

*Journal of*

---

# **INDUSTRIAL TECHNOLOGY**

---

*Volume 16, Number 2 - February 2000 to April 2000*

---

## ***Neural Network-Based Tool Breakage Monitoring System for End Milling Operations***

*By Mr. Po-Tsang Huang and Dr. Joseph C. Chen*

### **KEYWORD SEARCH**

**CAM**  
**CIM**  
**Computer Programming**  
**Computer Science**  
**Machine Tools**  
**Manufacturing**  
**Materials & Processes**  
**Production**  
**Quality Control**

*Reviewed Article*

---

*The Official Electronic Publication of the National Association of Industrial Technology • [www.nait.org](http://www.nait.org)*

© 2000

---



Mr. Po-Tsang Huang receives his M.S. degree from the Department of Industrial & Manufacturing System Engineering at Iowa State University. He is currently a Ph.D. student in the Department of Industrial Education and Technology at Iowa State University. His research field includes the industrial applications of using fuzzy logic theory and statistical tools, and the dynamics control systems for detecting the tool condition in end milling operations.



Dr. Joseph C. Chen, PE is an associate professor in the Department of Industrial Education and Technology at Iowa State University. Recently, he received Early Achievement in Teaching Award from the Iowa State University. His research interests are in cellular manufacturing systems, machine tool dynamics, and intelligent control as it relates to neural networks and fuzzy control.

# Neural Network-Based Tool Breakage Monitoring System for End Milling Operations

By Mr. Po-Tsang Huang and Dr. Joseph C. Chen

because it replaces human operators with robotic counterparts in manufacturing and assembly cells. In addition to performing the same function as the FMS, the UFMS reduces direct labor costs and prevents personal oversights.

However, since human operators are absent in these systems, electronic sensors associated with a decision-making system must monitor the process. The decision-making system analyzes information provided by sensors to make appropriate control actions. In order to ensure efficiency within the system, monitoring equipment and algorithms for the adaptation of the manufacturing process must be executed accurately (Altintas, Yellowley, and Tlusty, 1988).

To apply the UFMS effectively, manufacturers must confirm that the tool is in good condition in process; therefore, automatic and rapid detection of tool breakage is central to successful UFMS operation. By themselves, computer numerical control (CNC) machines are not typically capable of tool breakage detection. Since CNC machines cannot detect tool conditions, they cannot halt the process if the tool becomes damaged. Materials costs increase and product quality suffers if a broken tool is used in production. To reduce costs of materials and prevent damaged tools from negatively affecting production, a detecting technology for unexpected tool breakage is needed (Lan and Naerheim, 1986).

In this research, an in-process tool breakage detection system was developed in an end milling operation with cutting force and machining parameters of spindle speed, feed rate, and depth

of cut selected as input factors. The neural networks approach was employed as the decision-making system that judges tool conditions.

## Literature Review

The commonest method of detecting tool breakage in process involves force signals resultant from tool processes on raw materials. A dynamometer sensor is the main device used to measure force signals in different machining operations. Lan and Naerheim (1986) proposed a time-series autoregression (AR) model of force signals to detect tool breakage. The time-series-based tooth period model technique (TPMT), which used the fast a posteriori error sequential technique (FAEST), was applied by Tansel and McLaughlin (1993) to detect tool breakage in milling. Tarnag and Lee (1993) proposed using the average and median force of each tooth in the milling operation. Measured by sensors, the average and median forces of each tooth were used as input information. An appropriate threshold was built to analyze information and detect tool conditions.

Jemielniak (1992) proposed that sudden changes in the average level of force signals could be due to catastrophic tool failure (CTF) in turning operations. In this study, analyzing force signals and determining amplitude fluctuations allowed on-line tool breakage detection. Zhang, Han, and Chen (1995) used a telemetering technique, a battery-powered sensing-force/torque cutter holder mounted on the spindle head with the transmitter, to measure force in milling operations.

