate (%)	83.3	90	80	70	96.7	**84**	83.3	90	100	**94.7**

* Class information was not provided for this data until the clusters were formed.

Figure 5.7 illustrates one of the results from these tests, where (a) is the distribution of the original experimental data in the first two principal component directions, (b) is the clustering result of a three-class test, and (c) is that of a five-class test. The experiments showed that the proposed MPC fuzzy neural networks for tool condition monitoring in machining was successful in the classification of the unlabelled testing data.

C. Tests for the Knowledge Updating

Two sets of experimental data, A and B, were generated from the cutting tests. They were taken under the same cutting conditions but from different records. Each set of data contained 75 labelled samples belonging to the five tool conditions considered in this study (*i.e.*, 15 samples from each tool condition). The samples were transformed into the six-dimensional monitoring feature space before they were used for learning and testing.

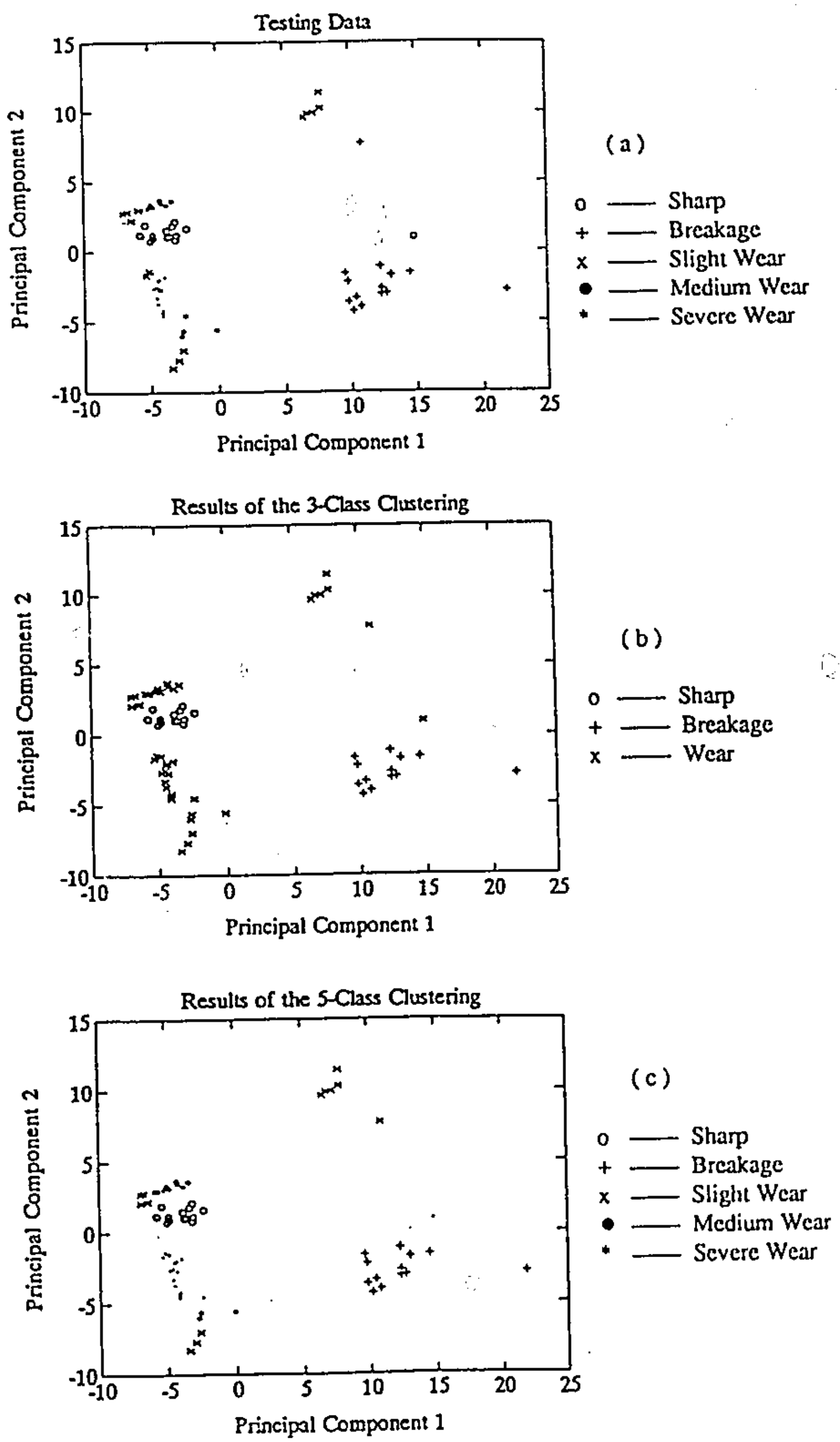


Figure 5.7 Clustering of the Tool Conditions by the MPC Fuzzy Neural Networks

Table 5.5 Classification Results of the Knowledge Updating

Test	Training Set	Retraining Set	Classification Results (with 30 testing samples in each tool condition)						
			SHP	BRK	SLW	MDW	SVW	Total	
1	A	none	27	30	26	25	24	132	88.0 %
2	A & B	none	29	28	27	30	26	140	93.3 %
3	A	B	27	30	30	27	21	135	90.0 %

Three experiments were designed. The first one used set A as the learning samples to train the monitoring system. A and B were then classified through the trained fuzzy neural network. Thc second test used both A and B to train the system, and then a classification test was performed with the same samples as those used for learning. In the third test, the system was first trained using set A, then retrained by set B. Classification was then performed using both A and B sets. The design of these three tests was aimed at providing a comparison of the classification results using the monitoring systems trained by the different learning procedures. As can be seen, the first test is a common classification without retraining. Test two is a self-classification test there all the testing samples are used for the learning. Test three is a classification with retraining. The results of these experiments are given in Table 5.5 and the comparison is illustrated in Figure 5.8.

From these results, it is obvious that the best classification results are given by the all-data-trained (self-classification) system in test two. It is also shown that test three, with knowledge updating, gives better classification results than test one which is without knowledge updating. These experimental results indicate that the proposed MPC fuzzy neural

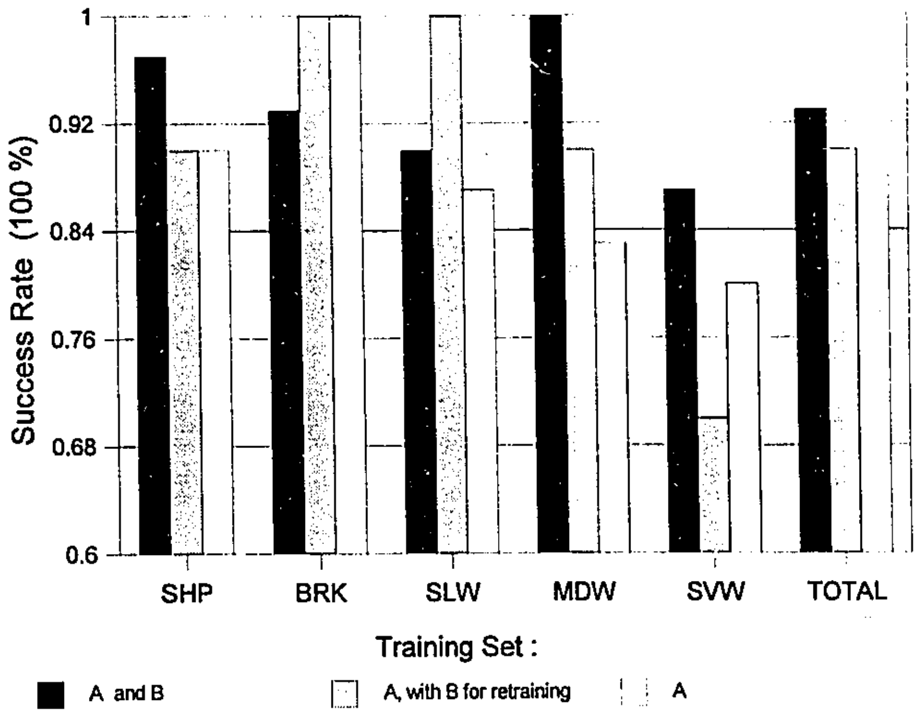


Figure 5.8 Comparison of the Classification Results with Retraining

networks for automated tool condition monitoring has the abilities of self-learning and knowledge updating. The new knowledge in the retraining data is easily added into the system by a simple retraining procedure, and the classification results with the retrained system are improved.

5. 3 DRILLING EXPERIMENTS

5.3.1 Definition of the Tool Conditions

In drilling, the tool wear changes along the cutting edge from the margin to the chisel edge due to the complex geometry of the drill bit and the cutting process. At the drill point, wear occurs at the flute (crater wear), the clearance face (flank wear), the chisel edge, and the margin. Flank wear was mainly considered in this study (refer to Appendix).

The tool conditions in the drilling experiments are recognized as four categories: *Sharp* tool (SHP), *Chipping* (CHP), *Small wear* (SMW) and *Large wear* (LGW). Sharp tool is defined as fresh tools and the tools with a flank wear less than 0.1 mm. Chipping means chipping occurs on the cutting edge. Small wear is defined as the flank wear between 0.1 mm and 0.3 mm, and large wear is defined from 0.3 mm to 0.6 mm. These definitions are shown in Table 5.6

5.3.2 Experimental Setup

The drilling tests were performed on a 5-axis CNC milling machine. 3/16 inch (≈ 4.76 mm) high speed steel drill bits were used for the experiments. The workpiece was medium carbon steel AISI 1045. The depth of the holes were 25 millimetres. Cutting speeds ranged from 12 m/min to 31 m/min, and feedrates ranged from 60 mm/min to 160 mm/min.

