FUZZY REGRESSION MODELING FOR TOOL PERFORMANCE PREDICTION AND DEGRADATION DETECTION

X. LI[*]
Singapore Institute of Manufacturing Technology
71 Nanyang Drive, Singapore 638075
xli@simtech.a-star.edu.sg

M. J. ER
School of Electrical & Electronic Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798
EMJER@ntu.edu.sg

B. S. LIM[†], J. H. ZHOU[‡] and O. P. GAN[§]
Singapore Institute of Manufacturing Technology
71 Nanyang Drive, Singapore 638075
[†]*bslim@simtech.a-star.edu.sg*
[‡]*jzhou@simtech.a-star.edu.sg*
[§]*opgan@simtech.a-star.edu.sg*

L. RUTKOWSKI
Department of Computer Engineering
Częstochowa University of Technology
al. Armii Krajowej 36, 42-200 Częstochowa, Poland

Academy of Management (SWSPiZ)
Institute of Information Technology
Sienkiewicza 9, 90-113 Lódź, Poland
lrutko@kik.pcz.czest.pl

In this paper, the viability of using Fuzzy-Rule-Based Regression Modeling (FRM) algorithm for tool performance and degradation detection is investigated. The FRM is developed based on a multi-layered fuzzy-rule-based hybrid system with Multiple Regression Models (MRM) embedded into a fuzzy logic inference engine that employs Self Organizing Maps (SOM) for clustering. The FRM converts a complex nonlinear problem to a simplified linear format in order to further increase the accuracy in prediction and rate of convergence. The efficacy of the proposed FRM is tested through a case study — namely to predict the remaining useful life of a ball nose milling cutter during a dry machining process of hardened tool steel with a hardness of 52–54 HRc. A comparative study is further made between four predictive models using the same set of experimental data. It is shown that the FRM is superior as compared with conventional MRM, Back Propagation Neural Networks (BPNN) and Radial Basis Function Networks (RBFN) in terms of prediction accuracy and learning speed.

Keywords: Fuzzy logic; multiple regression modeling; self organizing maps; tool performance prediction; cutters degradation detection.

[*]Corresponding author.

1. Introduction

The applications of Predictive Intelligence (PI) for real-time/online machine fault detection and prognostic is a relatively new development particularly for the monitoring of critical machinery and equipment. Research efforts in this area have increased tremendously over the last few years due to the need to ensure high availability of critical equipment, improved reliability and performance, particularly for unmanned factory operation. Over the past decade, PI applications have evolved from simple laboratory demonstrations to sophisticated commercial products. Many novel Artificial Intelligence (AI) or machine learning algorithms have been developed and applied to many application areas in recent years, for example, neural network models for earthquake magnitude prediction using multiple seismicity indicators,[1] online detection of the modality of complex valued real world signals.[2] New theories on machine learning have shed some light relating to some fundamental issues, such as self-organizing mixture autoregressive model for non-stationary time series prediction[3]; improving supervised learning by adapting the problem to the learner[4]; neural dynamic model[5–9,11–14]; dynamic wavelet neural networks for modeling complex time-dependent phenomena such as vibration control of structures under earthquake loading[15–18]; wavelet-chaos-neural network methodology for EEG analysis[19]; probabilistic neural networks[20–22]; complex-valued neural networks[23–33]; spiking neural networks[34–36]; No-reference video quality measurement with support vector regression.[37,38] The used of PI for real-time machine fault detection, tool performance prediction, degradation detection and optimal control is a prime topic for numerous investigations. Vachtsevanos *et al.*[39] and Lee[40] demonstrated the viability of using a dynamic wavelet neural networks prognostic framework and virtual sensors for the prediction and detection of bearing failure. Prope *et al.*[41] developed a real-time architecture for prognostic enhancements systems and applied it to the detection of incipient failure conditions and prediction of the remaining useful life of the US navy's existing stockade of artillery shells.

In the high precision machining industry, major research efforts are focused on the development of self-adjusting and integrated systems capable of monitoring performance degradation and work piece integrity under various operational conditions with minimal operator's supervision. Most of the key research work in this research field is focus on the development of techniques and approaches for analytical forecast, dynamic structure identification, monitoring and adaptive control. The viability of these techniques are tested in applications such as tool breakage detection in milling operations,[42] tool wear monitoring in reconfigurable machining systems[43] and selection of relevant features for tool wear in face milling processes.[44] In the machining process, many variables and parameters affect the work piece integrity and tool performance over the production regime. As such, it is common for researchers to install a suit of accelerometer, dynamometer, acoustics emission and other sensors at critical locations to allow in-situ signals to be captured, processed, analyzed and transformed into useful reference models for condition and performance monitoring.[45,46] Examples of these models include dynamic models for diffusion and adhesive wear models[47]; linear steady-state models for tool wear and cutting forces[48]; auto-regressive chatter and tool breakage models.[49,50] Other advanced methods include hidden Markov models,[51] Bayesian networks,[52] neural networks and fuzzy neural networks[53–55] for tool condition monitoring.

In the development of tool performance monitoring systems, convergence speed and adaptive learning capability are two important issues[56] to allow for greater effectiveness and robustness. The principal constituents to concurrently address these issues include the use of fuzzy logic reasoning for imprecise data,[57–60] neural networks for learning and probability reasoning for uncertainty mapping.[61–64] More investigations have been done for defects quickest detection,[65] statistical quality control[66] and nonparametric detection.[67–69] However, in order to take advantage of these technologies for predictive performance and integrity monitoring applications collectively, a systematic, efficient and robust approach is critically required.

Linear regression technique has been demonstrated for the prediction of tool-wear with respect to a specific training dataset, but its accuracy will be compromised if the data is nonlinear. Neural networks or fuzzy neural networks are extremely good candidates for solving non-linear problems, but require large training sample, longer training time and suffer from local minima issues.[70–75] To circumvent these weaknesses, a new hybrid intelligent method, termed Fuzzy-rule-based Regression Modeling (FRM) algorithm, is proposed. The FRM is developed based on a multi-layered fuzzy-rule-based hybrid system with Multiple Regression Models (MRM) embedded into fuzzy logic inference employing Self Organizing Maps (SOM).[76] The FRM converts a complex non-linear problem to a simplified linear format and adopts a divide and conquer methodology by decomposing it into smaller, less complex and more manageable sub-problems that allows higher prediction accuracy and faster learning rate. The viability of the FRM technique is tested for the prediction of tool life in a dry milling operation. A comparative study was made in relation to Multi-regression models, Backpropagation Neural Networks (BPNN)[77,78] and Radial Basis Functional Networks (RBFN)[79–87] using the same set of experimental data.

2. The FRM Architecture

The architecture of the FRM is shown in Fig. 1. Basically, it is a five-layer fuzzy-rule- based regression modeling hybrid system. In accordance with the common network notation, a node in any layer n of the network termed net^n performs a specific operation.[88]

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

$$net_k^1 = u_{kq} \tag{1}$$

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.[89] The membership functions are normal distributions in the range of 0 and 1(inclusive 1), governed by

$$net_q^2 = e^{(-\frac{u_{kq} - m_{kq}}{\sigma_{kq}})^2} = u_{qi} \tag{2}$$

where $i = 1, 2, \ldots, h$. The terms m_{kq} and σ_{kq} are the mean and variance of the input membership function respectively.

Layer 3: The nodes in this layer perform a fuzzy min-max operation on the node inputs, i.e. a fuzzy AND operation followed by a fuzzy OR operation.[89]

$$net_i^3 = \min\{u_{qi}\} = u_i;$$

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

$$Node = \max(\mathrm{net}_1^3, \mathrm{net}_2^3, \ldots, \mathrm{net}_h^3) \tag{3}$$

where $c \in \{1, 2, \ldots, h\}$ is the node number in the layer 3 with the maximum net$_i^3$ value. Node c is called the winner rule node of the fuzzy min-max operation.

