Fuzzy Neural Network Modelling for Tool Wear Estimation in Dry Milling Operation

X. Li[1*], B.S. Lim[1], J.H. Zhou[1], S. Huang[1], S.J. Phua[1], K.C. Shaw[1], and M.J. Er[2]

[1] *Singapore Institute of Manufacturing Technology, 71 Nanyang Drive, Singapore 638075*
[2] *School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798*
* *Correspondence author, xli@simtech.a-star.edu.sg*

ABSTRACT

Tool failure may result in losses in surface finish and dimensional accuracy of a finished part, or possible damage to the work piece and machine. This paper presents a Fuzzy Neural Network (FNN) which is designed and developed for machinery prognostic monitoring. The FNN is basically a multi-layered fuzzy-rule-based neural network which integrates a fuzzy logic inference into a neural network structure. The fuzzy rules help to speed up the learning process of the complex conventional neural network structure and improve the accuracy in prediction and rate of convergence. A case study for prediction of tool life in a dry milling operation is presented to demonstrate the viability of the proposed FNN for tool condition monitoring. A comparison was made in the case study on prediction performances of different models established with the same set of experimental data. It is shown that the FNN is superior to conventional Multi-Regression Models (MRM), Backpropagation Neural Networks (BPNN) and Radial Basis Function Networks (RBFN) in term of prediction accuracy and BPNN in learning speed.*

Key Words: Fuzzy neural networks, tool condition monitoring, tool wear estimation, milling machining.

1. INTRODUCTION

In metal cutting processes, tool condition monitoring (TCM) can play an important role in maintaining the quality of surface finishing. Monitoring of tool wear to prevent surface damage is one of the difficult tasks in the context of TCM (Sick, 2002). Currently, a general common approach is to measure several process parameters that are indirectly correlated to the tool performance, such as cutting force, tool vibration and acoustics emissions, transform the measured data into useful reference models for condition and performance monitoring. Numerous condition monitoring methodologies have been proposed and evaluated during the past two decades. Among them, Yamaguchi [Yamaguchi *et al.*, 2007] investigated the cutting force and acoustic emission (AE) signals to gauge tool life of diamond cutting tool; Vallejo Jr (Vallejo Jr *et al.*, 2005) presented online monitoring of the cutting tool condition based on Hidden Markov Models.

Fuzzy-logic, neural-network and their combinations like Fuzzy-Nets (FN) are widely used in modeling and prediction in precision engineering. Haber (Haber *et al.*, 2003) applied intelligent process supervision for predicting tool wear in machining processes; and Li et al (Li *et al.*, 1996) applied the adaptive neuro-fuzzy inference system (ANFIS) (Er *et al*., 2003) and wavelet transforms to tool condition monitoring. Similar cases are also discussed by (Sick, 2002). A fuzzy-neuro adaptive surface roughness control is proposed for the prediction of the surface roughness and adaptive feed-rate control (Yang *et al.*, 2006). A hybrid Taguchi-genetic learning algorithm is used to set up a nonlinear model to correlate the surface roughness values with distinct spindle-speed, feed-rate and depth-of-cut (Ho *et al.*, 2009). It is shown that Gaussian membership functions are suitable choices for fuzzy layer of the network for predicting the surface-roughness in (Ho *et al.*, 2009; Aliustaoglu *et al.*, 2009; Lo *et al.*, 2003).

Although fuzzy neural networks are widely used in modeling and prediction in milling machining processes, even the most promising methods are not easily adoptable in real industrial operations (Huang *et al.*, 2000; Wang *et al.*, 2001; Dong *et al.*, 2004), particularly due to insufficient generalisation capabilities (e.g. the use is restricted to a specific machine tool, only a small range of cutting conditions is allowed, or time-consuming 'teach-in' cycles are needed) or lack of precision.. As such, very limited effort has been reported on the development of a generic toolkit that provides reference models for on-line tool condition monitoring and remain useful life prediction. It is therefore desirable to develop an intelligent predictive monitoring system (IPMS) (Zhou *et al.*, 2005) (Li *et al.*, 2006) with capabilities in feature extraction, feature

selection, correlation modelling, and data clustering for tool condition monitoring, non-destructive characterization and tool life span prediction. This paper focuses on an approach using fuzzy neural network (FNN) to establish tool wear reference models for prediction of tool life span in dry milling operation. Experimental study is carried out to establish reference models for predicting wear of a 6mm ball-nose two flutes tungsten carbide milling cutters in a milling operation, using the proposed FNN technique. The developed FNN models are benchmarked with the Multi-Regression Models (MRM), Backpropagation Neural Networks (BPNN) and Radial Basis Functional Networks (RBFN) using the same set of experimental data.

2. THE FNN ARCHITECTURE

Figure 1 shows the architecture of the FNN, which is basically a five-layer fuzzy-rule-based neural network. In accordance with the common neural network notation, a node in any layer n of the network has its input termed *net-in*^{n} or simply *net*^{n}. The node performs a certain operation on the input and generates an output which is a function of its input, i.e. output $f(net\text{-}in^n)$ or $f(net^n)$.

Layer 1: The nodes in this layer transmit input values u_k to next layer directly as u_{kq}, where $u_k = u_{kq}$. That is,

$$net_k^1 = u_k \;;\; f(net_k^1) = net_k^1 = u_{kq} \qquad (1)$$

where $k = 1, 2, \ldots, p$ and $q = 1, 2, \ldots, n$

Layer 2: The nodes in layer 2 are the input membership functions. They work as a fuzzifier transforming a numerical input into a fuzzy set. The membership functions are normal distributions with a range between 0 and 1 (inclusive 1), governed by

$$net_q^2 = -\frac{(u_{kq} - m_{kq})^2}{(\sigma_{kq})^2} \quad \text{and}$$

$$f(net_q^2) = e^{net_q^2} = u_{qi} \qquad (2)$$

where $i = 1, 2, \ldots, h$

where m_{kq} and σ_{kq} are the mean and variance of the input membership function, respectively.