Four cutting process signals were measured for the tool condition monitoring. They were the torque, the thrust force and two vibrations in vertical and horizontal directions. The sensor setup is shown in Figure 5.9. A rotating cutting force dynamometer was used for

Table 5.6 Definition of Tool Conditions in Drilling Tests

Tool Conditions	Sharp Tool	Chipping	Small Wear	Large Wear
Tool Features	wear < 0.1 mm	chipping on cutting edge	0.1 < wear < 0.3 mm	0.3 < wear < 0.6 mm

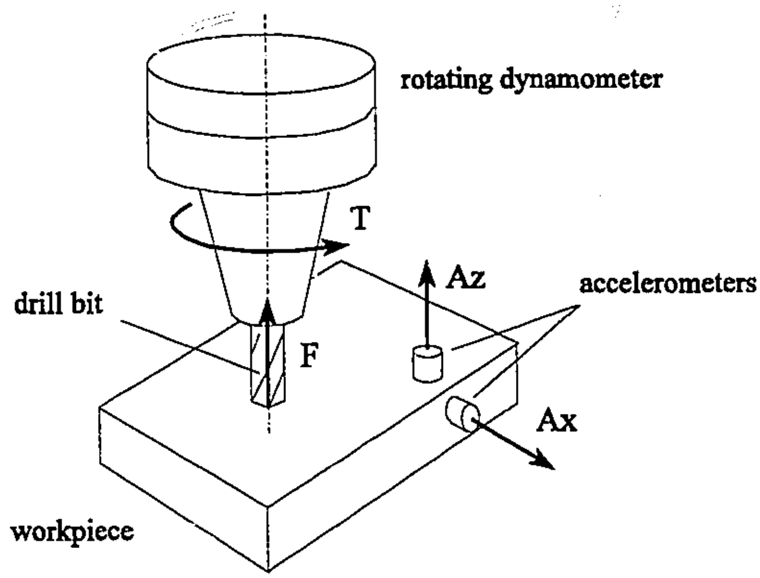


Figure 5.9 Sensor Setup for Drilling Tests

measuring the cutting force and the torque. Two accelerometers were used for measuring the vibrations. The measured signals passed through low-pass filters with the cut-off frequency of 1 kHz, then were sampled at 2 kHz. The signals were sampled and further processed by a PC computer.

5.3.3 Signal Conditioning and Feature Selection

The same signal processing scheme was used for the drilling tests as for the turning tests. The measurements were randomly taken in steady cutting processes for one second. The sampled data during this time period made up the measurement vectors. The measurement vectors represented five signatures in the time domain and seven signatures in the frequency domain. The time-domain signals included the mean values of the cutting force, the torque and the force-torque ratio, as well as the variances of the cutting force, the torque and the vibrations. The frequency-domain signals involved the powers of these measured signals in several specified frequency bands. The frequency range of the analyses was 0 to 400 Hz.

The measurement vectors were further treated with the principal component analyses to generate the feature vectors for learning and classifications.

5.3.4 Experimental Results

Two sets of experimental data were obtained from the drilling tests. Both A and B were randomly generated within the experimental data obtained within the considered range of cutting conditions. Each data set contained 60 samples from the four considered tool conditions, 15 from each condition. After the principal component analysis was completed, three features were extracted from the experimental data to form the feature vectors. All the samples for learning and classification were transferred into these directions.

Three tests were carried out in the experiments for the drilling tool condition monitoring. Test 1 used both sets A and B for training the neural network, which was a self-

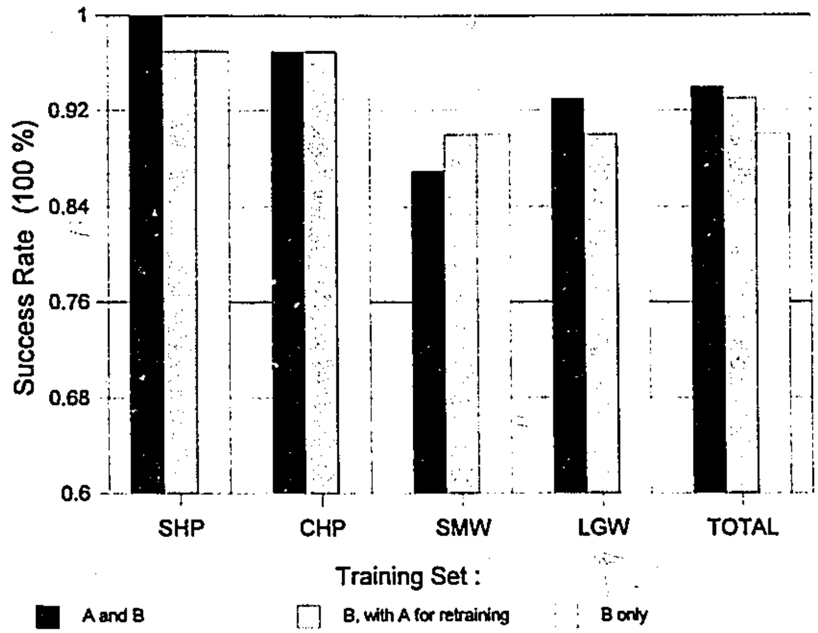


Figure 5.10 Comparisons of the Classification Results for Drilling Tests

classification. Test 2 used only one set, A or B, to training the neural networks, which was a common classification test. Test 3 was a knowledge updating test which used either set A or set B as the learning data, and then used the other set to retrain the system. These tests were designed to evaluate the performance of the MPC fuzzy neural networks for tool condition monitoring in drilling process. The performance with different learning data and the system retraining was also tested. Comparisons of the performance with different learning data are depicted in Figure 5.10. Table 5.7 gives the details of these experimental results. The results have shown the good performance of the proposed MPC fuzzy neural networks for automated tool condition monitoring in drilling tool condition monitoring.

Table 5.7 Experimental Results in Drilling Tests

Test	Training Set	Retraining Set	Classification Results (with 30 testing samples in each tool condition)					
			SHP	CHP	SMW	LGW	Total	
1	A & B	none	30	29	26	28	**113**	**94.2 %**
2	A	none	30	26	24	25	**105**	**87.5 %**
	B	none	29	28	27	24	**108**	**90.0 %**
3	A	B	30	27	25	27	**109**	**90.8 %**
	B	A	29	29	27	27	**112**	**93.3 %**

5. 4 SUMMARY

The experiments in turning and drilling were conducted to test performances of the proposed MPC fuzzy neural networks for automated tool condition monitoring in machining. The tests were designed to verify performances of the system in different cutting processes within a range of cutting conditions. These performances included supervised classification, unsupervised classification and knowledge updating of the system.

Several sensors were used for the monitoring feature selection. Force, torque, vibration, and spindle motor power signals were fused by using the principal component analyses to give a highly sensitive feature space. The tool conditions considered in the monitoring tests included sharp tool, tool breakage (or chipping), and the different states of tool wear. Various cutting conditions were selected during the experiments.

Supervised classification and knowledge updating tests were conducted in both turning and drilling. Unsupervised classification was done in turning. In turning tests, the experimental results showed approximately 94 % success rates in self-classification tests (*i.e.*, the same data samples were used for both learning and classification), 84 % in the tests performed using different records for classification than those used for learning under the same cutting conditions, and approximately 80 % in the tests performed using the samples obtained at different cutting conditions for classification than those used for learning within the same range of cutting conditions. The proposed classification strategies by the fuzzy neural networks performed better than the back-propagation neural networks in these tests. In drilling, the self-classification resulted in a success rate of 94%, and about 88% was obtained in one-set trained classification.

Knowledge updating with only the new learning samples did improve the performance of the system in these tests. About 50% improvement was obtained from the one-set trained system (half of the samples were used for training) comparing to the self-trained system (all the samples were used for training).

CHAPTER VI

CONCLUSIONS

6.1 INTRODUCTION

The main objective of this dissertation is to explore a new approach to automated tool condition monitoring in machining by using fuzzy neural networks. This chapter summarizes the work achieved in developing the Multiple Principal Component (MPC) fuzzy neural networks for automated tool condition monitoring in machining. The possibility of the future work is also suggested.

Statistical pattern classification, neural networks and fuzzy logic have been individually applied in machining process and tool condition monitoring for a long time. Each of these approaches attacks the problem from different angles and has certain limitations. A simple pattern classifier, like a linear classifier, is available for solving the simple problems which are linearly partitioned. Most statistical pattern recognition methods, such as Bayes classifier, need the parameter estimations from a lot of samples. However, the classification based on the distribution and statistical models is still a powerful tool to solve the pattern recognition problems. Neural networks have the abilities of parallel computation, self-learning and non-linearity. They are powerful for solving complex pattern recognition problems. Some algorithms, such as the back-propagation, have more computing work in the training procedures so that the training times are usually significant. Fuzzy classification uses the fuzzy memberships to present similarities of the classes. This representation is very advantageous for solving uncertain and class-overlapped problems, especially in machining process and tool condition monitoring.