All rights reserved. This study or parts thereof may not be reproduced in any form without written permission of the authors. This paper has not been published or submitted for publication in any other journal.

## Introduction

In recent years, global competition in industry has led to exploration of new means of more efficient production. In particular, flexible manufacturing systems (FMS) have been investigated as a tool for raising manufacturing productivity and product quality while decreasing production costs. One type of FMS is the Unmanned Flexible Manufacturing System (UFMS), which has received a great deal of attention

The application of neural networks and fuzzy logic in detecting tool breakage has also been studied in recent years. Tae and Dong (1992) developed a fuzzy pattern recognition technique and a time-series AR model to detect tool wear in turning operations. The variation of dynamic cutting force was used to construct the fuzzy dispersion pattern needed to distinguish tool conditions. Ko, Cho, and Jung (1994) introduced an unsupervised self-organized neural network combined with an adaptive time-series AR modeling algorithm to monitor tool breakage in milling operations. The machining parameters and average peak force were used to build the AR model and neural network.

Chen and Black (1997) also introduced a fuzzy-nets system to distinguish tool conditions in milling operations. The fuzzy-nets system was designed to build the rule-bank and solve conflicting rules with a computer. Variance of adjacent peak force was selected as an input parameter to train the system and build a rule-bank for detecting tool breakage.

Two detrimental research methods were discovered in the literature review, summarized as follows:

1. Some researchers used only one broken or one good tool to conduct experiments, a practice that does not provide enough data to explain whether a method is applicable to other tool breakage situations.
2. Some research used soft materials, such as aluminum T6061, which often did not strain tools enough for breakage to occur.

Therefore, in this research, to verify that the models detected tool conditions successfully, multiple cutting tools using the same pattern were selected to execute the experiment. Use of multiple tools provided an advantage by supplying sufficient data to explain whether the models apply to different tool breakage situations. A relatively hard material, steel AISI-1018, was selected as a workpiece to confirm this experiment's

applicability in real world processes that use hard materials.

### Methodology

#### Force signals in the end milling cutting process

Milling is a fundamental machining process in the operation of CNC machines, and milling operations can be of two varieties: peripheral and face milling. Of the two, face milling is commonest in manufacturing systems, and was the type studied in this research. Milling is an interrupted cutting process, meaning that each cutting tooth moving in the same direction generates a cyclic cutting force ranging from zero to maximum force, and back to zero. This cyclic force is graphed as a series of peaks. The principle of cutting force can be further defined as resultant force,  $F_R$ ,

generated in X and Y directions. The resultant force,  $F_{R_i}$ , generated from X and Y directions, was used in this experiment expressed as:

$$(1) F_{R_i} = \sqrt{F_{x_i}^2 + F_{y_i}^2}$$

Where

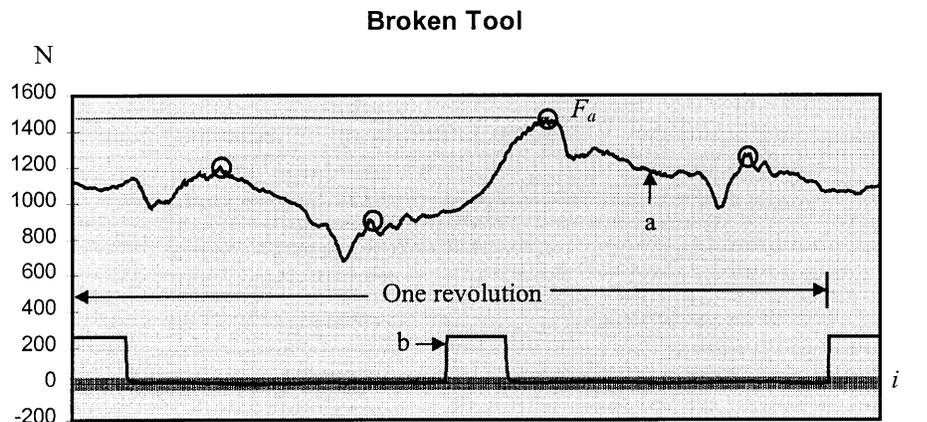
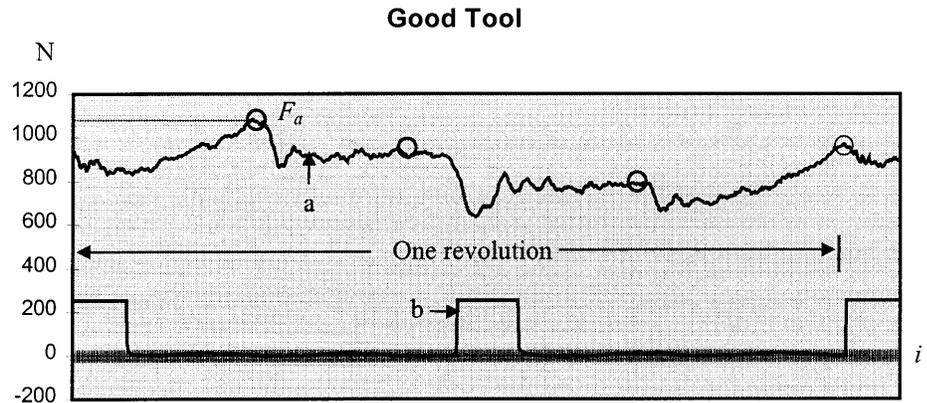
$F_{R_i}$ : the resultant force of point  $i$

$F_{x_i}$ : the force in X direction of point  $i$

$F_{y_i}$ : the force in Y direction of point  $i$

Tool conditions and machining parameters affect the magnitude of resultant force; therefore, if the tool condition is good, the peak measurement of each tooth's force should be roughly the same from tooth to tooth during one revolution of the cutting

Figure 1. The amplitude of cutting force of a good and broken tool



Key:  
 a = force signal  
 b = revolution signal

Cutting Parameters:  
 speed = 650 rpm, feed = 15 ipm  
 depth = 0.08 inch

process. Comparatively, if a tooth is broken, it generates a smaller peak force because it carries a smaller chip load. As a result, the tooth that follows a broken tooth generates a higher peak force as it extracts the chip that the broken tool could not.

Applying the force principle, two main differences can be used to detect tool breakage: (1) maximum peak force in each revolution should differ between good and broken tools, and maximum peak force of a broken tool must be larger than that of a good tool; and (2) maximum variance force of adjacent peaks should differ between good and broken tools, and maximum variance force of adjacent peaks of broken tools must be larger than in undamaged tools. Fig. 1 (page 3) illustrates the diagram of undamaged and broken tools.

In research, an in-process tool breakage detection system was developed in an end milling operation. The cutting force and machining parameters, such as spindle speed, feed rate, and depth of cut, were selected as input factors. Finally, the neural networks approach was used as a decision-making system using input from sensors to judge tool conditions.

**Neural Networks**

Neural networks are parallel, distributed information-processing structures designed to learn from examples and mistakes. They have been successfully applied to many exercises in engineering, business, and industry because they possess many advantages, including a mathematical basis, the ability to learn by example, fault tolerance, generalization capabilities, and abstraction (Jang, Sun, and Mizutani, 1997). In this research, a back propagation net (BPN) was chosen as the decision-making system because it is the most representative and commonly used algorithm. It is relatively easy to apply, has been proven effective in dealing with this kind of task, and has also proven successful in practical applications.