Layer 3: The nodes perform a fuzzy min-max operation on the node inputs, i.e. a fuzzy AND operation followed by a fuzzy OR operation. We have

$$net_i^3 = min\{u_{qi} \cdot w_{qi}\} \quad \text{and}$$

$$f(net_i^3) = net_i^3 = u_{ij}, \qquad \text{where } i = 1, 2, \ldots, h$$

$$u_{cj} = max(net_1^3, net_2^3, \ldots, net_h^3) \qquad (3)$$

where $j = 1, 2, \ldots, m$ and $c \in \{1, 2, \ldots, h\}$

Here the link weight w_{qi} is unity. The node c is termed the winner node of the fuzzy min-max operation.

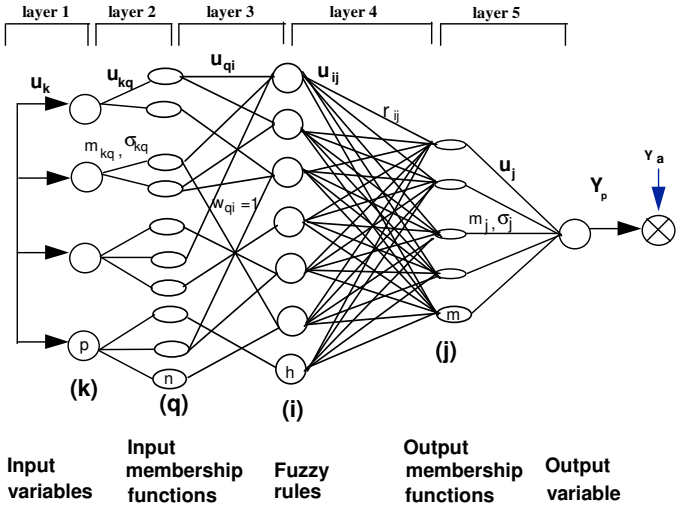

Fig. 1: The architecture of FNN

Layer 4: The links in this layer represent the dampened outputs of the winner node. We have

$$net_j^4 = u_{cj} \cdot r_{cj} \quad \text{and} \quad f(net_j^4) = net_j^4 = u_j \qquad (4)$$

where $i = c$ and $j = 1, 2, \ldots, m$

The dampening coefficients are the rule values r_{ij}s. The initial rule values are either random values or assigned directly by an expert. They can also be established outside the network from historical data and then incorporated into the network. The rule values are subsequently fine-tuned during learning.

Layer 5: This layer performs defuzzification of outputs. The defuzzification method used is the centre of gravity method (Kosko 1992), which uses the centroid of the membership function as the representative value. Thus if m_j and σ_j are the mean and the variance of the output membership function respectively, then the defuzzified outputs are given by eq. 5 as follows:

$$f(net_{output}^5) = \frac{net_{output}^5}{\sum_{j=1}^{m} \sigma_j u_j} = \hat{y}(t) \qquad (5)$$

where $net_{output}^5 = \sum_{j=1}^{m} \sigma_j m_j u_j$ and $m_j \bullet \sigma_j$ is the link weight, $j = 1,2, \ldots, m$.

3. LEARNING ALGORITHMS

3.1. Self Learning Algorithm

The Kohonen's Self-Organized Maps (SOM) algorithm (Kohonen, 1988) is used here to define the number of membership functions and their respective means and variances. For a given set of data X = (x_1, x_2, …, x_n), initial mean values m_1, m_2, …, m_k are assigned arbitrarily, where

$$min\ (x_1, x_2, \ldots, x_n) < m_i < max\ (x_1, x_2, \ldots, x_n)$$

The data are then grouped around the initial means according to:

$$| x_j - m_c | = \min_i \{ | x_j - m_i | \} \quad 1 \le i \le k \quad \text{and} \quad 1 \le j \le n \qquad [6]$$

where m_c is the mean to which the datum x_j belongs. The data grouping and the mean values are optimised by the following iterative process:

Let $x_j(t)$ be an input and $m_c(t)$ the value of m_c at iteration t ($t = 0,1,2,\ldots$), then

$$m_c(t+1) = m_c(t) + \alpha(t)[\, x_j(t) - m_c(t)] \qquad [7]$$

if x_j belongs to the group of m_c, and

$$m_c(t+1) = m_c(t) \qquad [8]$$

if x_j does not belong to the group of m_c

$\alpha(t)$ $[0<\alpha(t)<1]$ is a monotonically decreasing scalar learning rate. The iterations stop at either after a certain number of cycles decided by the user or when the condition $|m_c(t+1)-m_c(t)| \le \delta$ is satisfied, where δ is an error limit assigned by the user. The variances of membership functions can be determined by eq. (9) below:

$$\sigma_i = \frac{1}{R}\sqrt{\frac{1}{p_i}\sum_{j=1}^{p_i}(x_j - m_i)^2} \quad 1 \le i \le k \qquad [9]$$

where σ_i is *ith* variance of membership function, m_i *ith* mean of membership function, x_j *jth* observed data sample, k total number of membership function nodes, p_i total number of data samples in *ith* membership function group, and R overlap parameter.

For a given input or output variable, the number of initial mean values (m_1, m_2, …, m_k) is assigned by trials and errors. This involves striking a balance between learning time and accuracy. Too small a number results in an oversimplified structure and might therefore adversely affect accuracy. On the other hand, too large a number increases network complexity unnecessarily, resulting in a considerable increase in learning time with very little or no increase in accuracy.

3.2. Supervised Learning Algorithm

A supervised learning algorithm has been developed. The objective of the supervised learning is to minimize the error function E as defined in eq. 10 below by means of a learning algorithm.

$$E = \frac{1}{2}[y(t) - \hat{y}(t)]^2 \qquad (10)$$

where $y(t)$ is the actual output, and $\hat{y}(t)$ the predicted output.