"Soft computation" is proposed in the machine intelligence for dealing with approximation and dispositionality (Zadeh, 1993). Its principal constituents include fuzzy logic for imprecision, neural networks for knowledge learning, and probability reasoning for uncertainty. The combination of these three techniques makes the pattern classifiers much more flexible and powerful. Using the soft computation, along with sensor fusion, for the development of automated tool condition monitoring in machining is the major effort and the contribution of this thesis. The proposed system is the Multiple Principal Component (MPC) fuzzy neural networks. The algorithms for supervised classification, unsupervised classification and knowledge updating have been developed. The classification performances of the system have been tested experimentally in turning and drilling.

6.2 CONCLUSIONS FROM THE FINISHED WORK

The original contributions of this work to the development of automated tool condition monitoring in machining can be summarized as follows:

A. System Architecture

Three major components of the "soft computation" are involved in the construction of the proposed fuzzy neural networks. The combination of fuzzy logic with neural networks has a sound technical basis because these two techniques approach the design of intelligent machines from different angles. Fuzzy neural networks employ the advantages of both neural networks and fuzzy logic. Neural networks offer good performance in dealing with sensor information in parallel at a low computational level. The high interconnection within the networks gives the capabilities of exchanging the information sufficiently and managing nonlinearity. Fuzzy logic gives a means for representing , manipulating, and utilizing the data and the information that possess nonstatistical uncertainties.

The proposed tool condition monitoring system is a partially connected neural network with fuzzy classification at the neurons and with the interconnections with fuzzy membership grades. Statistical reasoning is also used in constructing the fuzzy neural networks.

The Multiple Principal Component fuzzy neural network algorithm is a new contribution to the concept of pattern recognitions. Fuzzy neural networks are applied to the tool condition monitoring in machining for the first time.

B. Supervised Classification

A supervised classification algorithm has been developed for the MPC fuzzy neural networks. The "maximum partition" is proposed to give better divisions of the involved classes. All the monitoring indices are examined against all the tool conditions to select the "pivot indices" for the maximum partition. Statistical methods are used for locating the class centres and the fuzzy membership is used for generating "soft" boundaries between the classes. Neurons in the network are generated with the pivot indices, the class centres, the fuzzy membership grades and the connections to other related neurons.

In classification, the fuzzy membership grades of the input pattern are calculated at the neurons. The membership grades define the path directions at each neuron for the classification. The final results can be given with the fuzzy membership grades which measure the uncertainty in the classification.

C. Unsupervised Classification

Unsupervised classification is based on the principal component analysis and the fuzzy memberships to the potential clusters. Two major procedures of the clustering at a neuron in the network are the principal component analysis in multiple directions and the cluster combinations. A fuzzy membership function is defined for the optimum combinations. The fuzzy neural network is built while such procedures are repeated in the learning procedure.

The clustering results of certain published data by the proposed clustering method have shown the good performance comparing to other clustering methods.

D. Knowledge Updating

Knowledge updating is easily implemented with the structure of the MPC fuzzy neural networks. The proposed knowledge updating algorithm deals with two issues: tuning old neurons and adding new neurons. The updating algorithm is developed with the assumption that the system has been trained and it keeps only the necessary information for the classification. New information in the new training data about the classification is combined with the stored information which were learned previously. The old training data are not required. The main structure of the system is not destroyed and only some parameters and minor connections are modified.

The proposed knowledge updating algorithm without the old training data is simple, fast and effective. Knowledge updating improves the system's performance in classification.

E. Sensor Fusion and Feature Extraction

Several sensors are used for selection of the monitoring features. The principal cutting mechanics is studied for choosing the measurement of the process signals which are the most sensitive to changes in the machining tool conditions. Signals in both time domain and frequency domain are used. The signals from multi-sensors of different types at different locations are fused by the principal component analysis to produce the highly information-bearing features (monitoring indices).

The monitoring indices are further analyzed in the learning procedure of the fuzzy neural networks. They are selectively used at each individual neuron in the form of the pivot index. The monitoring features are extracted in both signal processing and learning procedure.

F. Experimental Tests in Turning and Drilling

The experiments for testing the proposed MPC fuzzy neural networks for automated tool condition monitoring in machining were performed in turning and drilling. The proposed algorithms of supervised classification, unsupervised classification and knowledge updating were tested by using the experimental data.

The experimental data for testing the MPC fuzzy neural networks were obtained under various cutting conditions. The data sets were randomly selected from the experimental samples. Different data sets were designed to test the performance of the proposed system in self-learning, in learning with the sample in different time records, and in learning with the samples under different cutting conditions (within the considered range).

The experimental results showed the good performance of the system with these tests. The success rates of self-learning were 94 - 96 %. Those of learning with different time records under the same cutting conditions were 84 - 89 %. The learning with the samples under different cutting conditions gave 80 %. The proposed MPC fuzzy neural networks performed better than the back-propagation neural networks in theses tests. In unsupervised classification, the system gave the success rates of 80 - 97%, depending on the number of classes (tool conditions) pre-defined. Knowledge updating improved the performance of the system by 50 % comparing the success rates between the self-learning (all the samples were used for learning) and the half-learning (half of the samples were used for learning).

Summarizing the above results, we come to the conclusion that the proposed algorithm of the MPC fuzzy neural networks is a fast, effective and simple method for dealing with multi-sensor, multi-class, overlapped classification problems. The system has been successfully applied for the machining tool condition monitoring in turning and drilling.

6. 3 SUGGESTIONS FOR FUTURE WORK

Due to recognizable limitations, this single thesis cannot tackle all the aspects of a fully developed automated/intelligent machining process and tool condition monitoring system, by using only the fuzzy neural network approaches. The following recommendations are given here to enhance the performance and capabilities of the proposed system:

A. Expert System and Modelling Decision

Introduction of some well-developed knowledge-base (expert) systems and models in the monitoring system to create some direct routines (or "short cuts") will improve the system's performance and speed up the decision making process. A knowledge base of metal cutting mechanics can be added for better feature selection and decision making strategies for different cutting processes, different tool geometries, different materials, and different cutting condition ranges. Sensor fusion (or sensor synthesis) through multiple models can also be introduced into the system for this purpose.

B. Further Developments of Fuzzy Decision Making Strategies

Due to time limits, the attempt to use fuzzy classification in neural networks for the tool condition monitoring in machining is done only with the fuzzy *c*-means method. Other fuzzy classification algorithms, such as fuzzy min-max, fuzzy if-then rules, and the like, can also be used to optimize the performance of a monitoring system. Other architectures of neural networks may be also worth of exploration.

C. Communication with Control System

Automated tool condition monitoring is only a part of the work in manufacturing automation. Monitoring systems have to be connected to the control system to play their full roles. The communication with the control systems or other monitoring systems is necessary. The on-line implementation of the tool condition monitoring system still needs to be further developed.

The fuzzy neural networks for automated tool condition monitoring in machining is generally a rough and rudimentary idea. The basis of this idea is to combine a few simple and advanced data processing methods and to utilize their advantages in different aspects. Numerous ways of merging different advanced technologies exist. It is the author's intention that the proposed MPC fuzzy neural network algorithm works like a trigger to help exploring the applications of various new techniques in automated/intelligent process and tool condition monitoring in machining.

REFERENCES

Andrews, H. C., 1972, Introduction to Mathematical Techniques in Pattern Recognition, Wiley-Interscience.

Armarego, E. J. A. and Brown, R. H., 1969, The Machining of Metals, Prentice-Hall Inc.

Bezdek, J. C., 1981, "Pattern Recognition with Fuzzy Objective Function Algorithms," Plenum Press, New York.

Bezdek, J. C., 1987, "Analysis for Fuzzy Information," Vol. III, CRC Press, Inc.

Bezdek, J. C., 1993, "Fuzzy Models — What Are They, and Why ?" *IEEE Transaction on Fuzzy Systems*, Vol. 1, No. 1, pp. 1-6.

Boothroyd, G., 1975, Fundamentals of Metal Machining and Machine Tools, Scripta Book Company.

Caudill, M., 1989, "Neural Networks Primer," *Special Issue of AI EXPERT*, Miller Freeman Publications.

Chen, Y., 1993, "A Fuzzy Decision System for Fault Classification under High Levels of Uncertainty," *Manufacturing Science and Engineering*, ASME PED-Vol. 64, pp. 359-368.

Chryssolouris, G. and Domroese, M., 1988, "Sensor Integration for Tool Wear Estimation in Machining," *Sensors and Control for Manufacturing*, ASME PED-Vol. 33, pp. 115-123.

Chryssolouris, G. and Domroese, M., 1989, "An Experimental Study of Strategies for Integrating Sensor Information in Machining," *Annals of the CIRP*, Vol. 38, No. 1, pp. 425-428.

Chryssolouris, G., Domroese, M. and Beaulieu, P., 1992, "Sensor Synthesis for Control of Manufacturing Process," *ASME J. of Eng. for Industry*, Vol.114, pp. 158-174.

Chryssolouris, G. and Guillot, M., 1988, "An A.I. Approach to the Selection of Process Parameters in Intelligent Machining," *Sensors and Control for Manufacturing*, ASME PED-Vol. 33, pp. 199-206.

Chryssolouris, G. and Guillot, M., 1990, "A Comparison of Statistical and AI Approaches to the Selection of Process Parameters in Intelligent Machining," *ASME J. of Eng. for Industry*, Vol.112, pp. 121-131.