Layer 4: Each node in this layer performs correlation modeling. It builds up a Multiple Regression Model

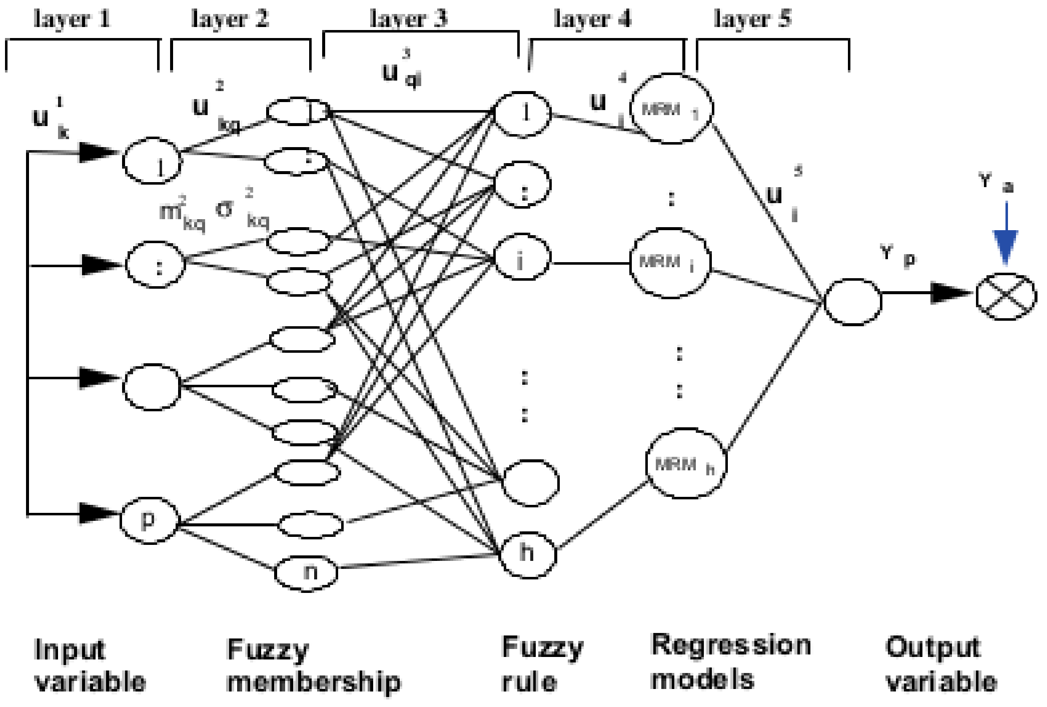

Fig. 1. The architecture of FRM.

(MRM) between input variables and output target with the data set belonging to node c cluster at the 3rd layer as shown in Eq. (4). The nodes in this layer only link with the nodes at the third layer where rules have been trained. According to the assumption of statistics, a MRM is not carried out in a rule node if the data sets are less than 10 samples so that there is no linkage between the fuzzy-rule layer and regression model layer till the number of data sample is more than ten. The node ID at this layer is the same as rule mapping with the rule layer, i.e. we have

$$u_j = u_{ci} = a_0 + \sum_{k=1}^{p} a_k x_k \tag{4}$$

where $j = 1, 2, \ldots, h$, $k = 1, 2, \ldots, p$ (node number of the 1st layer), u_{ci} is the output value of the winner node c at the 3rd layer.

Layer 5: The output at layer 5 is the predicted result from MRM and it given by

$$Y_p = \hat{y}(t) = u_j \tag{5}$$

where Y_p is the predicted output target.

3. Working Procedures of FRM

The process starts with self-organized learning to establish the membership functions for each of the input and output variables. This is followed by identification of the fuzzy rules that are associated with the respective input-output data sets used. Once these have been done, the winner rule node clusters the input-output data sets with similarity of fuzzy membership patterns and builds their regression model. Each fuzzy rule has its own regression model. The FRM network then used these regression models for prediction and the results obtained are used for retraining of fuzzy rules. The setting-up process thus involves five stages:

(1) Self-organized learning.
(2) Identification of fuzzy rules.
(3) Fuzzy SOM clustering
(4) MRM correlation modeling with clusters.
(5) Prediction and fuzzy rule retraining.

3.1. *Self-organized learning*

The Kohonen's Feature Maps algorithm[90] is used here to find the number of membership functions and their respective means and variances of each node at layer 2. The salient features of the algorithm are as follows:

For a given set of data $X = (x_1, x_2, \ldots, x_n)$, initial mean values $m_1, m_2, \ldots, m_k$ 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 as follows:

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

where m_c is the mean with which the datum x_j associates. Data groupings and the mean values are optimized by the following iterative process:

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

$$m_c(t+1) = m_c(t) + \alpha(t)[x_j(t) - m_c(t)] \tag{7}$$

if x_j belongs to the grouping of m_c, and

$$m_c(t+1) = m_c(t) \tag{8}$$

if x_j does not belong to the grouping of m_c.

Note that $\alpha(t)$ $[0 < \alpha(t) < 1]$ is a monotonically decreasing scalar learning rate. The iteration stops at 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) as follows:

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

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

For a set of given input variables, the number of initial mean values $(m_1, m_2, \ldots, m_k)$ 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 improvement in accuracy.

3.2. *Identification of fuzzy rules*

After the membership functions have been constructed, the next stage is to identify the fuzzy rules using the same set of data samples. The identification process starts with a fully-connected neural network structure. The total number of initial rules is bounded above by $T(x_1)$, $T(x_2), \ldots, T(x_k), \ldots, T(x_p)$ where $T(x_k)$ is the number of membership functions of the kth input variable.[88]

A set of input data are fed to the network from layer 1. They are fuzzified in layer 2. A fuzzy AND operation is performed on the fuzzified data in layer 3, followed by a fuzzy OR operation. 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. The rule identification and clustering process are illustrated in Fig. 2.

Once rule node c identified, a rule, or say cluster generated with the links in this layer represents the input of the winner node. In other words, winner node c clusters all the data sets which have the similarity of fuzzy membership patterns according to fuzzy inference and SOM learning.

3.3. *Multiple regression modeling with clusters*

Although the overall relationship between an output and its related inputs could be very complex, the relationship can be linear or close to linear within the specific small range of variation of the inputs after fuzzy clustering[91] which collects all similar data patterns within a rule. Therefore, the correlation among the inputs and output within the specific range corresponding to each rule is built up using MRM is applied with the relevant data sets, as shown in Fig. 3. Commonly accepted t-test, F-test, D-W test and coefficient determination r^2 are carried out to verify the MRM's statistical significance.[92]

The nodes at layer 3 convert a complex non-linear problem to a simplified linear format with multiple regression models. When the number of data sets is less than 10, regression modeling is skipped till the number is more than 10 data sets according to MRM assumption of statistics.