In FNN, the learning algorithm used is derived from the back-propagation algorithm of (Rumelhart *et al.*, 1986). Thus if η is the assigned learning rate, the rule value r_{ij} are fine-tuned as follows:

$$r_{ij}(t+1) = r_{ij}(t) - \eta\frac{\partial E}{\partial r_{ij}} \qquad (11)$$

$$\frac{\partial E}{\partial r_{ij}} = \frac{\partial E}{\partial f(net_j^4)} \cdot \frac{\partial f(net_j^4)}{\partial (net_j^4)} \cdot \frac{\partial (net_j^4)}{\partial r_{ij}} \qquad (12)$$

From *eqs.* (4) and (5), we have

$$r_{ij}(t+1) = r_{ij}(t) - \eta\frac{\partial E}{\partial r_{ij}} = r_{ij}(t) + \eta u_{ij}[Y_a(t) - Y_p(t)]\frac{(\sum_{j=1}^{m}\sigma_j u_j)\sigma_j m_j - (\sum_{j=1}^{m}\sigma_j m_j u_j)\sigma_j}{(\sum_{j=1}^{m}\sigma_j u_j)^2} \qquad (13)$$

where $u_{ij} = min[e^{-\frac{(u_k - m_{kq})^2}{(_{kq})^2}}]$, the *net-in* to the node of layer 4, u_k is value of input in the 1st layer, k = 1,2,...,K, m_{kq} is mean of input membership function in the 2nd layer, σ_{kq} is variance of input membership function in the 2nd layer, η is learning rate, r_{ij} is rule value or damping coefficient, $Y_a(t)$ is actual output, $Y_p(t)$ is predicted output, m_j is mean of output membership function in the 5th layer, σ_j is variance of output membership function in the 5th layer and u_j is net-input to the node at the 5th layer.

The learning process is iterated until an acceptable minimum error between actual output Y_a and predicted output Y_p is achieved.

4. EXPERIMENTAL

4.1. Experiment Set-up and Data Acquisition

A high speed CNC machine (Röders Tech RFM760) with spindle speed up to 42,000 rpm was selected for the experiment. The workpiece material used in the machining test was stainless steel (HRC52). The workpieces were cut

off from original stock and their surfaces were prepared through face milling to get rid of the original skin layer containing hard particles. The surface was then machined to have a slope with 60˚ to accommodate the 2-flute ball nose cutter. A Kistler quartz 3-component platform dynamometer was mounted between the workpiece and machining table to measure the cutting forces in the form of charges, and converted to voltages by the Kistler charge amplifier. Three Kistler piezo accelerometers were mounted on the workpiece to measure the machine tool vibrations of cutting process in X, Y, Z direction respectively. A Kistler acoustic emission (AE) sensor was mounted on the workpiece to monitor the high frequency stress wave generated by the cutting process.

The outputs of these sensors were conditioned through corresponding signal conditioning accessories such as charge amplifiers or couplers. The voltage signals were captured by a NI DAQ PCI 1200 board with 12KHz frequency. The DAQ board generates 16-bit digitized data and directly streamed to a hard disk of an Intel Core 2 Quad 2.66GHz based industrial PC with 8GB RAM.

Eight channels of signals (*force_x*, *force_y*, *force_z*, *acce_x*, *acce_y*, *acce_z*, AE_RMS, AE) were captured by the DAQ card with an accumulated sampling rate of $12kHz \times 8 = 96kHz$. Fig. 2 illustrates the experimental setup.

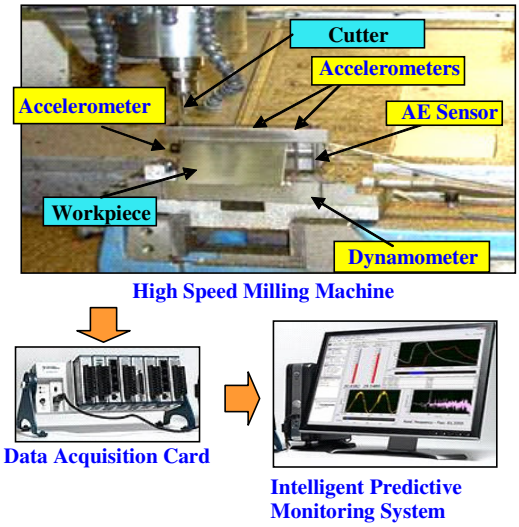

Figure. 2: Tool Conditions monitoring in high speed milling process

The machining tests were carried out in down milling operation with cutting speeds 4.7*m/min* and spindle speed 23,600 rpm. The workpiece length in the feed direction *L* was 108 *mm*. After one horizontal cutting line along the *y*-axis direction (1st), the cutter then retracted to another start with a cutting depth of 0.2 *mm* in *z*-axis (2nd) direction.

In each experiment, the cutter was used to cut the workpiece slope in succession to machine a complete slope surface. The total length of cut for one surface (i.e. 252 passes) was $108mm \times 252 = 27216\,mm$. The cutter's flank wear was measured after a complete 27,216*mm* cutting distance using a LEICA MZ12 microscopy system.

The National Instruments LabVIEW 8.2 is chosen to create a user-friendly graphical user interface (GUI). A GUI was designed to show real-time signals acquired from the 8-channels. With the final GUI, users can observe the signal changes through the continuous graphical displays of the entire milling operation and can select various levels of response when registering anomalies or impending failure.

4.2. Feature Extraction

The tool wear estimation is mainly based on the force features in this work as the cutting force is highly sensitive to tool wear and can be measured with fairly good accuracy (Altintas *et al.*, 1989). The first step is to filter out the noise and remove non-cutting signals using a joint time-frequency distribution algorithm (Li *et al.,* 2007). Then sixteen main features were identified and captured from the force signals using statistical methods (Zeng *et al.,* 2006), as summarized in Table 1.