Chung, E. S., Chiou, Y. S. and Liang, S. Y., 1993, "Tool Wear and Chatter Detection in Turning via Time-Series Modelling and Frequency Band Averaging," *Manufacturing Science and Engineering*, ASME PED. Vol.64, pp. 351-358.

Cover, N., 1965, "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition," *IEEE Trans. on Computers*, Vol. 14, pp. 326-334.

Den, L. and Mathew, J., 1990, "Tool Wear and Failure Monitoring Techniques for Turning — a Review", *Int. J. of Mach. Tool and Manuf.*, Vol. 30, No. 4, pp. 579-598.

Diei, E. N. and Dornfeld, D. A., 1987, "A Model of Tool Fracture Generated Acoustic Emission During Machining," *ASME J. of Eng. for Industry*, Vol.109, pp. 207-233.

Dornfeld, D. A., 1986, "Acoustic Emission Monitoring for Untended Manufacturing," *Proc. of Japan/USA Symp. on Flexible Automation*, JAACE.

Dornfeld, D. A., 1990, "Neural Network Sensor Fusion for Tool Condition Monitoring," *Annals of the CIRP*, Vol.39, No.1, pp. 101-105.

Dornfeld, D. A., 1992, "Monitoring of Machining Process — Literature Review," presented at *CIRP STC"C" meeting*, Paris, France.

Dornfeld, D. A. and Pan, C. S., 1985, "Determination of Chip Forming States Using A Linear Discriminant Function Technique with Acoustic Emission," *Proc. of 13th NAMRC*, ASME, pp. 299-303.

Doyle, L. E., Keyser, C. A., Leach, J. L., Schrader, G. F. and Singer, M. B., 1969, Manufacturing Processes and Materials for Engineers, 2nd Ed., Prentice-Hall, Inc.

Du, R. X., Elbestawi, M. A. and Li, S., 1992, "Tool Condition Monitoring in Turning Using Fuzzy Set Theory," *Int. J. of Mach. Tool and Manuf.*, Vol. 32, No. 6, pp. 781-796.

Du, R., Elbestawi, M. A. and Wu, S. M., 1995, "Automated Monitoring of Manufacturing Processes, Part 1: Monitoring Methods," *ASME J. of Eng. for Industry*, Vol.117, pp. 121-132.

Elanayar, S. V. T., Shin, Y. C. and Kumara, S., 1990, "Machining Condition Monitoring for Automation Using Neural Networks," *Monitoring and Control for Manufacturing Processes*, ASME PED. Vol.44, pp. 85-100.

Emel, E., 1991, "Tool Wear Detection by Neural Network Based Acoustic Emission Sensing," *Control of Manufacturing Processes*, ASME PED. Vol.52, pp. 79-85.

Emel, E. and Kannatey-Asibu, E. Jr., 1987, "Tool Failure Monitoring in Turning by Pattern Recognition Analysis of AE Signals," *Sensors for Manufacturing*, ASME PED. Vol.30, pp. 39-57

Emel, E. and Kannatey-Asibu, E. Jr., 1988, "Tool Failure Monitoring in Turning by Pattern Recognition Analysis of AE Signals," *ASME J. of Eng. for Industry*, Vol.112, pp. 137-145

Endres, W. J., Sutherland, J. W., DeVor, R. E. and Kapoor, S. G., 1990, "A Dynamic Model of the Cutting Force System in the Turning Process," *Monitoring and Control for Manufacturing Process*, ASME PED. Vol.44, pp. 193-212.

Fisher, D. H. and McKusick, K. B., 1988, An Empirical Comparison of ID3 and Back-Propagation, Tech,. Rep. TRCS-88-14, Dept. of Comp. Sci., Vanderbilt Univ., Nashville, TN.

Govekar, E. and Grabec, I., 1994, "Self-Organizing Neural Network Application to Drill Wear Classification," *ASME J. of Eng. for Industry*, Vol.116, pp. 233-238.

Gupta, T. and Ghosh, B. K., 1988, "A Survey of Expert Systems in Manufacturing and Process Planning," *Computers in Industry*, Vol.11, pp.195-204.

Horowily, E. and Sahni, S., 1976, Fundamentals of Data Structures, Computer Sci. Press Inc.

Höskuldsson, A., 1988, "PLS Regression Methods," *J. of Chemometrics*, Vol.2, pp.211-228.

Isermann, R., 1984, "Process Fault Detection Based on Modeling and Estimation Methods — A Survey," *Automatica*, Vol.20, No.4, pp. 387-404.

Iwata, K., 1988, "Application of Expert Systems to Manufacturing in Japan," *Int. J. of Advanced Manufacturing Technology*, Vol. 3(3), pp.23-37.

Iwata, K. and Moriwaki, T., "An Application of Acoustic Emission Measurement to In-Process Sensing of Tool Wear," *Annals of the CIRP*, Vol.25, No.1, pp. 21-26.

James, M., 1985, Classification Algorithm, William Collins Sons & Co. Ltd.

Kaldor, S. and Lenz, E., 1980, "Investigation in Tool Life of Twist Drills," *Annals of the CIRP*, Vol.29, No.1, pp. 23-27.

Kanai, M., Iwata, K., Fujii, S., Kanda, Y., 1978, "Statistical Characteristics of Drill Wear and Drill Life for the Standardized Performance Tests," *Annals of the CIRP*, Vol.27, No.1, pp. 61-66.

Kannatey-Asibu, E. Jr., 1982, "On the Application of the Pattern Recognition Method to Manufacturing Process Monitoring," *Proc. 10th NAMRC*, pp. 487-492.

Kannatey-Asibu, E. Jr., 1985, "A Transport-Diffusion Equation in Metal Cutting and its Application to Analysis of the Rate of Flank Wear," *ASME J. of Eng. for Industry*, Vol.107, pp. 81-89.

Kegg, R., 1984, "On-Line Machine and Process Diagnostics," *Annals of the CIRP*, Vol.33, No.2, pp. 469-473.

Kim, K. J., Eman, K. F. and Wu, S. M., 1982, "Model Analysis of Mechanical Structures by Time Series Approach," *Proc. of the 10th NAMRC*, pp. 417-421.

Kishi, N, 1986, "Trends of Artificial Intelligence Applications in Automotive Industry — Trouble-Shooting," *Journal of JSME*, 89(815), pp.1189.

Ko, T. J. and Cho, D. W., 1994, "Tool Wear Monitoring in Diamond Turning by Fuzzy Pattern Recognition," *ASME J. of Eng. for Industry*, Vol.116, pp. 225-232.

Ko, T. J. and Cho, D. W., 1994, "Cutting State Monitoring in Milling by a Neural Network," *Int. J. of Mach. Tool and Manufact.*, Vol.34, No.5, pp. 659-676.

Ko, T. J., Cho, D. W. and Lee, J. M., 1992, "Fuzzy Pattern Recognition for Tool Wear Monitoring in Diamond Turning," *Annals of the CIRP*, Vol.41, No.1, pp. 125-128.

Koren, Y., 1978, "Flank Wear Model of Cutting Tools Using Control Theory," *ASME J. of Eng. for Industry*, paper No. 77-WA/PROD-20.

Koren, Y., Danai, K., Ulsoy, A. G. and Ko, T. R., 1987, "Monitoring Tool Wear Through Force Measurement," *Proc. of 15 NAMRC*, pp. 463-468.

Koren, Y., Ko, T. R., Ulsoy, A. G. and Danai, K., 1991, "Flank Wear Estimation Under Varying Cutting Conditions," *ASME J. of Eng. for Industry*, Vol.113, pp. 300-307.

Kumar, G. S. and Ernst, G. W., 1987, "An Expert System Architecture for Predictive Monitoring," *Intelligent and Integrated Manufacturing Analysis and Synthesis*, ASME PED-Vol.25, pp. 121-128.

Lai, C. T., [illegible], The Influence of Tool Wear and Breakage on Forces in Bar Turning, M Eng. thesis, McMaster University.

Li, G. S., Lau, W. S. and Zhang, Y. Z., 1992, "In-Process Drill Wear and Breakage Monitoring for a Machining Centre based on Cutting Force Parameters," *Int. J. of Mach. Tool and Manuf.*, Vol. 32, No. 6, pp. 855-867.

Li, P. G. and Wu, S. M., 1988, "Monitoring Drill Wear States Using a Fuzzy Pattern Recognition Technique," *ASME J. of Eng. for Industry*, Vol. 110, No. 2, pp. 297-300.

Li, S. and Elbestawi, M. A., 1994, "Tool Condition Monitoring in Machining by Fuzzy Neural Networks," *Dynamic Systems and Control,* ASME DSC-Vol. 55-2, pp. 1019-1034.

Li, S. and Elbestawi, M. A., 1995a, "Fuzzy Clustering for Automated Tool Condition Monitoring in Machining," to appear, *J. of Mechanical Systems and Signal Processing.*

Li, S. and Elbestawi, M. A., 1995b, "Knowledge Updating for Tool Condition Monitoring in Machining," *Dynamic Systems and Control,* ASME DSC-Vol. 57-2, pp. 1063-1071.

Li, S., Elbestawi. M. A. and Du, R. X., 1992, "A Fuzzy Logic Approach for Multi-Sensor Process Monitoring in Machining," *Sensors and Signal Processing for Manufacturing*, ASME PED-Vol. 55, pp. 1-16.

Liang, S. Y. and Dornfeld, D., 1989, "Tool Wear Detection Using Time Series Analysis of Acoustic Emission," *ASME J. of Eng. for Industry,* Vol.111, pp. 199-205.