3.4. *Prediction and rule retraining*

Once the membership functions have been constructed, the fuzzy rules generated with their clusters and the regression model built each cluster, and the FRM network is ready for prediction. For a given input data set, if the fuzzy rule identified during

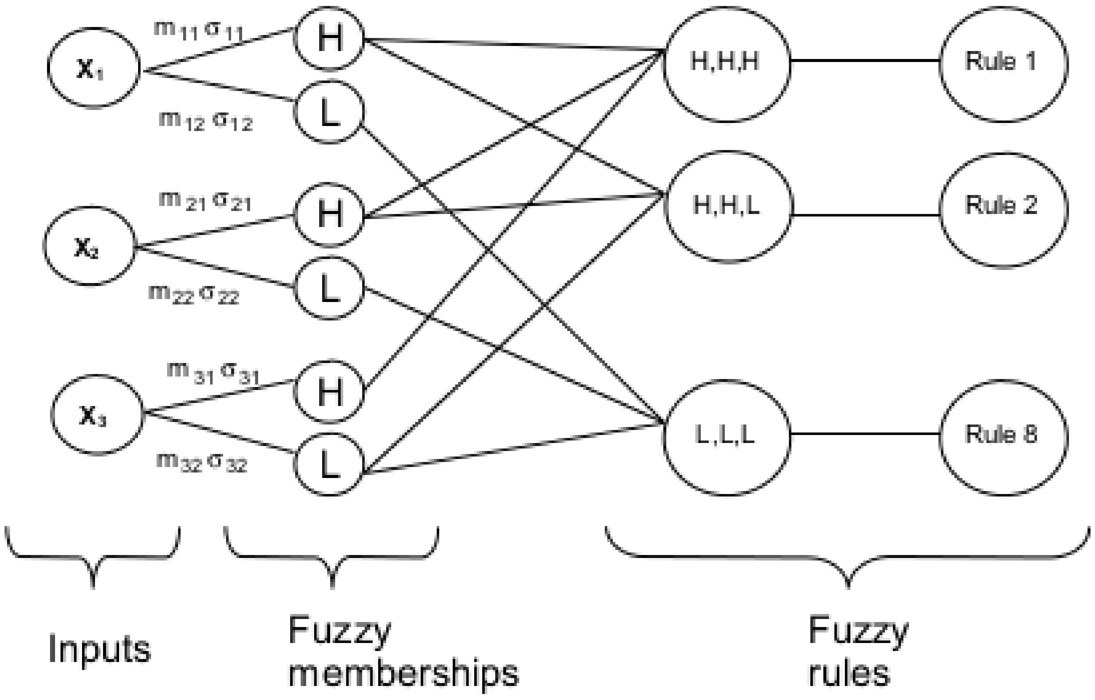

Fig. 2. Fuzzy rule identification and clustering process.

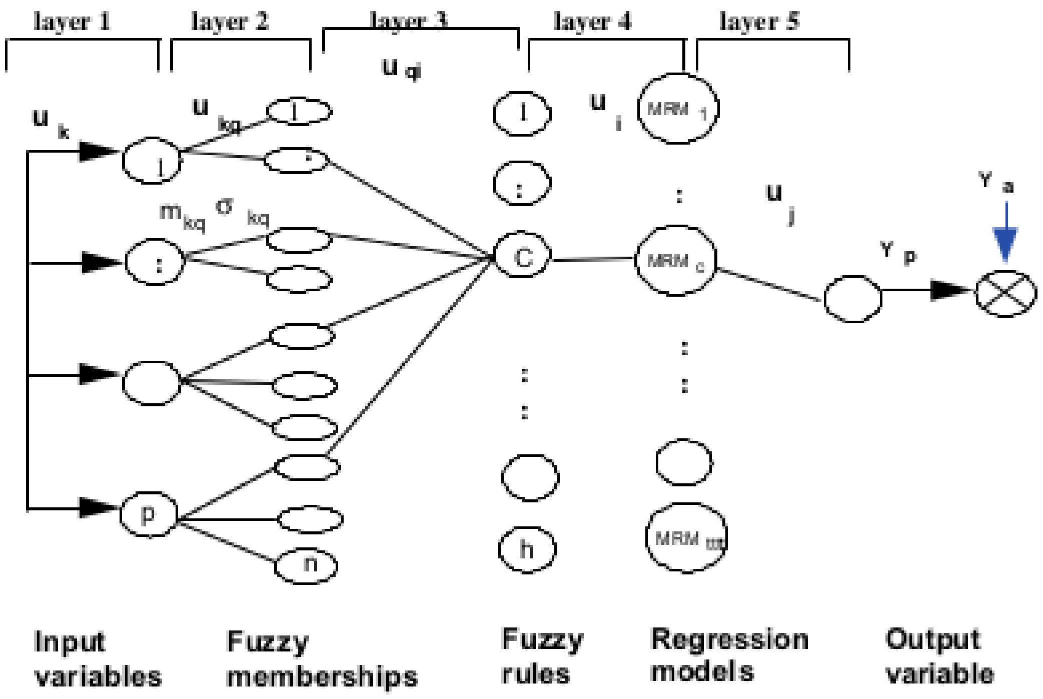

Fig. 3. Fuzzy rule-based regression modeling process.

prediction does not match any of the existing rules in the rule base, the system will choose[a] as replacement from the rule base a rule that is closest[b] to the rule identified 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 to ensure that the training is complete.

The FRM network can be retrained whenever new data become available. Retraining involves repeating stages 3.1 to 3.4 to reconstruct membership functions, identify new fuzzy rules and new clusters, if any, and update regression models. In general, the more the FRM network is retrained, the more accurate it will be.

Three commonly used measures, namely Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE) and R-squared Values (R^2)[93] were used to evaluate the prediction accuracy in this work.

4. Case Study

The efficacy of the proposed FRM is tested through a case study — namely to prediction of the remaining useful life of a ball nose milling cutter during a dry machining process of hardened stainless steel (HRC52). Tool condition monitoring and failure prediction are important factor in automated machining processes. Undetected or pre-mature tool failures often leads to costly scrap or rework arising for damaged surface finishing and loss of dimensional accuracy or possible damage to the work piece and machine.[42] The types of workpiece material; cutting speed, feed rate, depth of cuts used; cutting tool materials and nano-composite coatings used; together with the age and rigidity of the machine tool and fixture will collectively affect the remaining useful life of the tool in question. Similar to our previous works,[94] the aim of this study is to predict the performance of a 6mm ball nose milling cutter during the machining of hardened tool steel material, which is often used in the mould-making industry where surface integrity is of a paramount importance.

4.1. *Experiment set-up and features extraction*

Presently, most of the CNC machine centers are not equipped with highly reliable tool condition monitoring system since the relationships between the work piece material, cutting parameter, tool characteristics and geometry has yet to be clearly established. In our experimental setup, we chose to use

[a] In the event of a tie, i.e. two or more possible replacements exist, one is selected at random.

[b] The closeness between two rules is measured in terms of the number of common membership function nodes that the rules involve.

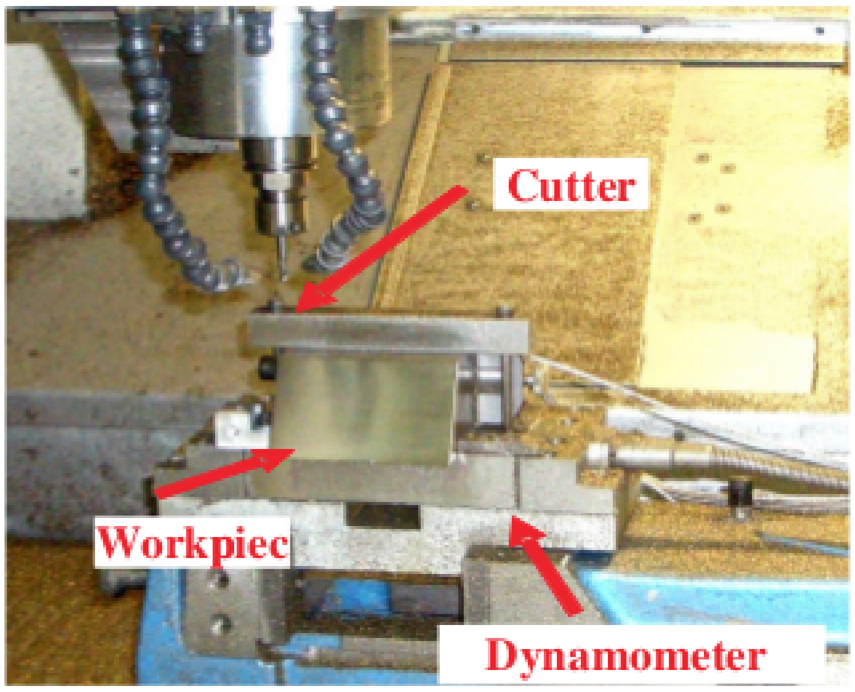

Fig. 4. Tool condition prognostic monitoring in a milling machine.

the cutting force as our input signal for establishing the FRM models for tool life prediction since it is highly sensitive and can be measured with fairly good accuracy[95] in order to provide relatively clear indication of tool wear and degradation. The experimental set up for the monitoring of cutting tool condition is shown in Fig. 4.