Back propagation is intended for training layered (i.e. nodes are grouped in layers), feed forward (i.e. the arcs joining nodes are unidirectional, and

there are no cycles) nets, as shown in Fig. 1. This approach involves supervised learning, which requires a ‘teacher’ that knows the correct output for any input, and uses gradient descent on the error provided by the teacher to train the weights. As the weights of the neural network were obtained, the prediction function was achieved via the weight information. The propagation rule, also called a summation or aggregated function, was used to combine or aggregate inputs passing through the connections from other neurons. It can be expressed as

$$S_i = \sum a_i W_{ji} - a_0 W_{j0} \quad (2)$$

$$S_k = \sum a_i W_{ji} - a_0 W_{j0} \quad ,(3)$$

where, i is an input neuron, j is a hidden neuron, and k is an output neuron,  $W_{ji}$  and  $W_{kj}$  denote weight from input to hidden neuron, and from hidden to output neuron respectively, while  $a_0$  represents the bias, usually 1, and  $W_{ji}$  and  $W_{j0}$  are the weight of bias.

The transfer function, also called the ‘output’ or ‘squashing’ function, is used to produce output based on level of activation. Many different transfer functions can be used to transfer data, and one is called the Sigmoid Function, expressed as:

$$O_y = \frac{1}{(1 + e^{-ay})} \quad ,(4)$$

where ay is a function of  $S_j$  and  $S_k$  respectively.

Comparing actual output of neural networks to desired output, the process is repeated until the error percentage falls into a reasonable range.

**Experimental Design And System Development**

**Experimental design**

This experiment employed a Fadal CNC vertical machining center. A Kistler 9257B 3-component dynamometer mounted on the table measured cutting force, and a steel (BHN200) workpiece was mounted on the dynamometer. A proximity sensor was built

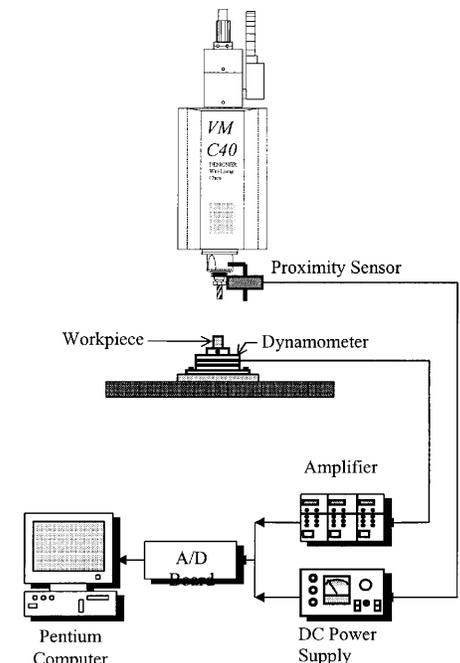
near the spindle to confirm data in each revolution. Four 3/4-inch double-end four-flute high-speed steel cutters were used. In each cutter, one side of the tool was in proper working order and the other side was broken. The broken side of the tool possessed varying degrees of breakage (Fig. 2). The experimental setup was shown in Fig. 3. The cutting parameters were set as: five level of spindle speed (740, 500, 550, 600, and 650 revolution per minute), five level of feed rate (6, 12, 18 and 24 inch per minute), and five levels of depth of cut (0.06, 0.07, 0.08, 0.09 and 0.1 inches.

The cutters used to execute the experiment were selected randomly. Cutting force was measured in voltage by the Charge Amplifier and transformed to Newtons (N) via computer.

**Neural Network-BP system development**

To develop back propagation of neural networks as a decision-making system, PCN (PC Neuron) software was applied to analyze data. Seven steps were conducted. In step one, prediction factors were determined in order to perform the training process. Step two was necessary to analyze

Figure 2. The experimental setup



differences between scaling data and unscaling data. Step 3 dealt with separating data into training and testing categories. From steps 4 through 6, parameters were developed for the training process, including the hidden layer/hidden neuron, learning rate, and momentum factor,. Finally, in step 7, information from the training process was used to predict tool conditions.

*Step 1. Determine the factors*

Five input neurons were used for tool breakage prediction data: (1) spindle speed; (2) feed rate; (3) depth of cut; (4) maximum peak force; and (5) maximum variance of adjacent peak force. Output neurons were either (1) Good, or (2) Broken.

Three hundred data points were used in this research. Good tools collected half of these and broken tools collected the rest, and all data were randomized using MS Excel software.