Liao, Y. S., 1986, "Development of a Monitoring Technique for Tool Change Purpose in Turning Operations," *Proc. of the 26th Int. Machine Tool Design and Research Conf.*, pp. 331-337.

Lippmann, R. P., 1987, "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, April. 1987, pp. 4-22.

Lippmann, R. P., 1989, "Pattern Classification Using Neural Networks," *IEEE Communication Magazine*, Nov. 1989, pp. 47-64.

Liu, T. I. and Ko, E. J., 1990, "On-Line Recognition of Drill Wear via Artificial Neural Networks," *Monitoring and Control for Manufacturing Process*, ASME PED-Vol.44, pp. 101-110.

Liu, T. I. and Wu, S. M., 1988, "On-Line Drill Wear Monitoring," *Sensors and Control for Manufacturing*, ASME PED-Vol. 33, pp. 99-104.

Liu, T. I. and Wu, S. M., 1990, "On-Line Detection of Drill Wear," *ASME J. of Eng. for Industry*, Vol.112, pp. 299-302.

Lu, W. F. and Klamecki, S. E., 1990, "Prediction of Chatter Onset in Turning with a Modified Chatter Model," *Monitoring and Control for Manufacturing Process*, ASME PED- Vol.44, pp. 237-252.

Marks, J. and Elbestawi, M. A., 1988, "Development of Machine Tool Condition Monitoring System Using Pattern Recognition," *Sensors and Controls for Manufacturing*, ASME PED- Vol.33, pp. 89-98.

Matsumoto, Y., Tjinag, N. and Foote, B., 1988, "Tool Wear Monitoring Using A Linear Extrapolation," *Sensors and Control for Manufacturing*, ASME PED- Vol. 33, pp.83-88.

McClelland, S., 1988, "Tacking the Problem of Sensor Fusion," *Sensor Review*, Vol.8, No. 2, pp. 89-91.

Merchant, M. E., 1945, "Mechanics of the Metal Cutting Process," *J. of Appl. Phys.*, Vol.16, No.5, pp.318.

Mesina, O. S. and Langari, R., 1994, "A Neuro-Fuzzy System for Tool Condition Monitoring in Metal Cutting," *Dynamic Systems and Control,* ASME DSC-Vol. 55-2, pp. 931-938.

Pham, D. T. and Pham, P. T. N., 1988, "Expert Systems in Mechanical and Manufacturing Engineering," *Int. J. of Advanced Manufacturing Technology*, Vol. 3(3), pp.3-21.

Pramod, B. R. and Bose, S. C., 1993, "System Identification Using ARMA Modelling and Neural Networks," *ASME J. of Eng. for Industry*, Vol.115, pp. 487-491.

Ramamurthi, K. and Hough, C. L. Jr., 1993, "Intelligent Real-Time Predictive Diagnostics for Cutting Tools and Supervisory Control of Machining Operations," *ASME J. of Eng. for Industry*, Vol. 115, pp. 268-277.

Rangwala, S. and Dornfeld, D., 1987, "Integration of Sensors via Neural Networks for Detection of Tool Wear States," *Intelligent and Integrated Manufacturing Analysis and Synthesis*, ASME PED- Vol.25, pp. 109-120.

Rangwala, S. and Dornfeld, D., 1990, "Sensor Integration Using Neural Networks for Intelligent Tool Condition Monitoring,"*ASME J. of Eng. for Industry*, Vol.112, pp. 219-228.

Rich, E., 1983, Artificial Intelligence, McGraw-Hill Book Company.

Ruokangas, C. C., Blank, M. S., Matin, J. F. and Schoenwald, J. S., 1986, "Integration of Multiple Sensors to Provide Flexible Control Strategies," *Proc. 1986 IEEE Int. Conf. on Robotics and Automation*, pp. 947-953.

Ruspini, E., 1981, "Numerical Methods for Fuzzy Clustering," *Information Science*, Vol.6, pp. 273-284.

Simpson, P. K., 1992, "Fuzzy Min-Max Neural Networks — Part 1: Classification," *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, pp 776-786.

Simpson, P. K., 1993, "Fuzzy Min-Max Neural Networks — Part 2: Clustering," *IEEE Transactions on Fuzzy Systems*, Vol. 1, No. 1, pp 32-45.

Spiewak, S. and Wu, S. M., 1988, "Tool Wear Monitoring and Breakage Detection Based on 'Intelligent Filtering'," *Int. J. of Mach. Tool and Manuf.*, Vol. 28, No. 4, pp. 483-494.

Stein, J. L., Colvin, D., Clever, G. and Wang, C. H., 1986, "Evaluation of DC Servo Machine Tool Feed Drives as Force Sensors", *ASME J. of Dynamic Systems, Measurement, and Control*, Vol.108, pp. 279-288

Takata S. and Sata, T., 1986, "Model Referenced Monitoring and Diagnosis -- Application to the Manufacturing System," *Computers in Industry*, Vol.7, pp. 31-43.

Tansel, I. N., Mekdeci, C., Rodriguez, O. and Uragun, B., 1993, "Monitoring Drill Conditions with Wavelet Bassed Encoding and Neural Networks," *Int. J. of Mach. Tool and Manuf.*, Vol. 33, No. 4, pp. 559-575.

Tansel, I. N., Mekdeci, C., and McLaughlin, C., 1993, "Detection of Tool Failure in End Milling with Wavelet Transformations and Neural Networks (WT-NN)," *Manufacturing Science and Engineering* , ASME, pp. 369-374.

Tlusty, J., 1978, "Analysis of the State of Research in Cutting Dynamics," *Annals of the CIRP*, Vol.27, No.2, pp. 583-589.

Tlusty, J. and Andrews, G. C., 1983, "A Critical Review of Sensors for Unmanned Machining," *Annals of the CIRP*, Vol.32, No.2, pp. 563-572.

Tsai, S. Y., Eman, K. F. and Wu, S. M., 1983, "Chatter Suppression in Turning," *Proc. of 11th NAMRC*, pp. 399-402.

Tönshoff, H. K. and Wulfsberg, J. P., 1988, "Developments and Trends in Monitoring and Control of Machining Processes," *Annals of the CIRP*, Vol.37, No.2, pp. 611-622.

Wang, M., Zhu, J. Y., and Zhang Y. Z., 1985, "Fuzzy Pattern Recognition of the Metal Cutting States," *Annals of the CIRP*, Vol.34, No.1, pp. 133-136.

Whitney, P. K., 1971, "A Direct Method of Non_parametric Measurenment Selection," *IEEE Trans. on Computer*, Vol. 20, pp. 1100-1103.

Woolbury, M. A. and Clive, J. A., 1974, "Clinical Pure Types as a Fuzzy Partition," J. Cybern., Vol. 4-3, pp. 111-121.

Wu, S. M., 1977, "Dynamic Data System: A New Modelling Approach," *ASME J. of Eng. for Industry*, Vol.99, pp. 708-714.

Wu, S. M., Tobin, T. H. Jr. and Chow, M. C., 1980, "Signature Analysis for Mechanical Systems via Dynamic Data Systen (DDS) Monitoring Technique," *ASME J. of Eng. for Industry*, Vol.102, pp. 217-221.

Yang, W. Q., Hsieh, S. H. and Wu, S. M., 1982, "Adaptive Modelling and Characterization for Machining Chatter," *Measurement and Control for Batch Manufacturing*, ASME, pp. 135-144.

Zadeh, L. A., 1965, "Fuzzy Sets," *Inform. and Control.* Vol. 8, pp. 338-353.

Zadeh, L. A., 1973, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Transactions on System, Man, and Cybernetics*, Vol. SMC-3, pp 28-44.

Zadeh, L. A., 1993, "Fuzzy Logic, Neural Networks and Soft Computation," lecture seminar at University of Guelph, Ont., Canada

Zhang, Y. Z., Liu, Z. F, Pan, L. X., Liu, Y. J. and Yang, W. B., 1982, "Recognition of the Cutting States for the Difficult-to-Cut Materials Application of the Pattern Recognition Technique," *Annals of the CIRP*, Vol.31, No.1, pp. 97-101.

APPENDIX

SIGNAL FEATURES FOR MACHINING PROCESS AND TOOL CONDITION MONITORING

A. 1 PROCESS AND TOOL CONDITIONS

A.1.1 Turning Operation

A typical cylindrical turning operation is illustrated in Figure A.1. While studying the cutting force in turning, it is convenient to consider three mutual-perpendicular force components: one parallel with the cutting velocity, F_c, called the cutting force which is the power contributing force, one along feed direction, F_f, called the feed force, and the third perpendicular to the finished work surface, F_r, called the radial force. These three force components are shown in Figure A.1. The tangential component is the cutting force. The

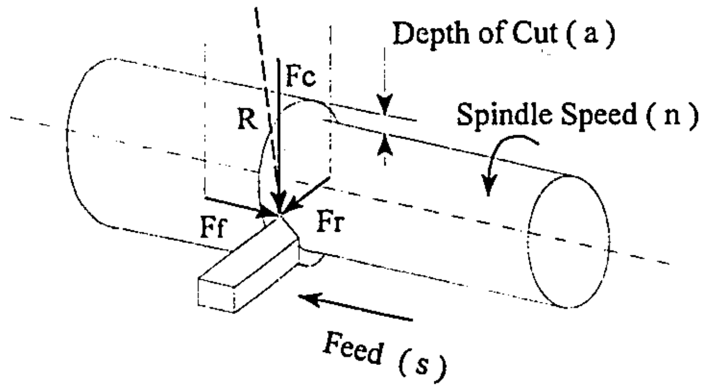


Figure A.1 Turning Operation and the Cutting Force Components

longitudinal component is the feed force. These three force components are of the major interests in machining process and tool condition monitoring.