Table 1: Extraction features from force signals

No	Feature	Notation	Objective
1	Residual Error	re	TBD
2	First Order Differencing	fod	TWD
3	Second Order Differencing	sod	TWD
4	Maximum Force Level	fm	TWD
5	Total Amplitude of Cutting Force	fa	TWD
6	Combined Incremental Force Changes	df	TWD
7	Amplitude Ratio	ra	TWD
8	Standard Deviation of the Force Components in Tool Breakage Zone	fstd	TBD
9	Sum of the Squares of Residual Errors	sre	TBD
10	Peak Rate of Cutting Forces	kpr	TBD
11	Total Harmonic Power	thp	TWD
12	Average Force	Fa	TBD, TWD, TWE
13	Variable Force	vf	TBD
14	Standard Deviation	std	TWD
15	Skew	skew	TWD
16	Kurtosis	kts	TWD

* TBD: Tool Breakage Detection; TWD: Tool Wear Detection; TWE: Tool Wear Estimation.

4.3. Feature Selection through Genetic Algorithms

Although all of the 16 features listed in Table 1 are statistically significant, it has been observed that, beyond a certain point, involvement of all the features leads to an

unsatisfactory calculation performance in establish correlation models. Therefore, selection of most relevant feature subset is necessary for rapid establishment of desired correlation models with acceptable computing performance. The Genetic Algorithms (Beasley *et al.*, 1993) is used for feature subset selection since they are generally effective for rapid global search of large, non-linear and poorly understood spaces.

The basic concept of GAs are designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem. The fundamental mechanisms of the GAs are derived from the evolutionary process include selection, crossover and mutation within chromosomes. Selection occurs on the current population by choosing the highly fit individuals to reproduce. The selected individuals reproduce new individuals as offspring by crossover with other individuals in the population. Mutation may happen in the reproduction. In this way, over many generations, good characteristics spread in the population, mixing and exchanging with other good characteristics as they go. GAs are computer programs that create an environment where populations of data compete and only the fittest survive.

According to Darwin's theory of evolution the best ones survive to create new offspring (children). There are many methods in selecting the best chromosomes. In this work, we choose roulette wheel selection (Obitko, 1998) to evaluate how "good" a feature and the level of association of that feature with the measured tool wear values.

The major design components of the GA system used in this work include the initialisation process, the crossover method, mutation method, roulette wheel selection and an objective fitness function. The input parameters, their representation and the optimum values used are summarized in Table 2.

Table 2: GA parameter setting

Parameter	Representation	Value
Population Size	The size of the population for every generation	20
Crossover points	The number of crossover points to be defined when performing crossover	1
Crossover rate	The chance of crossover being applied to a chromosome	0.7
Mutation rate	The chance of a chromosome being mutated	0.001
Confidence level	A limiting value on the required association *r* value.	0.9
Maximum Number of generation	The maximum number of generations the GA is allowed to evolve to. Also the terminating condition of the algorithm.	200
Number of test sets	The amount of data sets used for the calculation of *r*.	2000

In the feature subset selection system, a fitness function is used to evaluate how "good" a feature and the level of association of that feature with the tool wear. Pearson Correlation Coefficient (PCC) is adopted to give a score to each data set. PCC is a statistical measure of interdependence of two or more random variables. The PCC correlation coefficient is defined as *r*

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}} \quad (1)$$

The expression can be simplified into the following form:

$$r = \frac{S_{xy}}{S_x \bullet S_y} \quad (2)$$

where S_{xy} is the standard deviation of $X \bullet Y$, and S_x and S_y refers to the standard deviation of X and Y respectively. In our case, x represents the signal feature data and y represent the corresponding tool wear data.

As the magnitude of r is needed for scoring of a particular bit of the chromosome, and its value can be negative, the absolute of r is returned as the score of that feature. To filter out low r value as zero association, a confidence level (set at 0.90) is introduced so that only r values above the confidence level are considered in the total score. Therefore, set bits that have a score less than the confidence level is given a score of 0, and the bit is reset to 0. In this way, the results only consist of selected features whose r values are higher than the confidence level.

There are three steps to implement a (GA):

(1) Randomly generate a initial population with 20 chromosomes represented with binary encoding. Each chromosome consists of 16 binary bits, corresponding to the number of features extracted from the experimental data, as shown Table 3. 0 means that feature is ignored while 1 the feature survives and is relevant.

Table 3: Initial population

Chromosome 1	1011001011001010
……	……
Chromosome 20	1111111000001100

(2) Create new population by repeating the following steps until 200 populations are generated::

a. Merge the chromosomes in the initial population with the following rule to generate a new chromosome: if a feature bit in any of the 20 chromosomes has a value of 1, the corresponding feature bit in the new chromosome is 1, otherwise 0.

b. Randomly select 2000 sets of feature data together with their corresponding interpolated tool wear data from the available 52,800 data sets and calculate the fitness *r* for each feature in the new

chromosome if the corresponding bit is 1.

c. Put those features with bit 1 into a roulette wheel which is set up with partition area proportional to the value of the fitness r values. Spin the wheel for to generate two 'parents' consisting also 16 bits, spinning 20 times (same as the number of the chromosomes) for each parent. A winner feature in any spinning will result in a bit 1 while the failed ones with 0.

d. Crossover to produce 'offspring' chromosomes - 2 new offspring (children) chromosomes are formed by a crossover method from the two parents as shown Table 4.

Table 4: Cross over example

Chromosome 1	**11011 || 00100110110**
Chromosome 2	11011 || 11000011110
Offspring 1	**11011** || 11000011110
Offspring 2	11011 || **00100110110**

e. Mutation - switch a few randomly chosen bits from 1 to 0 or from 0 to1. Mutation is intended to prevent falling of all solutions in the population into a local optimum.

f. Repeat steps c and d until 20 new offspring chromosomes are generated. Take the new offspring chromosomes as a new population.

(3) Select the features according to their fitness values in the 200th population. The features are selected if their r values are above the confidence level.

Repeat steps (1) to (3) for 50 times for repeatability test. The feature selection results are shown in Table 5.

Table 5: Experiment result

	Data Set (50 runs)				
		Test No 1	Test No 2	Test No 3	Test No 4
Features	re				
	fod				
	sod				
	fm	49	47	47	44
	fa	50	48	47	47
	df				
	ra				
	fstd				
	sre				
	kpr				
	thp				
	Fa	49	48	48	47
	vf				
	std	47	49	49	40
	skew				48
	kts				47

To give an example for understand the meaning of the result, feature "*fm*" has been selected for 49 times in the 50 runs in the first experiment, 47, 47 and 44 for another three experiments. It is obvious that "*fm*" is significant for the tool wear. According to the results shown in Table 5, four features, { *fm, fa, Fa, std* }, have the most significant influence to generate correlation models between tool wear and the features with fairly good accuracy (Zhou *et al.*, 2006).