*Step 2. Analyze unscaling and scaling data*

In order to avoid experimental errors resulting from bigger values of some data sets, some pre-processing was needed to obtain good training and prediction results. Since histograms of all data sets were uniform or normal distributions, the Simple Linear Mapping method was employed for scaling. To compare the difference between two sample sizes, some parameters were first set and fixed. The number of hidden neurons was set at 4, the learning rate was set at 1, and the momentum item was 0.5. The number of training cycles was 2000, and the testing period was 5. Table 1 shows the comparison of the difference between scaling and unscaling data. As one can see, errors in scaling data are smaller than in unscaling data

*Step 3. Impact of the ratio of training and testing data*

The 300 original 300 data records were randomized and separated into three groups. The first group had 200 training data and 100 (200\*100) testing data, the second had 225 training and 75 (225\*75) testing data, and the third had 250 training and 50 testing (250\*50)

data. Table 2 shows the Back Propagation Net (BPN) with different sample sizes of training and testing data. The last four columns of Table 2 show training and testing errors. The training, testing, and RMS (root mean square) errors of training of the second group were smaller than in other groups. The RMS errors of testing data of the second group sample were larger than in the first; however, the RMS errors of each sample size were very similar. If samples had similar error percentages, the sample with the largest training sample size was selected because it provided sufficient information to predict testing data. From the experimental design, the ideal ratio between training and testing data was 3:1 for neural networks. The 225\*75 sample size was employed in this analysis.

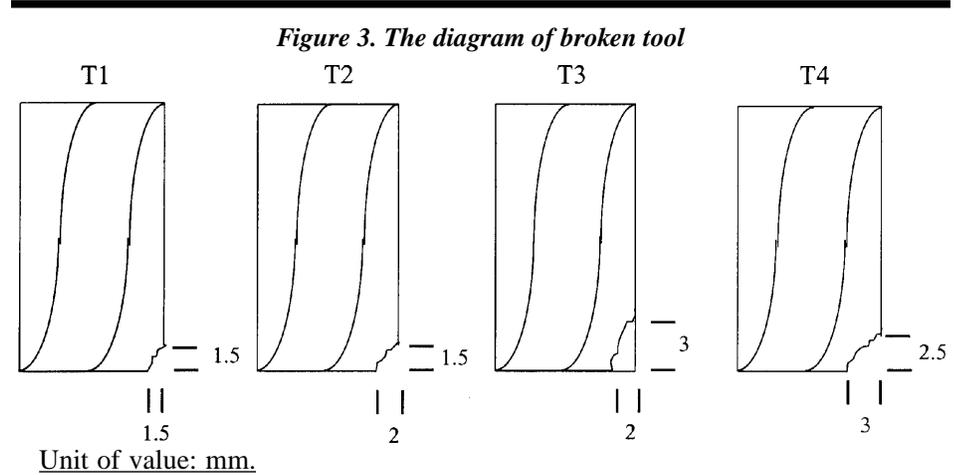
*Step 4. Impact of the hidden layer and hidden neuron*

In the beginning, the number of hidden neurons was set at 5, and the

hidden layer was set at 1. Different hidden neurons and layers were tested to determine which values would lead to the smallest error percentage. To this end, the hidden neurons were set at 4 and 6, and the hidden layers were set at 1 and 2. Table 3 shows the BPN with a different number of hidden neurons and layers. According to this data, the percentage error of the trial with 4 hidden neurons and 1 hidden layer was less than it was in all other trials. Thus, the configuration contained in the 4 hidden neuron/1 hidden layer experiment was chosen because it led to the best results. The formula, (input neurons + output neurons)/2, was useful for determining the number of hidden neurons at the beginning.