A high speed CNC milling machine (Röders Tech RFM760) with spindle speed up to 42,000 rpm was selected for the experiment. The tool used in the experiment is 6 mm 2-flutes ball nose tungsten carbide cutter while the material is hardened stainless steel (HRC52) machined under a dry cutting condition. The surface was machined to have a slope with 60° to accommodate the 2-flute ball nose cutter. The spindle speed chosen was 23,600 RPM, feed rate 4,710 mm/min, depth of cut 0.2 mm. The cutting force along the x-, y- and z-directions of the machining tool was measured by a Kistler quartz 3-component platform dynamometer in the form of charges, and converted to voltages by the Kistler charge amplifier. The voltage signal was captured by a PCI 1200 board with assembling rate at 1,500 Hz per channel and directly streamed to the hard disk of a computer. The flank wear of each individual tooth of the cutting tool are measured with a Leica microscope.

Sixteen main force features was captured and summarized in Table 1. Among the sixteen features, four of them, {fm, fa, Fa, std}, have been identified to have the most significant influence to tool wear.[96] The feature data and measured tool wear are then stored into the database.

Table 1. Extracted force features.

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

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

4.2. *SOM learning and rule generation*

A total of 52,800 sets of data were captured - half of the data are used principally for rule training and the remaining are used for testing. As mentioned in Sec. 4.1, four input variables which are thought to have an influence over the tool wear (Y) are selected. They are maximum force level ($X_1 : \boldsymbol{f}_m$), total amplitude of cutting Force ($X_2 : \boldsymbol{f}_a$), average force ($X_3 : \boldsymbol{F}_a$) and standard deviation of force ($X_4 : \boldsymbol{std}$). The selection method used is based on an optimum feature subset selection with a modified wrapper-based multi-criteria approach using Genetic Algorithms.[96]

The FRM starts its self learning of fuzzy membership parameters (mean and variance) with SOM. The data samples were first normalized to the range of $\{0, 1\}$. The number of membership function nodes is a set of each input variable established through the trials and errors mining of the distribution patterns from the data samples. There are 10 nodes (very low, low, ... medium, ... high, very high) for $\boldsymbol{x}_1$, 12 nodes for $\boldsymbol{x}_2$, 8 nodes for $\boldsymbol{x}_3$ and 12 nodes for $\boldsymbol{x}_4$. Thus, the maximum number of possible fuzzy rules is 11,520 which is derived from 10 × 12 × 8 × 12. However, the rule base stores only 83 rules which have been trained and are represented as clusters, as shown in Table 2. Figure 5 shows the membership distributions of X_1 before and after learning. With the adjusted means and variances, the data sets are then clustered according to their similarity. The x-axis represents the normalization of the selected feature being normalized to $\{0, 1\}$.

4.3. *Fuzzy-rule-based multiple regression modelling*

Since the data sets within one fuzzy rule or one cluster have similarity characters, their relationships are most likely to be linear. MRM is applied with the data sets to build the correlation among the inputs and output within each rule.

The relationships between the input variables, fuzzy memberships, fuzzy rules, multi-regression models and output are shown in Fig. 6. Given one data set of x1, x2, x3 at the first layer, it is found that x1 belongs to "High", x2 is "High" and x3 is "Low" after the FRM is fuzzified and clustered using the SOM. The data set is further reorganized into fuzzy rule, #2 which linked with the data cluster "High, High, Low". A regression model named MRM2 will be linked directly with rule #2 if the frequency of occurrence within rule #2 exceeds more than ten sets. In our case study, a total of 28,400 data sets are trained. A total of 65 data sets have been associated with fuzzy rule #11 as shown in Table 2. A regression model is built up with the 65 data sets as shown in Eq. (10).

$$\begin{aligned} Y_{p(\text{rule }2)} &= -0.218 - 0.11 f_{m(t-1)} + 0.105 f_{a(t-1)} \\ &\quad + 0.157 F_{a(t-1)} + 0.118 f_{\text{std}(t-1)} \qquad (10) \end{aligned}$$

The T-test values for the coefficients are respectively

$$(-237.73)\ (-1.98)\ (1.868)\ (60.19)\ (29.52)$$

The F-test value for the overall significance is 218,954.

$$R^2 = 0.9759$$

Table 2. The learned fuzzy rules and associated data sets.

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
1133	36	35	3367	19	21
1135	16	17	3376	26	29
1141	12	17	3377	13	7
1142	258	256	4465	52	54
1143	155	157	4466	29	28
1153	39	39	5544	24	22
1221	19	19	5554	43	45
1242	41	47	5555	29	29
1243	44	37	5565	49	48
1244	16	16	5566	22	22
1253	40	45	5575	14	16
1254	17	16	5576	84	88
2221	50	51	5577	42	44
2222	17	16	5578	12	10
2231	11	8	6666	35	31
2232	168	168	6667	12	11
2233	21	16	6676	25	25
2234	11	11	6677	22	22
2242	25	27	6687	55	53
2243	119	122	6688	18	21
2244	79	82	7787	18	19
2245	19	21	7788	12	11
2253	169	170	7789	37	37
2254	127	123	8811	12	14
2255	34	33	8814	12	12
2263	21	22	8841	13	11
2264	36	32	8894	19	18
2265	21	23	8899	12	11
			9911	80	78

This model is always accessible once a data set is associated with rule # 2. The output Y_p from the FRM is compared with its real output tool wear Y_a. If the R^2 is not accepted, the FRM will restart from the fuzzy membership definition step, as shown in Fig. 7. Otherwise, fuzzy-rule-based regression