4.4. Self Learning, Rule Generation and Modelling

The selected feature data and measured tool wear are then stored into the database. A total of 52,800 sets of feature data were generated from raw signals, acquisited from about 490 m of cutting by the cutter in its entire life, matched with the interpolated tool wear data, half of which are used for rule training and the remaining for testing. The data samples were normalised to the range of {0,1} for further uses.

The FNN establishment starts from feeding the training data sets, one at a time, to the network from layer 1 as represented by eq. (1). the input data are fuzzified in layer 2 according to eq. (2). The FNN then goes to its self learning of fuzzy membership parameters (mean and variance) with SOM (eqs. 6, 7 and 8). Fig. 3 shows the membership distributions of X_1 before and after the self-learning. The x-axis represents the normalization of the selected feature. With the adjusted means and variances, the data sets were then clustered according to its similarity.

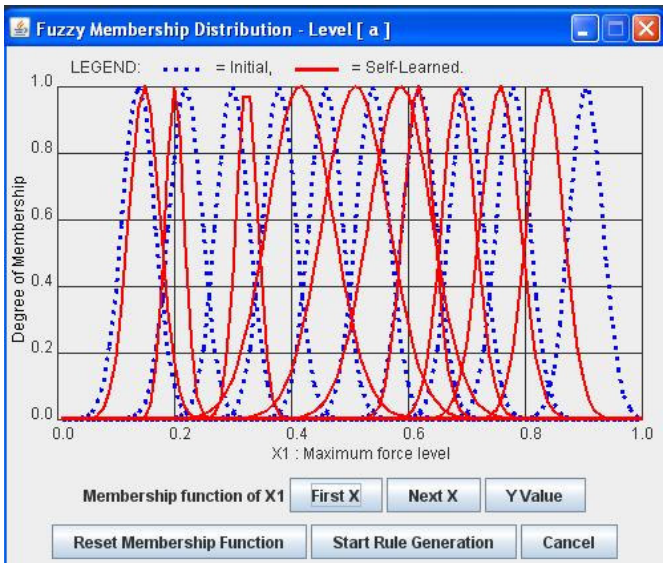

Figure 3: Before-after self-learned fuzzy membership distributions

The number of membership function nodes at the 2nd and 4th layer were determined through a trial-and-error procedure. Up to more than 10 combinations of node numbers in the two layers are selected manually and run through the system. The system will determine the best membership nodes combination according to final

performance results. In this case study, there were 10 nodes (very low, low, …medium, …high, very high) for x_1, 12 nodes for x_2, 8 nodes for x_3 and 12 nodes for x_4.

After the membership functions have been constructed, the next stage is to identify the fuzzy rules at the 3rd layer as represented by eq. (3). The identification process starts with a fully-connected neural network structure. The total number of initial rules is determined by $T(x_1)\cdot T(x_2)\cdot\ldots T(x_k)\cdot\ldots T(x_p)$ where $T(x_k)$ is the number of membership functions of the *k*th input variable (Li *et al.*, 1999). Therefore, the maximum number of possible fuzzy rules in the present case was 10×12×8×12 =11,520. However, many rules may not correspond to any real situation. To save data storage space and speed up the calculation operation, we only store those rules where the training data have been gone through the "winner" Node c as shown in Figure 4. The Node c is identified as a winner rule node of the fuzzy min-max operation in layer 3 and the relevant nodes in layer 2. In this case study, only 83 rules are real which have been trained and retained to represent the effective data clusters.

Table 6: Examples of the learned fuzzy rules

Rule #	No of training data	No of testing data		Rule #	No of training data	No of testing data
1	24152	24152		2275	13	13
11	65	67		3332	20	23
12	15	14		3333	78	76
13	12	11		3334	71	72
21	42	46		3343	145	145
24	17	16		3344	26	24
31	101	99		3345	27	27
131	21	24		3353	31	28
1111	61	61		3354	87	90
1121	55	54		3355	13	15
1124	57	58		3364	70	73
1131	72	66		3365	25	27
1132	80	81		3366	16	14

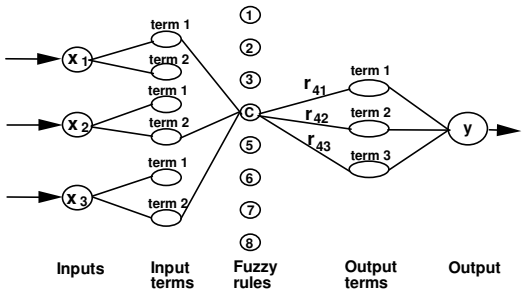

Figure. 4: Example of fuzzy rule generation and clustering process

Table 6 gives examples of the learned fuzzy rules. The rules here play the function of data clustering. Each rule captures similar data patterns. For example, Rule #11 holds 65 training data sets which have the same patterns like {if x1 is high, x2 low, x3 medium, x4 high, then y medium}. The rule identification and clustering process are illustrated A fuzzy OR operation is performed at layer 4 according to eq.(4). The output is defuzzified at layer 5 as shown by eq. (5).

Once the clustering and rule generation are completed, the FNN continues on correlation modeling with supervised leaning algorithm as shown in eq. (13) to fine tune the rule values till mean squared error reaches an accepted level.

4.5. Prediction of Tool Wear and Retraining

After the FNN has been trained, the test data were fed to it to obtain the forecasted values F_t. For a given new input data set, if the fuzzy rule generated does not match any of the existing rules, the system will choose as replacement a rule that is closest to the rule generated, and will proceed to prediction as usual. This will unavoidably introduce errors into the prediction result. It is therefore important that a sufficiently large pool of data samples be used during the training step to ensure that the trained rules are comperhensive. In general, the more the FNN is retrained, the more accurate it will be. The FNN can be retrained whenever new data become available. Retraining involves repeating stages 3.1 to 3.2 (see Section 3) to reconstruct membership functions, identify new fuzzy rules, if any, and update rule values.