A.1.2 Drilling Operation

Drilling involves feeding a rotating cutting tool along its axis into a stationary workpiece. Twist drills with two cutting edges are usually used for drilling. The feed velocity is always small compared to the peripheral velocity of the drill. The coordination of these two movements makes the tool cut into the workpiece and a straight hole is made. Figure A.2 depicts this operation and the related forces. *Spindle speed*, *n*, is the number of revolutions that the tool turns in a minute. The spindle speed, together with the diameter of the tool, defines the *cutting speed*. *Feed* in drilling, *s*, is the distance the tool moves toward the workpiece along the spindle axis for one rotation of the tool.

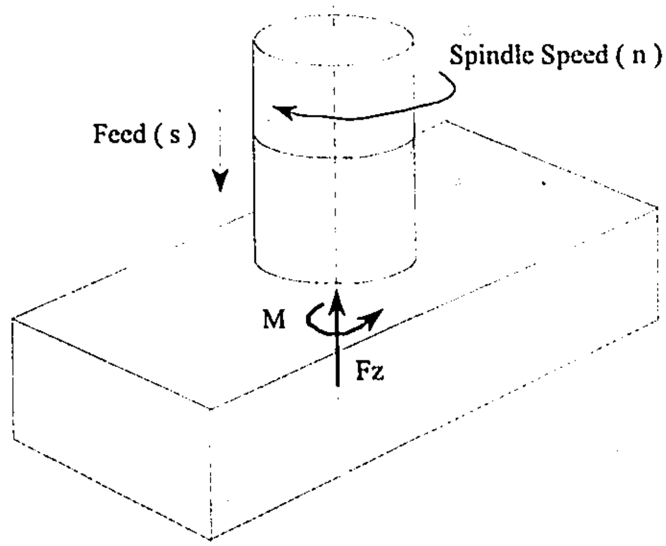


Figure A.2 Drilling Operation and the Cutting Force Components

In study of the cutting forces in drilling, the torque and the thrust force are the most important features. As shown in Figure A.2, the torque is the major power consuming component. It is determined by the cutting force along the tangential direction and the radius of the drill bit. The thrust force is created by the movement of feeding the drill bit into the workpiece.

A.1.3 Chatter

An important practical problem in metal cutting is machining vibration. It can be classified as two types: forced vibration produced, for example, by the force fluctuations in the cutting process and chatter. Chatter is a self-induced vibration existing between the tool

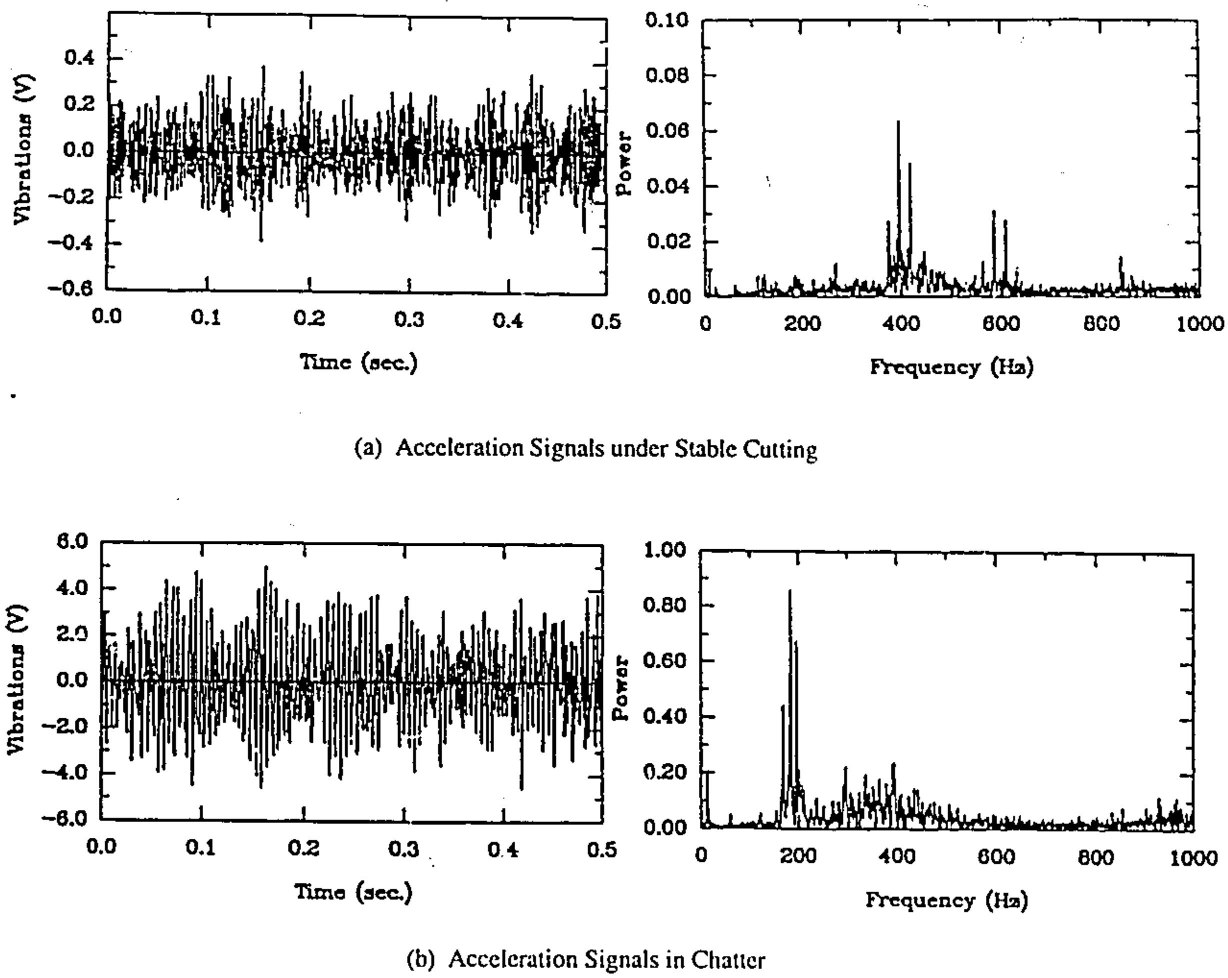


Figure A.3 Acceleration Signals under Stable Cutting and Chatter

and the workpiece in the cutting process. They may produce imperfections on the workpiece surface and also may increase the rate of tool wear or damage the machine. Chatter will leave chatter marks on the machined workpiece surface and will result in chatter noise. These two characters can be used to identify the occurrence of chatter.

When a vibration is measured by accelerations, the signals are generally a zero-mean time series. Under normal cutting conditions, the acceleration signal is essentially a white noise since there is no predominant vibration mode. Under chatter conditions, however, the signals are dominated by structural vibration modes. The structural modes are easily seen on the spectrum of the signal. Figure A.3 shows typical vibration signals in normal cutting condition and in chatter respectively. The spectrums of these signals are also given. The differences in both time and frequency domains are noticeable.

A.1.4 Tool Life Criteria

During metal cutting, cutting tools are subjected to high pressures, severe frictions and high temperatures. Tools most often fail by either sudden breakage or gradual wear. A tool life criterion is generally defined as a predetermined threshold value of a tool wear measure or the occurrence of a phenomenon (Boothroyd, 1975). A tool life criterion depends on the requirements of the components being produced. In a roughing operation, surface finish and dimensional accuracy are less important than the level of cutting forces and power required, so a threshold for excessive rise in forces should be set up. The surface finish and dimensional accuracy is of major importance in a finishing operation, therefore, tools will fail when the specified conditions can no longer be achieved.

The length of a tool life will depend on the chosen criterion, which mainly relies on the requirement of the products. In most of cases, tools fail either by abrupt failures which bring a premature end of the tool life or by gradual or progressive wear on tool rake and flank faces.

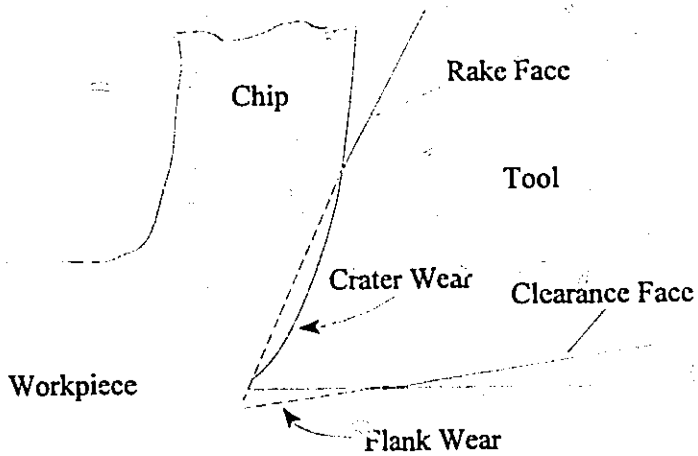


Figure A.4 Wear on Cutting Tool

A.1.5 Tool Wear

There are two major types of wear on a cutting tool: *crater wear* and *flank wear*. Crater wear occurs on the tool rake face, which is characterized by the formation of a crater and created by the action of the chip flowing along the tool face. The maximum depth of the crater is usually taken as a measure of the amount of crater wear. Flank wear occurs on the clearance face of a tool. The wear is quantified by the width of the wear land. These two types of wear are shown in Figure A.4.