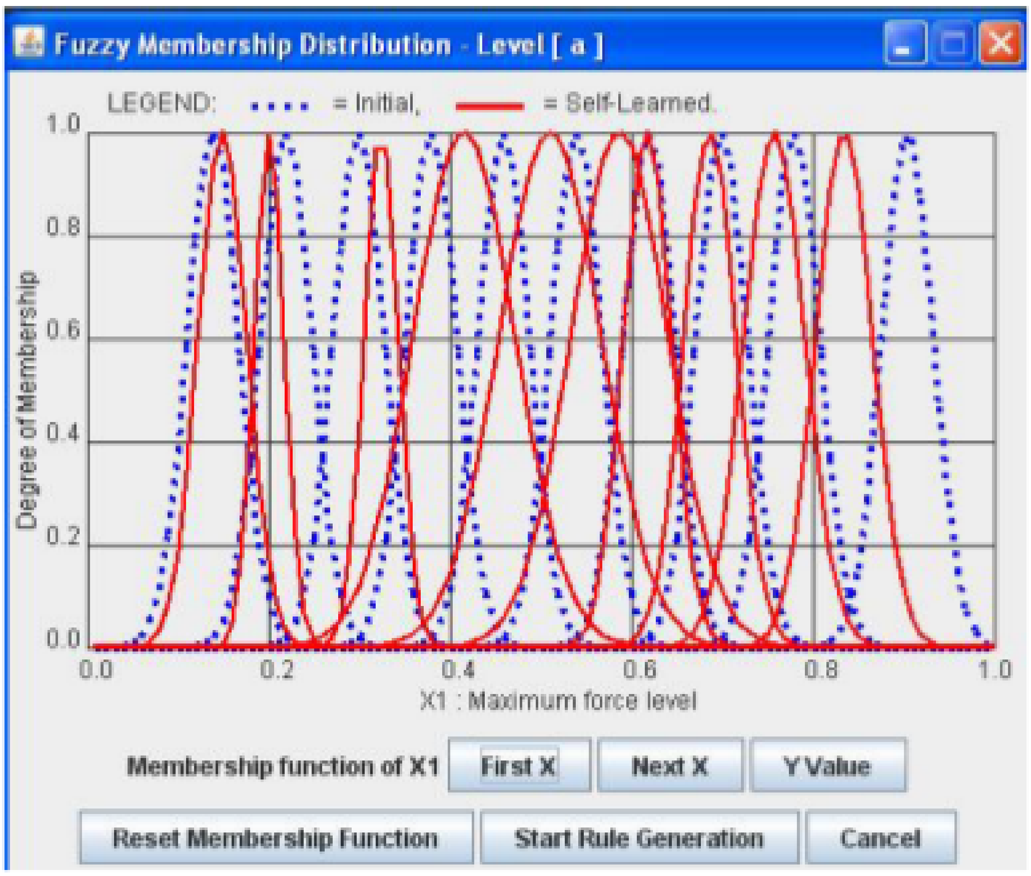
Fuzzy Membership Distribution - Level [a]
LEGEND: •••• = Initial, ▬▬ = Self-Learned.
Degree of Membership
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
X1 : Maximum force level
Membership function of X1
First X
Next X
Y Value
Reset Membership Function
Start Rule Generation
Cancel

Fig. 5. Before-after self-learned fuzzy membership distributions.

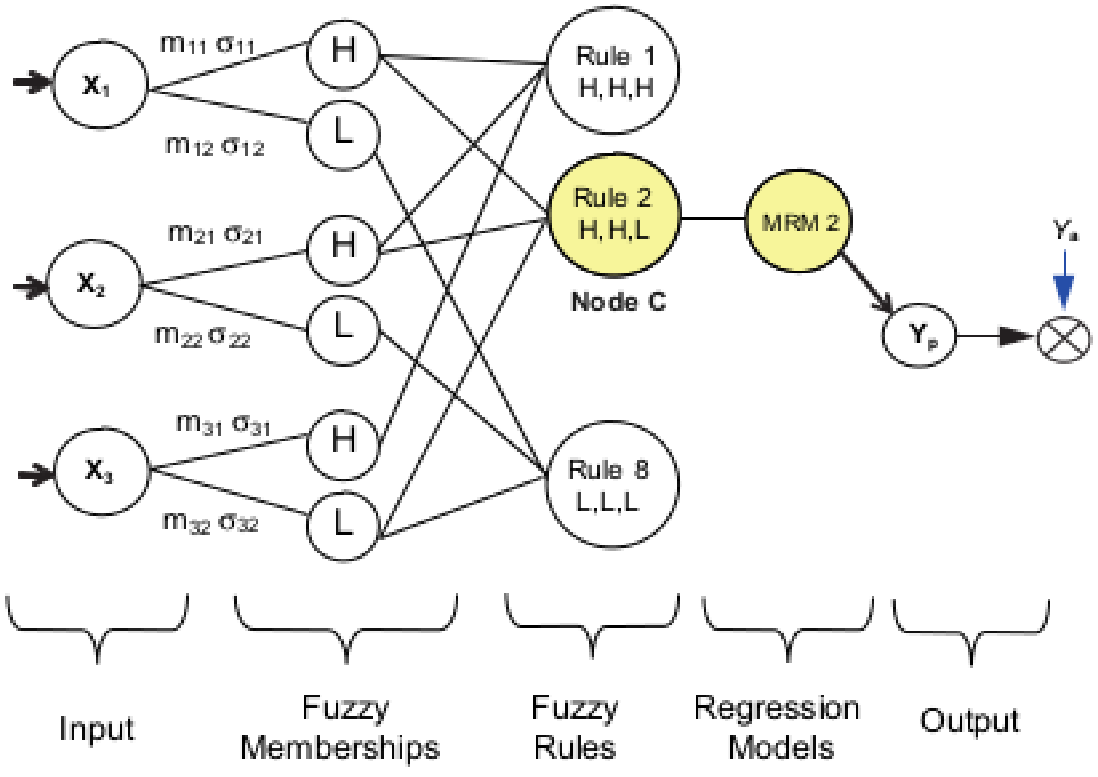

Fig. 6. Example illustration of fuzzy rule-based regression modeling.

modelling process will be terminated. The overall process of determining FRM fuzzy membership, data clustering, rule generation and input and output correlation modeling are shown in Fig. 7.

4.4. *Prediction of tool life span and re-training*

Once the clustering and modeling are completed, the FRM network is ready for prediction. For a given input data set, if the fuzzy rule identified during prediction does not match any of the existing rules in the rule base, the system will choose as replacement from the rule base a rule that is closest to the rule identified, and will proceed with the prediction process as shown in Fig. 8. This process will inevitability introduces errors and will affect the accuracy of prediction. It is therefore important that a sufficiently large pool of data samples must be used to ensure vigorousness with the training process and to ensure higher accuracy.

The FRM network can be retrained whenever a new data become available. Retraining involves repeating stages 3.1 to 3.4 (see Sec. 3) to reconstruct membership functions, identify new fuzzy rules, if any, and update regression models. In general, the more the FRM network is retrained, the higher will be its accuracy. The prediction of tool wear based on an untrained dataset using previous trained information is shown in Fig. 8.

After the FRM network is trained, test data are fed to the trained FRM to obtain the forecasted values Ft. Error measures for the 28,240 sets of test data are found to be: MSE $= 1.743 \times 10^{-5}$, MAPE $= 0.01$ and $R^2 = 0.9984$.

Through the FRM clustering, modeling and prediction test, it was noted that the amount of training time required to carry out the experiment is dependent on the number of fuzzy membership nodes assigned to each of the features. Therefore, it could be concluded that the higher the level of accuracy required, a larger number of membership items are required — which in turn will degrade the convergence speed. However, it must be also noted that high accuracy is not assured even when more fuzzy