*Step 5. Impact of the learning rate*

This step was necessary to determine the optimal learning rate. The initial learning rate was 1. Three additional learning rates, 0.5, 2, and 10, were used to compare with the



**Table 1. Difference between Scaling and Non-scaling**

	Hidden layer	Hidden neuron	Learn rate	Momentum factor	Train Error	Testing Error	RMS error of training	RMS error of testing
Unscale	1	4	1	0.5	0.505	0.550	0.508	0.515
<b>Scale</b>	1	4	1	0.5	<b>0.040</b>	<b>0.160</b>	<b>0.197</b>	<b>0.388</b>

**Table 2. Different Sample Size of Training and Testing Data**

Tra*Tes	Hidden layer	Hidden neuron	Learn rate	Momentum factor	Train Error	Testing Error	RMS error of training	RMS error of testing
200*100	1	4	1	0.5	0.040	0.160	0.197	0.388
<b>225*75</b>	1	4	1	0.5	<b>0.036</b>	<b>0.093</b>	<b>0.185</b>	<b>0.298</b>
250*50	1	4	1	0.5	0.044	0.106	0.204	0.285

(Note: The Data set is scaling data)

initial. Table 4 shows the BPN with different learning rate values. Table 4 shows that the error percentage of the learning rates of 0.5 and 1 were the same, in addition to being lower than all other learning rates. To achieve the objective of finding the smallest error percentage, the learning rate of 1 was used, because the software originally recommended that value.

*Step 6. Impact of the Momentum factor*

The final step of data analysis was to change the value of the momentum item to obtain the configuration leading to the lowest error percentage. The initial value of the momentum item was 0.5. Another three values, 0.3, 0.6, and 0.8, were selected to compare with the initial value. Table 5 shows the BPN with different values for the momentum item. Table 5 shows that the percentage of errors of momentum items of 0.3 and 0.5 are the same, and smaller than all others. To achieve the smallest error percentage, the 0.5 momentum item was used, because the software originally recommended that value.

*Step 7. Prediction*

After completing analysis and obtaining information about weight and input factors, equations to predict tool conditions were constructed. The variables  $a_1, a_2, \dots, a_5$  represent 5 input factors, maximum peak force, spindle speed, feed rate, depth of cut, and maximum variance of adjacent peak force, respectively. By application of equation (4), the weighted value of hidden factors  $a_{h1}, a_{h2}, a_{h3}, a_{h4}$  can be expressed as:

$$a_{h1} = \frac{1}{(1 + \exp^{-[a1*(-1.652)+a2*(0.448)+a3*(0.947)+a4*(-25.237)+a5*(0.853)-(-0.221)])} \quad (5)$$

$$a_{h2} = \frac{1}{(1 + \exp^{-[a1*(40.457)+a2*(39.421)+a3*(-15.261)+a4*(7.317)+a5*(-21.054)-(-44.505)])} \quad (6)$$

$$a_{h3} = \frac{1}{(1 + \exp^{-[a1*(-10.224)+a2*(-3.444)+a3*(24.252)+a4*(3.449)+a5*(4.215)-(-1.2891)])} \quad (7)$$

$$a_{h4} = \frac{1}{(1 + \exp^{-[a1*(1.321)+a2*(-24.736)+a3*(0.202)+a4*(0.79)+a5*(0.015)-(-0.829)])} \quad (8)$$

Then, the same methodology was used to calculate the output value,  $a_{out1}$  and  $a_{out2}$ , expressed as:

$$a_{out1} = \frac{1}{(1 + \exp^{-[ah1*(11.697)+ah2*(16.977)+ah3*(12.295)+ah4*(11.807)-(-2.945)])} \quad (9)$$

$$a_{out2} = \frac{1}{(1 + \exp^{-[ah1*(-11.697)+ah2*(-16.977)+ah3*(-12.295)+ah4*(-11.807)-(-2.945)])} \quad (10)$$

Finally, the output information was used to judge the tool conditions:

if  $a_{out1} > a_{out2}$ , then the tool condition is good

if  $a_{out1} < a_{out2}$ , then the tool is broken.