In this case study, the prediction accuracy is gauged with three commonly used measures, namely Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE) and R-squared Values (R^2) (Damodar *et al.*, 1995). The error measures for the 28,240 sets of test data were found to be: MSE = 1.743 x 10^{-5}, MAPE = 0.01 and R^2 = 0.9957 as shown in Table 7.

Through the FNN clustering, modelling and prediction test, it was noted that the amount of training time required was dependent on the number of fuzzy membership nodes assigned to each of the features. Therefore, for higher level of accuracy more membership items are required which will inevitably slow down the learning speed. However, the number of fuzzy membership items to be defined is not linear to the accuracy achieved. In other words, high accuracy is not guaranteed when more fuzzy membership items are defined. As such, auto-defining and optimizing the number of fuzzy membership items are still an unsolved problem.

Table 7: Prediction Performance of FNN

Error Measures	**FNN**
MSE	1.743 x 10^{-5}
MAPE	0.010
R^2	0.9957

4.6. Comparison with MRM, BPNN and RBFN

The same data sets were used for tool wear prediction with BPNN and RBFN and conventional MRM. Table 8 summarizes the parametric set-ups and training

performance of the FNN, BPNN and RBFN. The BPNN model used comprises of 4 layers: 1 input layer, 2 hidden layers and 1 output layer while the RBFN model has 3 layers: 1 input, 1 prototype and 1 output layer. The learning rules and transfer functions used for the BPNN are Delta and Sigmoid, respectively while the corresponding items for RBFN are Norm-Cum-Delta and Gaussian.

Table 8: Parametric set-ups and training performance of FNN, BPNN and RBFN

Parameters	FNN	BPNN	RBFN
Learning Rates	0.5	0.5	0.5
Momentum	-	0.4	0.4
Learning Rules	SOM	Delta	Norm-Cum-Delta
Transfer Function	Gaussian	Sigmoid	Gaussian
Number of Layers	5	4	3
Number of Nodes each layer	4, 42, 83,83,1	4, 5, 5, 1	4, 50, 1
Number of samples	28,400	28,400	10,000
MSE	1.74×10^{-5}	1.34×10^{-3}	4.43×10^{-2}

Table 9 compares the prediction performance and training time used for FNN, MRM, BPNN, RBFN methods. The FNN has the best performance with the smallest MSE and MAPE and comparable R^2. The RBFN has the worst performance as compared to the rest of the experiment results even it is shortest training time.

Table 9: Prediction performances of MRM, BPNN, RBFN and FNN

Error Measures	MRM	BPNN	RBFN	FNN
MSE	2.08×10^{-3}	1.34×10^{-3}	4.43×10^{-2}	**1.74×10^{-5}**
MAPE	0.0934	0.06458	0.34053	**0.010**
R^2	0.9757	0.9836	0.5793	**0.9957**
Training Time(hrs.)	-	12	-	**3**

The established FNN reference model has been tested with data from experiments with another three 6mm ball nose cutters under the same condition to evaluate its repeatability and the results are shown in Table 10. The results show that the FNN model established through the test outcome with one particular cutter exhibited quite good repeatability on the data generated with other three cutters. The predicted tool wear values were over 90% of their real values. Table 6 compares also the prediction performance of FNN and MRM methods. It can be seen that FNN performance is better than MRM.

Table 10: Applicability of the FNN model to data generated from different cutters under the same machining conditions and comparison of its prediction performance with FNN and MRM methods

Test Results using Fuzzy Neural Networks			
Cutters	G15N7	G14N8	G12N9
MSE	0.00013221	9.57948E-05	0.000560187
MAPE	0.243240049	0.200309348	0.318237011
R^2	0.9541	0.9145	0.9071
Test Results using Multiple Regression Models			
Cutters	G15N7	G14N8	G12N9
MSE	0.000438	4.07E-05	0.001299
MAPE	0.693563	0.789606	1.237696
R^2	0.8476	0.7597	0.4206

The ultimate aim of the tool wear estimation is to predict the remaining life of a cutter during the process. We choose to use the distance that a cutter can cut through the work piece to represent the tool life. It is easy to understand the tool life is correlated to the tool wear and a cutter will come to its life end when it cannot cut the work piece anymore with the same required finishing quality.

Figure 5 gives the correlation between the tool wear and tool life, where the tool wear is predicted based on signal features {*fm, fa, Fa, std*}. The figure shows that there are three points (D1 D2 and D3) at around cutting distance of 370m, 427m and 468m with predicted tool wear of 0.21mm and 0.33mm and 0.38mm, respectively, where the tool wear rate changes significantly from the previous data. These points could be taken as critical alarms indicating the degree of tool degradation. In our monitoring system with the FNN model, if a big tool wear changes before and after a cutting distance is predicted, the tool life with this distance is identified as a critical points immediately. Once the system gives the degradation of D3 level, one should prepare to stop the cutting to avoid costly work piece surface damages.

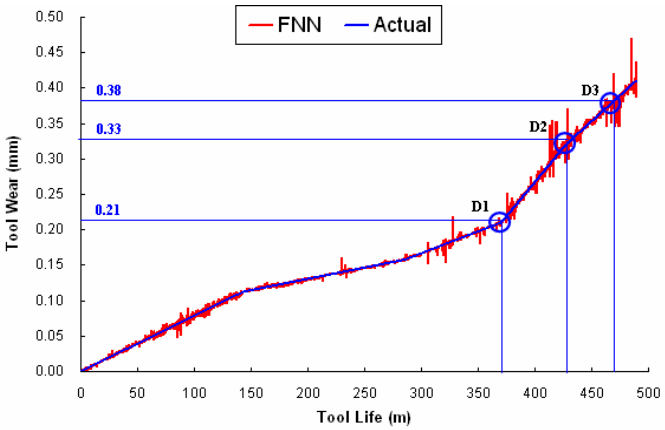

Figure 5: Correlation between predicted tool wear and tool degradation

In comparison with the experiment results with conventional neural network and multiple regression methods, it can be observed that the FNN outperforms the others in solving the problem of tool wear prediction on the dry milling operation. However, the limitation of the FNN lies in the fact that the optimal number of fuzzy membership items cannot be automatically defined. The reliability of FNN under different cutting conditions needs to be further investigated also.