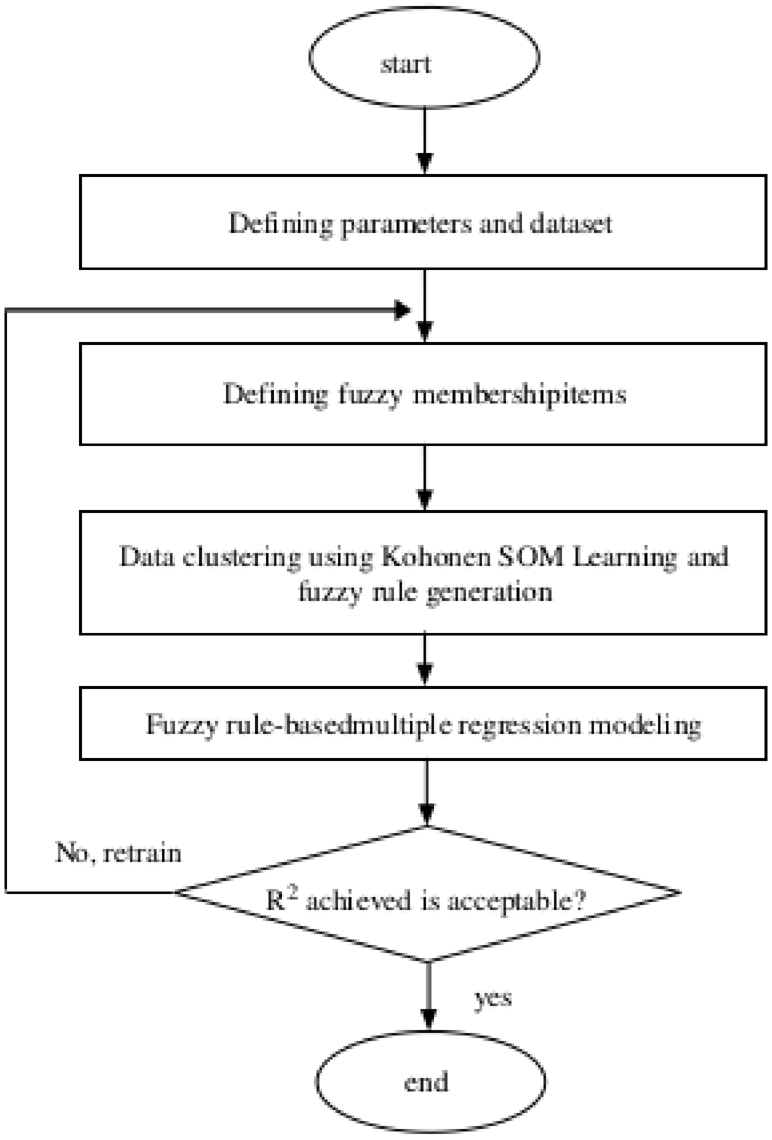
start
Defining parameters and dataset
Defining fuzzy membershipitems
Data clustering using Kohonen SOM Learning and fuzzy rule generation
Fuzzy rule-basedmultiple regression modeling
No, retrain
R² achieved is acceptable?
yes
end

Fig. 7. Flowchart of FRM clustering and modeling process.

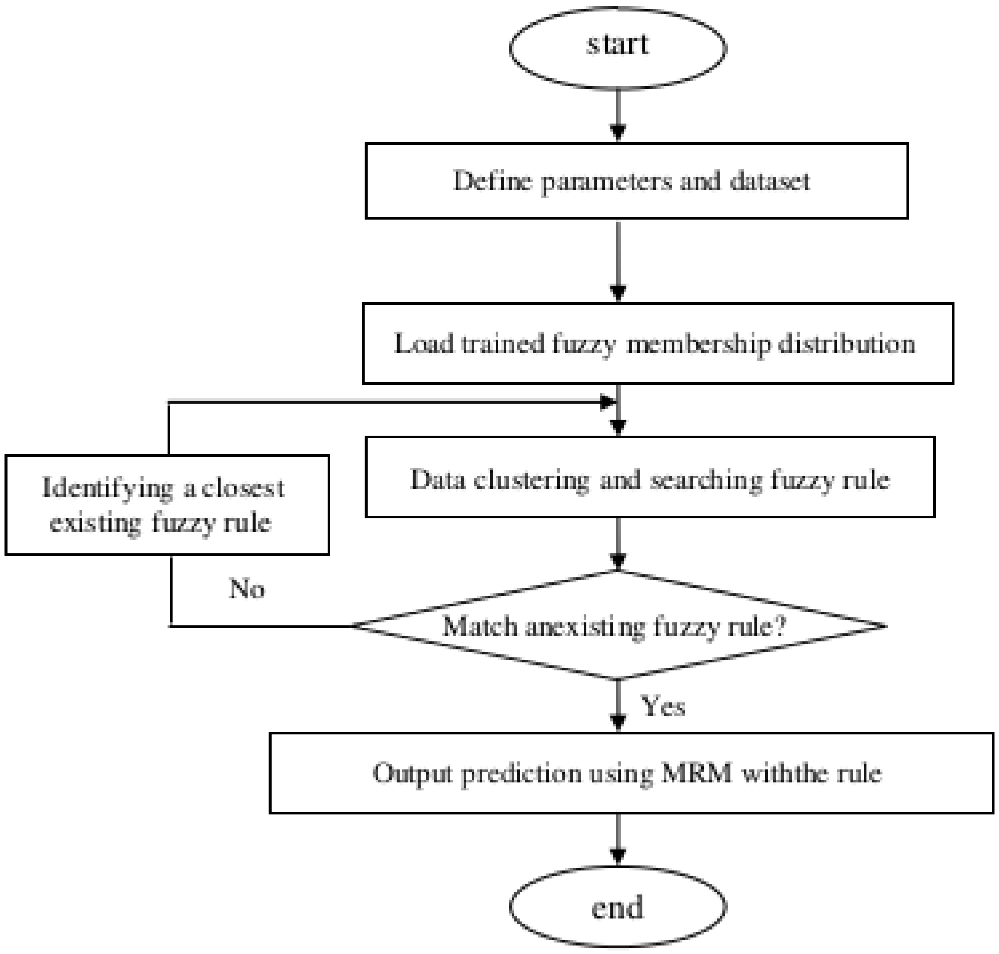
start
Define parameters and dataset
Load trained fuzzy membership distribution
Identifying a closest existing fuzzy rule
Data clustering and searching fuzzy rule
No
Match anexisting fuzzy rule?
Yes
Output prediction using MRM withthe rule
end

Fig. 8. Flowchart of prediction process.

Table 3. Prediction performance of FRM.

Error measures	FRM
MSE	1.743×10^{-5}
MAPE	0.010
R^2	0.9984

membership items are defined. As such, the determination of the optimum the number of fuzzy membership items remain an unresolved problem.

4.5. *Comparison with conventional neural networks*

The same dataset was used for tool performance prediction with BPNN and RBFN and conventional MRM.

Table 4 summarizes the parametric set-ups and training performance of both neural network models, BPNN and RBFN. The training results show that the RBFN performs than BPNN since lesser iterations are required.

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 rule used for the BPNN model is Delta and the transfer function used is Sigmoid. For the RBFN model, the Learning Rule used is Norm-Cum-Delta and the transfer function used is Gaussian.

Table 4. Parametric set-ups and training performance of FRM, BPNN and RBFN.

Parameters	FRM	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}	0.00556	0.00132

The relative performance of the MRM, BPNN, RBFN and FNN is shown in Table 5 and Fig. 9. The

Table 5. Prediction performances of MRM, BPNN, RBFN and FNN.

Error measurement	MRM	BPNN	RBFN	FRM
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	0.9984