Tool wear affects the performance of cutting tools in various ways (Armarego and Brown, 1969; Boothroyd, 1975). The cutting forces are normally increased by tool wear, even though at certain circumstances, crater wear may reduce forces by effectively increasing the tool rake angle. The wear on the clearance face almost invariably increases the cutting forces due to increased rubbing forces. A wear land also increases the tendency of a tool to dynamic instability, and vibrations or chatter can be induced. Another result of tool wear is the complete removal of the cutting point. This may come about by temperature rise, which virtually causes the tool tip to be softened until it comes to plastic deformation. A catastrophic failure which may take place is mechanical fracture of a relatively large portion of the cutting tip.

Tool wear increases the cutting forces and there is a consistently linear relationship between the cutting force components and the flank wear (Lai, 1986). There exists a strong correlation between the cutting force components and wear land, *VB*, in different degrees. Axial force(feed force) is the most sensitive. Radial force(thrust force) and tangential force(cutting force) are the second and the third, respectively.

It has been shown that the force ratio of the horizontal resultant force to the tangential force, $\frac{F_T}{F_C}$, is a good signature feature(monitoring index) for on-line monitoring of the tool wear. Figure A.5 from Lai (1986) illustrates the influence of flank wear on the force ratio.

Researchers also observed that the total amount of high-frequency vibration energy increased with increase in the wear-land. This was attributed to additional vibrations generated by the greater contact area between the worn cutting edge and the workpiece.

There are a few patterns of wear on a twist drill. Apart from flank wear and crater wear, there are corner wear, chisel wear and land wear, as shown in Figure A.6. The wear pattern changes along the cutting edge from the margin to the chisel edge due to complex geometry of the drill bit and the cutting process. Each of them may be correlated with a specific combination of cutting parameters. Of these, the outer corner wear is considered the

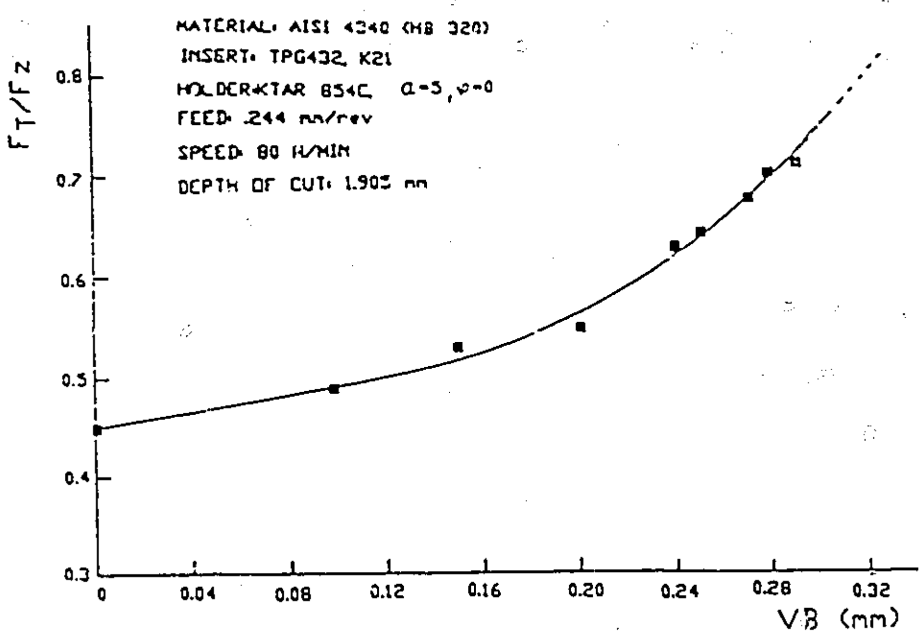


Figure A.5 Influence of Flank Wear on the Force Ratio / Lai, 1986 /

most dominant in the determination of the useful life of a drill under normal cutting conditions (Kanai *et al*, 1978; Kaldor and Lenz, 1980). Flank wear is also used as a criterion for tool wear detection (Liu and Wu, 1990).

In drilling, tool wear affects both the thrust force and the torque. Some researchers reported that serious tool wear might lead to 50% of observable wear or even more in amplitude of the both force parameters (Li *et al*, 1992). Dynamic components of the force signal have a close relationship with the tool wear condition. Tool wear will affect the energy of the signals, so that vibrations of the drill are also used for the tool wear monitoring signal. Since tool wear changes the geometry of the drill bit, the ratio of the force parameters, say $\frac{F_Z}{M}$, is also very sensitive to tool wear.

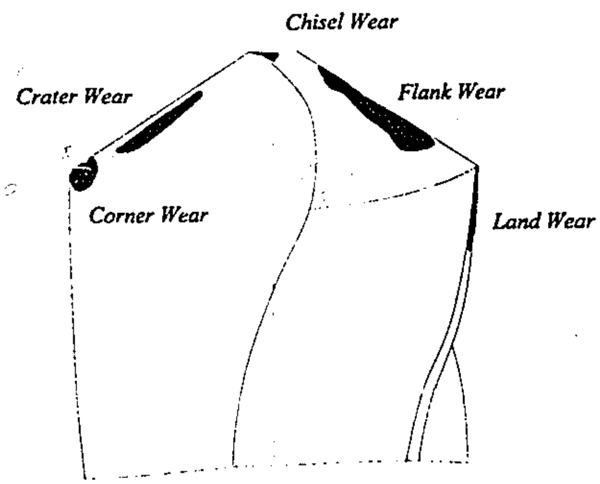


Figure A.6 Wear Patterns on Twist Drill

A.1.6 Tool Breakage and Chipping

Chipping of the cutting tool edge and breakage are important types of tool failure. Breakage and chipping being brittle fractures involve the development and the propagation of micro-cracks in tool material, which normally originate at a critical point where the stress state is such that it causes local rupturing of the inter-atomic bonds (Lai, 1986). Sudden loads and transient thermal stresses may also cause fracture of the tool. Chipping and breakages are usually transient and the whole process of a breakage signal occurs within several tens or hundreds of milliseconds.

The end of useful life of a tool is usually determined by one of the two criteria: excessive wear, and fracture or chipping. Failure due to excessive wear is commonly observed in turning and with large size drills (> 3 mm). Fracture of a tool is encountered with small size drills (< 3 mm).

In the event of extensive breakage, the force components change significantly. At the moment of breakage, in turning, the chip forming is stopped with consequent drop in the main cutting force and both of the feed force and the radial force increase rapidly because the broken tool is squeezing through between the workpiece and the main tool body. After that, the force components remain low as only a small portion of the broken tool comes in contact with the workpiece. When drill breakage occurs, both the thrust force and the torque drop suddenly in amplitude and then go up to a relatively stable level. The loss of the cutting edge or the chipping on it changes the force components. It was reported that the force ratio could be increased by 400% (Lai, 1986). Therefore, the force ratio is a powerful and effective sensing element in a tool breakage algorithm.

A. 2 SIGNAL FEATURES

A.2.1 Forces and Torque

Cutting force is one of the parameters that can be relatively easy to measure for the tool condition monitoring. Cutting forces change as the tool wears and have often been used to detect tool wear. Force sensing methods have been reported to be more sensitive than other measurements (Dan and Mathew, 1990). Some experimental results showed that the feed and thrust forces were influenced much more by tool wear than the main cutting force. It was also shown that a linear relationship between these forces and tool wear existed, as shown in Figure A.7. Others showed that the main cutting force gave the best indication of

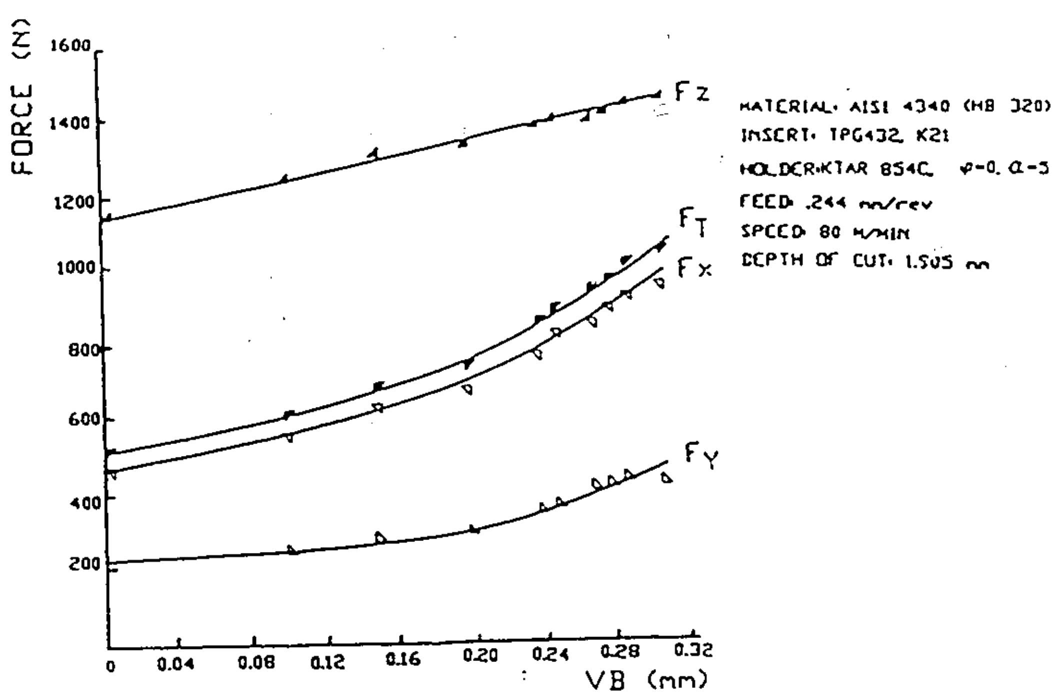
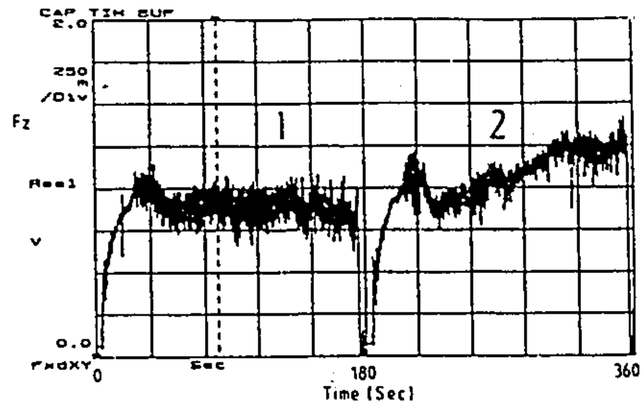