5. CONCLUSIONS

A FNN algorithm is developed for tool degradation monitoring through tool wear prediction. The FNN is basically a multi-layered fuzzy-rule-based neural network which integrates the basic elements and functions of a traditional fuzzy logic inference into a neural network structure. The fuzzy rules help to speed up the learning process of the complex conventional neural network structure and improve the accuracy in prediction and rate of convergence. The viability of the FNN technique is tested with the prediction of tool life in a dry milling operation. The repeatability of the FNN model established under a fixed cutting condition is demonstrated by comparing the prediction data with the real data from different cutters. A comparison was made on prediction performances of different models established with the same set of experimental data. It is shown that the FNN is superior to conventional MRM, BPNN and RBFN in prediction accuracy and BPNN in learning speed.

REFERENCES

(Aliustaoglu *et al.*, 2009) C. Aliustaoglu, H. M. Ertunc, and H. Ocak. Tool wear condition monitoring using a sensor fusion model based on fuzzy inference system. Mechanical Systems and Signal Processing, 23(2):539–546, February 2009.

(Altintas *et al.*, 1989) Y. Altintas, et al, In process Detection of Tool Failure in Milling Using Cutting Force Models, J. Engng Ind., Vol. 111, pp. 149-157, 1989.

(Beasley *et al.*, 1993) D. Beasley, D.R. Bull and R.R. Martin. An Overview of Genetic Algorithm Fundamentals, 15(2):58-69.

(Damodar *et al.*, 1995) D. N. Gujarati, "Basic Econometrics", Third Edition, McGRAW-HILL INTERNATIONAL EDITIONS ,1995.

(Dong *et al.*, 2004) J.F. Dong, Y.S. Wong and G. S, Hong, Bayesian Support Vector Classification for Machining Process monitoring, Fourth International ICSC Symposium on Engineering of Intelligent Systems, Portugal, 29th Feb. – 3rd Mar. 2004.

(Er *et al.,* 2003) M. J. Er and Y. Gao, *Online Adaptive Fuzzy Neural Identification and Control of Nonlinear Dynamic Systems*, in Fusion of Soft Computing and Hard Computing for Autonomous Robotic Systems edited by C. Zhou, D. Maravall and D. Ruan, New York, Heidelberg: Physica-Verlag, pp. 373-402, 2003.

(Haber *et al.*, 2003) R. E. Haber, A. Alique, "Intelligent process supervision for predicting tool wear in machining processes", Mechatronics Vol. 13, PP 825–849, 2003

(Huang *et al.*, 2000) P.T. Huang, and J.C. Chen, "Neural Network-Based Tool Breakage Monitoring System for End Milling Operations", Journal of Industrial Technology, 2000.

(Ho *et al.*, 2009) W.H. Ho, J.T. Tsai, B.T. Lin, and J.H. Chou. Adaptive network-based fuzzy inference system for prediction of surface roughness in end milling process using hybrid taguchi-genetic learning algorithm. Expert Systems with Applications, 36(2, Part 2):3216–3222, March 2009.

(Kohonen, 1988) T. Kohonen, Self-organization and Associative Memory, Springer-Verlag, Berlin, 1988.

(Li *et al.*, 1999) Li, X.,.Ang, C.L , Gay, R. , "An Intelligent Business Forecaster for Strategic Business Planning", Journal of Forecasting, Vol. 18, pp. 181-204, 1999.

(Li *et al.*, 2006) X. Li, J. Zhou, H.Zeng, Y.S. Wong, G.S. Hong, "An Intelligent Predictive Engine for Milling Machine Prognostic Monitoring", CD Proceeding of the 4th IEEE International Conference on Industrial Informatics, Singapore, pp 1075-1080, 2006.

(Li *et al.*, 2007) X. Li, H. Zeng, J.H. Zhou, S. Huang, T.B.Thoe, K.C. Shaw and B.S.Lim, "Milti-Modal Sensing and Correlation Modelling for Condition-based Monitoring in Milling Machine" SIMTech Technical Report, Volume 8 Number 1, pg 50-56, Jan-Mar, 2007.

(Li *et al.*, 1996) S. Li and M. A. Elbestawi, "Tool condition monitoring in machining by fuzzy neural networks", Journal of Dynamic Systems, Measurement, and Control, Transactions of the ASME 118, 665–672, 1996.

(Lo *et al.*, 2003) S.P. Lo. An adaptive-network based fuzzy inference system for prediction of workpiece surface roughness in end milling. Journal of Materials Processing Technology, 142(3):665–675, December 2003.

(Obitko, 1998) Marek Obitko, "Roulette Wheel Selection", http://www.obitko.com/tutorials/genetic-algorithms/selection.php, 1998

(Rumelhart *et al.*, 1986) D.E. Rumelhart, G.E. Hinton, and R.J. Williams, Learning internal representations by error propagation, Parallel Distributed Processing, Vol. 1, MIT Press, Cambridge, 1986.

(Kosko, 1992) B. Kosko, Neural networks and fuzzy system, Prentice-Hall International Editions, 1992.

(Sick, 2002) B. Sick, Review on-Line and indirect tool wear monitoring in turning with artificial neural networks: A review of more than a decade of research, Mechanical Systems and Signal Processing, 16(4), 487–546, 2002.

(Vallejo Jr *et al.*, 2005) A. G. Vallejo Jr, J. A. N. Flores, R. M. Menendez, L. E. Sucar, C. A. Rodriguez in M. Lazo and A. Sanfeliu (Eds.): Tool-Wear Monitoring Based

on Continuous Hidden Markov Models, CIARP 2005, LNCS 3773, pp. 880–890, 2005. c_Springer-Verlag Berlin Heidelberg, 2005

(Wang *et al.*, 2001) L. Wang, M.G. Mehrabi and E.K. Jr, "Tool Wear monitoring in Reconfigurable Machining Systems Through Wavelet Analysis", Transactions of NAMRI, pp. 399-406. 2001.