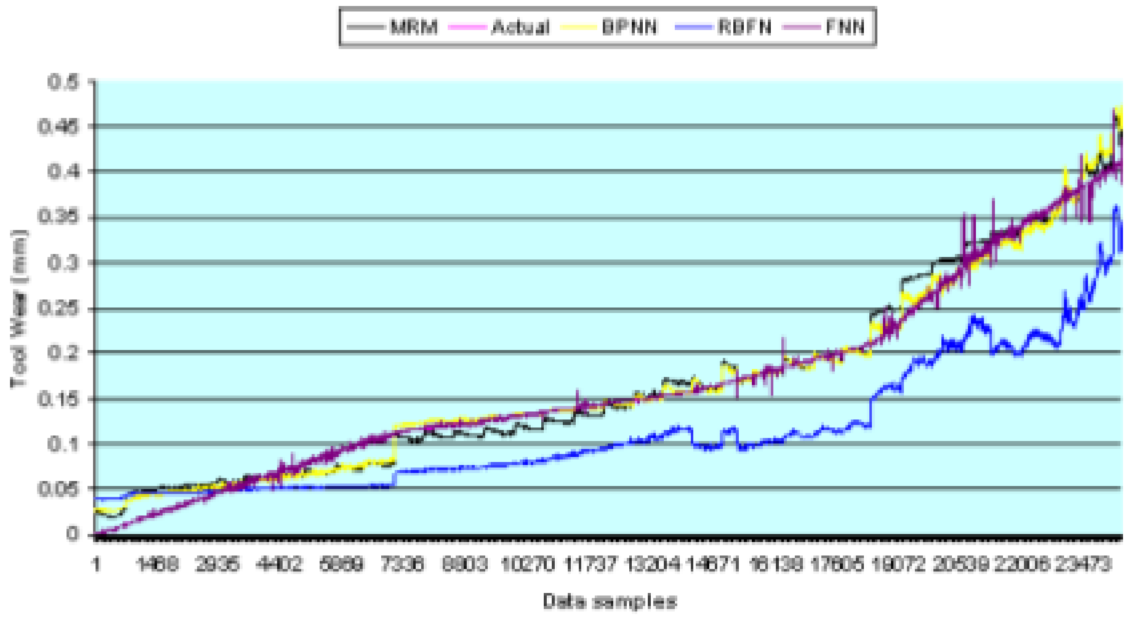

Fig. 9. Comparisons of prediction performance (MRM, BPNN, RBFN and FRM).

results show that amongst the 4 models evaluated, the FRM is most superior — it has the smallest MSE and MAPE and comparable R2 when compared with the actual data. The RBFN performs poorly since it has the highest error measurement compared with the other 3 models.

In comparison with conventional neural network, it can be observed that the FRM outperforms the others models in the prediction of ball nose cutter's performance and degradation during the machining of hardened mould steel material. However, the main constraints of the FRM lies in the fact that the optimal number of fuzzy membership items cannot be defined, automatically.

5. Conclusion

A FRM algorithm is designed and developed for the prediction of cutting tool performance and degradation. The FRM is developed based on a multilayered fuzzy-rule-based hybrid system with Multiple Regression Models (MRM) embedded into a fuzzy logic inference engine that employs Self Organizing Maps (SOM) for clustering. The FRM reduces a complex non-linear problem into a simplified linear format so as to enhance the accuracy of prediction and learning speed. A case study for predicting the performance and degradation of ball nose milling cutters is presented to demonstrate the viability of the FRM for the high precision machining industry. A comparative study in terms of prediction accuracy using 4 predictive models was made using the same set of experimental data within the case study. It is shown that the FRM is far superior to conventional MRM, BPNN and RBFN in terms of prediction accuracy and learning speed. However, the limitation of FRM lies in the fact that the optimal number of fuzzy membership items cannot be automatically defined. This could be a topic for further investigation.

Acknowledgment

This work was partly supported by the Polish Ministry of Science and Higher Education and Agency for Science, Technology and Research (A*STAR) Singapore, Science and Engineering Research Council (SERC, Polish-Singapore Research Project 2008-2011). The authors would like to thank all people who have helped the project in one way or another, especially to Sheng Huang, Lianyin Zhai, Amin J. Torabi and San Linn.

References

1. A. Panakkat and H. Adeli, Recurrent neural network for approximate earthquake time and location prediction using multiple seismicity indicators, *Computer-Aided Civil and Infrastructure Engineering* **24**(4) (2009) 280–292.
2. D. P. Mandic, P. Vayanos, M. Chen and S. L. Goh, Online detection of the modality of complex valued real world signals, *International Journal of Neural Systems* **18**(2) (2008) 67–74.
3. H. Ni and H. Yin, Self-organizing mixture autoregressive model for non-stationary time series prediction, *International Journal of Neural Systems* **18**(6) (2008).
4. J. Menke and T. Martinez, Improving supervised learning by adapting the problem to the learner, *International Journal of Neural Systems* **19**(1) (2009) 1–9.
5. H. Adeli and H. S. Park, A neural dynamics model for structural optimization — Theory, *Computers and Structures* **57**(3) (1995a) 383–390.
6. H. Adeli and H. S. Park, Optimization of space structures by neural dynamics, *Neural Networks* **8**(5) (1995b) 769–781.
7. H. Adeli and H. S. Park, Fully automated design of superhighrise building structure by a hybrid AI model on a massively parallel machine, *AI Magazine* **17**(3) (1996) 87–93.
8. H. S. Park and H. Adeli, A neural dynamics model for structural optimization — Application to plastic design of structures, *Computers and Structures* **57**(3) (1995) 391–399.
9. H. S. Park and H. Adeli, Distributed neural dynamics algorithms for optimization of large steel structures, *Journal of Structural Engineering, ASCE* **123**(7) (1997) 880–888.
10. H. Adeli and A. Karim, Neural network model for optimization of cold-formed steel beams, *Journal of Structural Engineering, ASCE* **123**(11) (1997) 1535–1543.
11. A. Karim and H. Adeli, CONSCOM: An OO construction scheduling and change management system, *Journal of Construction Engineering and Management* **125**(5) (1999) 368–376.
12. A. B. Senouci and H. Adeli, Resource scheduling using neural dynamics model of Adeli and Park, *Journal of Construction Engineering and Management, ASCE* **127**(1) (2001) 28–34.
13. A. R. Tashakori and H. Adeli, Optimum design of cold-formed steel space structures using neural dynamic model, *Journal of Constructional Steel Research* **58**(12) (2002) 1545–1566.