**Findings And Conclusions**

To operate the UFMS successfully, in-process sensing techniques that relate to rapid-response decision-making systems were required. In this research, a neural networks model was developed to judge cutting force for accurate in-process tool breakage detection in milling operations. The neural networks were capable of detecting tool conditions accurately and in process. The accuracy of training data was 96.4%, and the accuracy of testing data was 90.7%. Partial results of training and testing data are shown in Tables 6 and Table 7.

**Table 3. Different Number of Hidden Neurons and Layers**

Neuron in layer-1	Neuron in layer-2	Learn rate	Momentum factor	Train Error	Testing Error	RMS error of training	RMS error of testing
3	~	1	0.5	0.080	0.200	0.256	0.410
4	~	1	0.5	<b>0.036</b>	<b>0.093</b>	<b>0.185</b>	<b>0.298</b>
5	~	1	0.5	0.049	0.093	0.193	0.288
3	3	1	0.5	0.267	0.320	0.338	0.375
4	2	1	0.5	0.489	0.453	0.512	0.504
4	3	1	0.5	0.316	0.333	0.407	0.425
4	4	1	0.5	0.164	0.227	0.362	0.414

(Note: The Data set is 225\*75)

**Table 4. Different Value of Learning Rate**

Hidden layer	Hidden neuron	Learn rate	Momentum factor	Train Error	Testing Error	RMS error of training	RMS error of testing
1	4	<b>0.5</b>	0.5	<b>0.036</b>	<b>0.093</b>	<b>0.185</b>	<b>0.298</b>
1	4	1	0.5	<b>0.036</b>	<b>0.093</b>	<b>0.185</b>	<b>0.298</b>
1	4	2	0.5	0.116	0.133	0.306	0.319
1	4	10	0.5	0.111	0.133	0.286	0.317

(Note: The Data set is 225\*75)

**Table 5. Different Value of Momentum Item**

Hidden layer	Hidden neuron	Learn rate	Momentum factor	Train Error	Testing Error	RMS error of training	RMS error of testing
1	4	1	<b>0.3</b>	<b>0.036</b>	<b>0.093</b>	<b>0.185</b>	<b>0.298</b>
1	4	1	<b>0.5</b>	<b>0.036</b>	<b>0.093</b>	<b>0.185</b>	<b>0.298</b>
1	4	1	0.6	0.049	0.093	0.212	0.288
1	4	1	0.8	0.116	0.133	0.306	0.322

The weights of hidden factors and output factors were generated from pre-trained neural networks, and a program was written to process these weights in order to respond to the tool conditions. Therefore, the in-process detection system demonstrated a very short response-time to tool conditions. Since tool conditions could be monitored in a real-time situation, the broken tool could be replaced immediately to prevent damage to the machine and mis-machining of the product. However, since the weights were obtained from the pre-trained process, they were fixed when they were put into the detection program. Therefore, the whole system does not have the adaptive ability to feed back information into the system.

In this research, depth of cut was employed as one input factor. However, in actual industrial environments, the surface of work materials is often uneven, implying that the depth of cut set in the computer might differ from that used to cut the workpiece. Under the circumstances, the neural networks might generate a wrong decision and misjudge the tool conditions due to fluctuating depths of cut across machining. Therefore, future research should consider avoiding depth of cut as a determinate of whether obtaining a low error percentage is still possible. Furthermore, different decision-making tools, such as fuzzy logic or genetic algorithms, should also be applied to see which one could obtain a smaller error of detection.

### References

Altintas, Y., Yellowley, I. and Tlustý, J. (1988). The Detection of Tool Breakage in Milling Operations. *Engineering for Industry*, 110, 271-277.

Chen, J. C. (1996). A Fuzzy-Nets Tool-Breakage Detection System for End-Milling Operations. *Int. J. advanced manufacturing Technology*, Vol. 12, pp. 153-164.

Chen, J. C. & Black, J. T. (1997). A Fuzzy-Nets In-process (FNIP) System for Tool-Breakage Monitoring in End-Milling Operations. *Int. J. Mach. Tools Manufact.*, Vol. 37, No. 6, pp. 783-800.