(Yamaguchi *et al.*, 2007) T. Yamaguchi, M. Higuchi, S. Shimada and T. Kaneeda, "Tool life monitoring during the diamond turning of electroless Ni–P", Precision Engineering, Vol 31, 196–201, 2007.

(Yang *et al.*, 2006) L.D. Yang, J.C. Chen, H.M. Chow, and C.T. Lin. Fuzzy-netsbased in-process surface roughness adaptive control system in endmilling operations. International Journal of Advanced Manufacturing Technology, 28(3-4):236–248, 2006.

(Zhou *et al.*, 2005) J.H. Zhou, X. Li, A. Andernroomer, H. Zeng, K.M. Goh, Y.S.Wong, G.S.Hong, "Intelligent Prediction Monitoring System for Predictive Maintenance in Manufacturing", Proceedings of the 31st Annual Conference of the IEEE Industrial Electronics Society –IECON'05, pp 2314-2319, 2005.

(Zhou *et al.*, 2006) J.H. Zhou, X. Li, S.G. Han, and W.K. Ng, "Genetic Algorithms for Feature Subset Selection in Equipment Fault Diagnosis", CD proceeding of the World Congress on Engineering Asset Management (WCEAM), 11-14, Gold Coast, Australia, pp 1114-1123, 2006.

(Zeng *et al.*, 2006) H. Zeng, T.B. Thoe, X. Li, J.H. Zhou, "Multi-modal Sensing for Machine Health Monitoring in High Speed Machining", CD Proceeding of the 4th IEEE International Conference on Industrial Informatics, 16-18 August 2006, Singapore, pp 1217-1222.

Xiang Li (Singapore Institute of Manufacturing Technology) is a research scientist at the Singapore Institute of Manufacturing Technology, Singapore. She received her Ph.D from Nanyang Technological University, Singapore, in 2000. She has more than 15 years of experience in research and applications of data mining, artificial intelligence and statistical analysis, such as neural networks, fuzzy logic systems, data clustering and multiple regression modeling. She has led and was involved in many research and industrial projects, including computational intelligence for tool and surface integrity monitoring in precision machining, automould designs using intelligent optimization and simulation for semiconductor equipment builder, automation and integration of manufacturing executive system (MES) for semiconductor in Taiwan, data warehousing for semiconductor in Singapore, rule-based AEC object library for building constructor, customer demand knowledge discovery in new product design for electronics in Singapore and web-based intelligent forecasting system. Her current research interest include data mining, machine learning, computational intelligence, decision support systems, intelligent forecasting, knowledge-based systems, real time equipment condition monitoring and performance prediction. She is a member of the Decision Sciences Institute, USA and and member of IEEE.

Beng-Siong Lim (Singapore Institute of Manufacturing Technology) is a senior research scientist at the Singapore Institute of Manufacturing Technology, Singapore. He received PhD from the Production Engineering and Production Management, The University of Nottingham, England, in 1986. He led several teams to develop intelligent system for the characterization of cutters in the green machining of aero-engine material and hardened steel; EDM electrode verification and offset insertion; flexible manufacturing systems; prosthetic knee joint implants; knowledge-based systems for daily production scheduling, process planning for progressive dies and modular fixture design; automated guided vehicles and robotics off-line programming. His current research interests include proactive performance engineering of manufacturing system; Development of reference models for the infusion of manufacturing knowledge into tools, process, equipment and systems; Multi-sensors fusion, performance, evaluation and application characterization; Process and shop-floor monitoring, diagnostics and control; Equipment, plant and tools prognosis; EDM, tool engineering and process planning.

Junhong Zhou (Singapore Institute of Manufacturing Technology) is a senior research engineer at the Singapore Institute of Manufacturing Technology, Singapore. She received her MEng from the Department of Electrical Engineering, National University of Singapore, Singapore. Her industrial experience includes intelligent monitoring system for grinding process, adaptive control, shop-floor Supervisory Control And Data Acquisition (SCADA) systems, enterprise resource management system, simulation study of factory layouts and optimization of shop-floor utilization. Her expertise and research interests include artificial intelligence, feature extraction and selection, fault diagnosis and prognosis, simulation, SCADA systems.

Sheng Huang (Singapore Institute of Manufacturing Technology) is a research engineer in the Manufacturing Execution and Control Group is involved in the research and development of tool condition monitoring, fault characterization, proactive real time diagnosis, fault diagnostics & prognostics for equipment and system, rapid embedded application prototyping and intelligent sensor-node data processing.

Si-Jie Phua (Singapore Institute of Manufacturing Technology) is a research engineer at the Singapore Institue of Manufacturing Technology, Singapore. He received his BEng (1st Honors) from the School of Computer Engineering, Nanyang Technology University, Singapore, in 2006. He is currently a PhD candidate in the School of Computer Engineering at Nanyang Technology University.

His current research interests are data stream mining and sensor signal analysis.

Kah-Chuan Shaw (Singapore Institute of Manufacturing Technology) is a senior research officer in the machining technology group is involved in the research and development of precision machining technology, micro-machining and micro-ECM.

Meng-Joo Er (Nanyang Technological University, Singapore) is a Director of the Intelligent Systems Centre (NTU) and a full Professor in the School of Electrical and Electronic Engineering (EEE), NTU. He interests the research include control theory and applications, fuzzy logic and neural networks, computational intelligence, robotics and automation, sensor networks, cognitive systems and biomedical engineering. He has authored two books entitled "Dynamic Fuzzy Neural Networks: Architectures, Algorithms and Applications" and "Engineering Mathematics with Real-World Applications" published by McGraw Hill in 2003 and 2005 respectively, seven book chapters and more than 350 refereed journal and conference papers works in his research areas of interest.