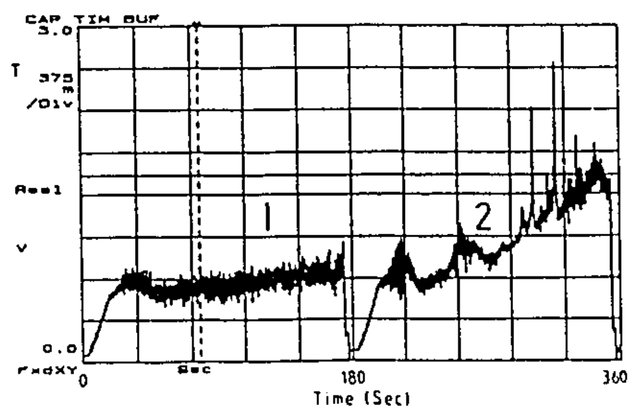
Figure A.7 Cutting Force Components *vs* Flank Wear / Lai, 1986 /

tool wear at any given time. Force rati s, as mentioned previously, were used in lots of cases as the major features in tool condition monitoring. It was also found that the frequency components and amplitude of the cutting force is greatly affected by the tool wear. The powers in different frequency bands are convenient monitoring features.

Cutting forces, represented by the thrust force and the torque, are also important features in monitoring drilling tool conditions. Researchers showed that different combinations of cutting condition brought about different tool wear locations, patterns and magnitudes, and this diversity in drill wear form would almost certainly lead to changes in



(a) Fz in normal and abnormal cutting conditions



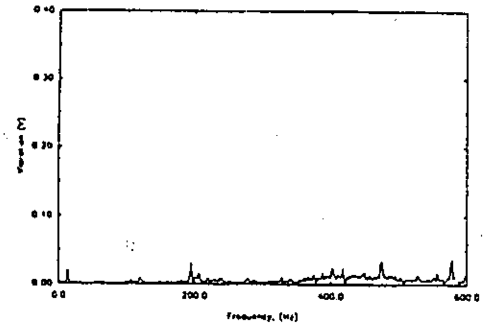
(b) T in normal and abnormal cutting conditions

Figure A.8 Thrust Force and Torque in Normal and Abnormal Cutting / Li *et al*, 1992 /

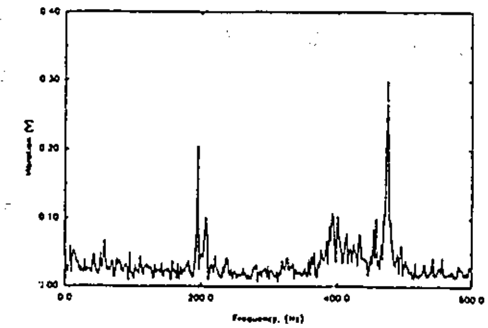
both the thrust force and the torque. Figure A.8 from Li *et al* (1992) shows the thrust force and the torque in normal and abnormal cutting conditions.

A.2.2 Vibrations

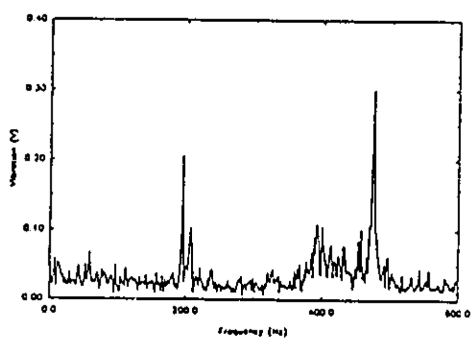
As metal-cutting progresses, the workpiece and chips rub against the tool and produce vibrations which can be easily measured by accelerometers. Vibration information is used



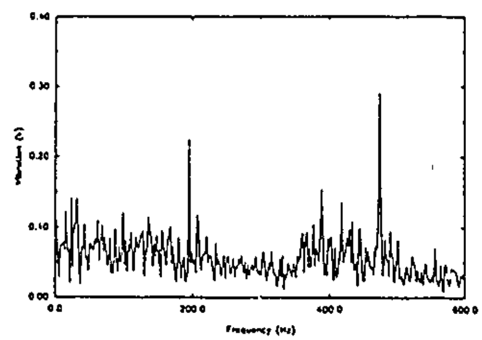
(a) sharp tool



(b) slight wear



(c) medium wear



(d) severe wear

Figure A.9 Spectra of Acceleration Signals under Different Tool Wear States / Du et al, 1992 /

in various ways for tool condition monitoring.

Vibrations are crucial features in the machining process conditions. In detection of chatters, vibration signals give straightforward the information. The mean-crossing rate of the signals is quiet different in stable cutting and in chatter. This is due to the fact that the vibration is essentially a signal with high frequency random noise in stable condition, and is mostly periodic with a dominant frequency which coincides with the weak structural mode.

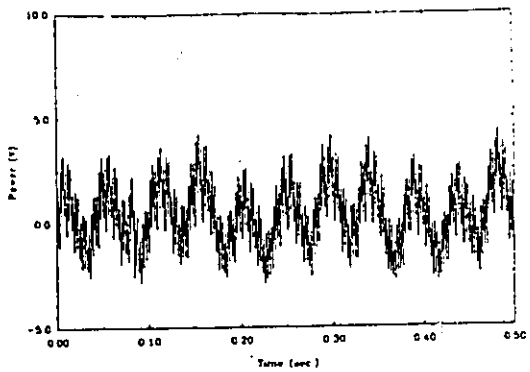
Vibration signals vary with tool failure in some frequency ranges. The mutual relationship between tool wear and power spectrum of vibrations of the tool during cutting has often been studied in the tool condition monitoring. Figure A.9 shows one of the studies on the effects of tool wear on the power spectra of the vibrations. Many other applications of vibration signals for tool condition monitoring in turning and drilling are easily seen in literatures.

A.2.3 Motor Current

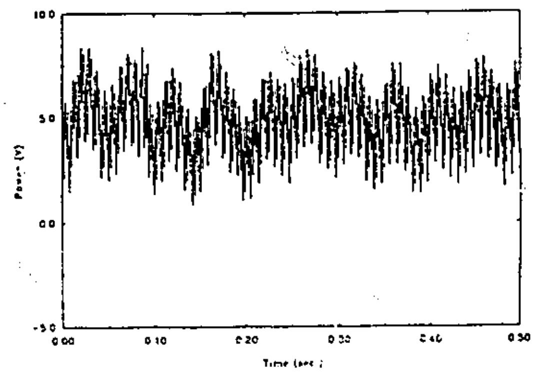
During the cutting process, power/current to the main drive motor is related to the shaft torque and thus the tangential component of the cutting force. Usually, less power is consumed when using a sharp tool compared to a broken or worn tool. This variance can be exploited for tool condition monitoring.

It was reported that a linear relationship existed between motor current and tool flank wear in turning (Liao, 1986). The resulting signal was found to drop instantaneously and soon recover to a level prior to the drop, when tool breakage occurred. It was also found that, under constant spindle speed cutting conditions, the percentage increase of motor current from the beginning till the end of a tool's life was approximately constant if the same material was machined. Figure A.10 from Du *at al* (1992) also approves the feasibility of using motor current signals for tool breakage detection. Researchers have shown that motor current is less sensitive to tool wear when compared to force and vibration signals.

The current measurement is relatively simple and mounting of the system will not affect the machining operation.



(a) sharp tool



(b) broken tool

Figure A.10 Effect of Tool Breakage on Motor Power / Du *et al*, 1992 /

In automated machining process and tool condition monitoring, indirect measurements are always employed. The most used sensor signals include cutting forces, torque, vibrations, motor current, acoustic emission, and so on. Various methods of analyzing the sensor signals have been developed in both time domain and frequency domain. Applications of these sensor signals and their features depend greatly on individual monitoring tasks. Selection of the sensor signals and the monitoring features is a critical issue in the development of automated process and tool condition monitoring in machining.