Jang, J. S., Sun, C. T. & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing*, New Jersey: Prentice Hall.

Jemielniak, K. (1992). Detection of Cutting Edge Breakage in Turning. *Annals of the CIRP*, Vol. 41, No. 1, pp. 97-100.

Lan, M. and Naerheim, Y. (1986). In-Process Detection of Tool Breakage in Milling. *Engineering for Industry*, 108, 191-197.

Ko, T. J. & Cho, D. W. & M. Y. Jung (1994). On-line Monitoring of Tool Breakage in Face Milling: Using a Self-Organized Neural Network. *Journal of Manufacturing systems*, Vol. 14, No. 2, pp. 80-90.

Tansel, I. N. & McLanghlin, C. (1993). Detection of Tool Breakage in Milling Operation-I. The Time Series Analysis Approach. *Int. J. Mach. Tools Manufact.*, Vol.33, No. 4, pp. 531-544.

Tansel, I. N. & McLanghlin, C. (1993). Detection of Tool Breakage in Milling Operation-II. The Neural Network Approach. *Int. J. Mach. Tools Manufact.*, Vol.33, No. 4, pp. 545-588.

Tang, Y. S. & Lee, B. Y. (1993). A Sensor for the Detection of Tool Breakage in NC Milling. *Journal of Materials Processing Technology*, Vol. 36, pp.259-272.

Zhang, D., Han, Y. & Chen, D. (1995). On-line Detection of Tool Breakage Using Telemetry of Cutting Forces in Milling. *Int. J. Mach. Tools Manufact.*, Vol. 35, No. 1, pp. 19-27.

Table 6. Partial Results of Training Data

Tool Conditions	Input factors					Output factors		Prediction
	a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>	a <sub>4</sub>	a <sub>5</sub>	a <sub>out1</sub>	a <sub>out2</sub>	
Good	904.5	600	12	0.09	45.8	1	0	Good
Broken	1220.54	600	12	0.09	954.04	0	1	Broken
Good	634.64	550	10	0.06	274.64	0.98	0.02	Good
Broken	780.14	550	10	0.06	368.54	0.03	0.97	Broken
Good	674.36	650	15	0.06	248.64	1	0	Good
Broken	847.06	650	15	0.06	537.54	0.2	0.98	Broken
Good	1239.4	500	18	0.07	225.2	1	0	Good
Broken	1677.92	500	18	0.07	1,159.64	0	1	Broken
<b>Good</b>	<b>1413.76</b>	<b>450</b>	<b>15</b>	<b>0.08</b>	<b>300.56</b>	<b>0</b>	<b>1</b>	<b>Broken</b>
Broken	1861.72	450	15	0.08	1,250.92	0.01	0.99	Broken

Table 7. Partial Results of Testing Data

Tool Conditions	Input factors					Output factors		Prediction
	a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>	a <sub>4</sub>	a <sub>5</sub>	a <sub>out1</sub>	a <sub>out2</sub>	
Good	711.94	550	15	0.06	177.02	1	0	Good
Broken	1296.56	550	15	0.06	481.66	0.01	0.99	Broken
<b>Good</b>	<b>723.32</b>	<b>550</b>	<b>12</b>	<b>0.07</b>	<b>311.52</b>	<b>0.35</b>	<b>0.65</b>	<b>Broken</b>
Broken	1215.96	550	12	0.07	1,042.50	0	1	Broken
Good	1084.32	550	18	0.07	192.5	0.99	0.01	Good
Broken	1542.92	550	18	0.07	1,303.92	0	1	Broken
Good	1024.46	600	18	0.07	172.22	0.98	0.02	Good
Broken	1253.28	600	18	0.07	580.3	0.03	0.97	Broken
Good	1507.18	450	20	0.08	550.06	0.98	0.02	Broken
Broken	1876.74	450	20	0.08	1,062.02	0	1	Broken