14. G. G. Rigatos and S. G. Tzafestas, Neurodynamics and attractors in quantum associative memories, **14**(3) (2007) 224–242.
15. X. Jiang and H. Adeli, Dynamic wavelet neural network for nonlinear identification of highrise buildings, *Computer-Aided Civil and Infrastructure Engineering* **20**(5) (2005a) 316–330.
16. X. Jiang and H. Adeli, Dynamic wavelet neural network model for traffic flow forecasting, *Journal of Transportation Engineering, ASCE* **131**(10) (2005b) 771–779.
17. X. Jiang and H. Adeli, Dynamic fuzzy wavelet neuroemulator for nonlinear control of irregular highrise building structures, *International Journal for Numerical Methods in Engineering* **74**(7) (2008a) 1045–1066.
18. X. Jiang and H. Adeli, Neuro-genetic algorithm for nonlinear active control of highrise buildings, *International Journal for Numerical Methods in Engineering* **75**(8) (2008b) 770–786.
19. H. Adeli, S. Ghosh-Dastidar and N. Dadmehr, A spatio-temporal wavelet-chaos methodology for EEG-based diagnosis of Alzheimer's disease, *Neuroscience Letters* **444**(2) (2008) 190–194.
20. A. Pande and M. Abdel-Aty, A computing approach using probabilistic neural networks for instantaneous appraisal of rear-end crash risk, *Computer-Aided Civil and Infrastructure Engineering* **23**(7) (2008) 549–559.
21. H. Adeli and A. Panakkat, A probabilistic neural network for earthquake magnitude prediction, *Neural Networks* **22** (2009) 1018–1024.
22. M. Ahmadlou and H. Adeli, Enhanced probabilistic neural network with local decision circles: A robust classifier, *Integrated Computer-Aided Engineering* **17**(3) (2010).
23. D. P. Mandic, P. Vayanos, M. Chen and S. L. Goh, Online detection of the modality of complex valued real world signals, *International Journal of Neural Systems* **18**(2) (2008) 67–74.
24. S. Buchholz and N. L. Bihan, Polarized signal classification by complex and quaternionic multilayer perceptrons, *International Journal of Neural Systems* **18**(2) (2008) 75–85.
25. S. Fiori, Learning by criterion optimization on a unitary unimodular matrix group, *International Journal of Neural Systems* **18**(2) (2008) 87–103.
26. D. Vigliano, R. Parisi and A. Uncini, Flexible nonlinear blind signal separation in the complex domain, *International Journal of Neural Systems* **18**(2) (2008) 105–122.
27. T. Nitta, The uniqueness theorem for complex-valued neural networks with threshold parameters and the redundancy of the parameters, *International Journal of Neural Systems* **18**(2) (2008) 123–134.
28. T. Isokawa, H. Nishimura, N. Kamiura and N. Matsui, Associative memory in quaternionic hopfield neural network, *International Journal of Neural Systems* **18**(2) (2008) 135–45.
29. M. Kobayashi, Pseudo-relaxation learning algorithm for complex valued associative memory, *International Journal of Neural Systems* **18**(2) (2008) 147–156.
30. V. S. Chakravarthy, Gupte, N. S. Yogesh and A. Salhotra, Chaotic synchronization using a network of neural oscillators, *International Journal of Neural Systems* **18**(2) (2008) 157–164.
31. V. S. H. Rao and G. R. Murthy, Global dynamics of a class of complex valued neural networks, *International Journal of Neural Systems* **18**(2) (2008) 165–171.
32. S. Kawata and A. Hirose, Frequency-multiplexing ability of complex-valued Hebbian learning in logic gates, *International Journal of Neural Systems* **18**(2) (2008) 173–184.
33. T. D. Jorgensen, B. P. Haynes and C. C. F. Norlund, Pruning artificial neural networks using neural complexity measures, *International Journal of Neural Systems* **18**(5) (2008) 389–403.
34. S. Ghosh-Dastidar and H. Adeli, Improved spiking neural networks for EEG classification and epilepsy and seizure detection, *Integrated Computer-Aided Engineering* **14**(3) (2007) 187–212.
35. J. Iglesias and A. E. P. Villa, Emergence of preferred firing sequences in large spiking neural networks during simulated neuronal development, *International Journal of Neural Systems* **18**(4) (2008) 267–277.
36. J. L. Rossello, V. Canals, A. Morro and J. Verd, Chaos-based mixed signal implementation of spiking neurons, *International Journal of Neural Systems* **19**(6) (2009) 465–471.
37. H. C. Lian, No-reference video quality measurement with support vector regression, *International Journal of Neural Systems* **19**(6) (2009) 457–464.
38. B. Cyganek, Colour image segmentation with support vector machines: Applications to road signs detection, *International Journal of Neural Systems* **18**(4) (2008) 339–345.
39. G. Vachtsevanos, P. Wang and N. Khiripet, Prognostication: Algorithms and performance assessment methodologies, ATP Fall National Meeting Condition-based Maintenance Workshop, San Jose, California (November 15–17, 1999).
40. J. Lee and J. Ni, Infotronics agent for tether-free prognostics, *Proceeding of AAAI Spring Symposium on Information Refinement and Revision for Decision Making: Modeling for Diagnostics, Prognostics, and Prediction*, Stanford Univ., Palo Alto, CA, March 25–27 (2002).
41. N. Propes, S. Lee, G. Zhang, I. Barlas, Y. Zhao, G. Vachtsevanos, A. Thakker and T. Galie, A real-time

architecture for prognostic enhancements to diagnostic systems, MARCON, Knoxville, Tennessee (May, 2002).

42. P. T. Huang and J. C. Chen, Neural network-based tool breakage monitoring system for end milling operations, *Journal of Industrial Technology* (2000).
43. L. Wang, M. G. Mehrabi and E. K. Jr, Tool wear monitoring in reconfigurable machining systems through wavelet analysis, *Transactions of NAMRI* (2001) 399–406.
44. 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).
45. S. Orhan, A. O. Er, N. Camuşcu and E. Aslan, Tool wear evaluation by vibration analysis during end milling of AISI D3 cold work tool steel with 35 HRC hardness, *NDT & International* **40**(2) (2007) 121–126.
46. E. Haddadi, M. R. Shabghard and M. M. Ettefagh, Effect of different tool edge conditions on wear detection by vibration spectrum analysis in turning operation, *Journal of Applied Science* **8**(21) (2008) 3879–3886.
47. W. J. Endres, J. W. Sutherland, R. E. DeVor and S. G. Kapoor, A dynamic model of the cutting force system in the turning process, *Monitoring and Control for Manufacturing Process*, ASME PED. **44** (1990) 193–212.
48. Y. Matsumoto, N. Tjinag and B. Foote, Tool wear monitoring using a linear extrapolation, *Sensors and Control for Manufacturing*, ASME-PED **33** (1988) 83–88.
49. S. Y. Liang and D. Dornfeld, Tool wear detection using time series analysis of acoustic emission, *ASME Journal of Engineering for Industry* **l11** (1989) 199–205.
50. R. Isermann, M. Ayoubi, H. Konrad and T. Reib, Model based detection of tool wear and breakage for machine tools, *Proceedings of the IEEE International Conference System Manufacturing Cybernetics* **3** (1993) 72–77.
51. A. G. Vallejo, Jr., J. A. Nolazco-Flores and R. Morales-Men'endez, Tool-wear monitoring based on continuous hidden Markov models, M. Lazo and A. Sanfeliu (eds.): CIARP 2005, LNCS 3773, Springer-Verlag Berlin Heidelberg (2005) 880–890.
52. S. Dey and J. A. Stori, A Bayesian network approach to root cause diagnosis of process, *International Journal of Machine Tools & Manufacture* **45** (2005) 75–91.
53. H. Saglam and A. Unuvar, Tool condition monitoring in milling based on cutting forces by a neural network, *International Journal of Production Research* **41**(7) (2003) 1519–1532.
54. X. Li, A. Djordjevich and P. K. Venuvinod, Current-sensor-based feed cutting force intelligent estimation and tool wear condition monitoring, *IEEE Transactions on Industrial Electronics* **47**(3) (2000) 697–702.
55. P. K. Simpson, Fuzzy min-max neural networks — Part 1: Classification, *IEEE Transactions on Neural Networks* **3**(5) (1992) 776–786.
56. B. Y. Lee, H. S. Liu and Y. S. Tarng, Modeling and optimization of drilling process, *Journal of Material Processing Technology* **74** (1998) 149–157.
57. K. Perusich, Using fuzzy cognitive maps to identify multiple causes in troubleshooting systems, *Integrated Computer-Aided Engineering* **15**(2) (2008) 197–206.
58. J. R. Villar, E. de la Cal and J. Sedano, A fuzzy logic based efficient energy saving approach for domestic heating systems, *Integrated Computer-Aided Engineering* **16**(2) (2009) 151–163.
59. A. Stathopoulos, L. Dimitriou and T. Tsekeris, Fuzzy modeling approach for combined forecasting of urban traffic flow, *Computer-Aided Civil and Infrastructure Engineering* **23**(7) (2008) 521–535.
60. X. L. Jin and H. Doloi, Modelling risk allocation decision-making in PPP projects using fuzzy logic, *Computer-Aided Civil and Infrastructure Engineering* **24**(7) (2009) 509–524.
61. Mo A. Elbestawi, M. Dumitrescu and E.-G. Ng, Tool condition monitoring in machining, in *Condition Monitoring and Control for Intelligent Manufacturing*, Springer-Verlag (London) Limited (2006).
62. J. M. Zurada, *Introduction to Artificial Neural System* (West Publishing Company, USA, 1992).
63. T. Takagi and M. Sugeno, Fuzzy identification of systems and its application to modeling and control, *IEEE Trans. Syst. Man. Cyber.* **15**(11) (1985) 116–132.
64. R. Babuska, *Fuzzy Modeling for Control* (Kluwer Academic Publishers, Boston, 1998).
65. H. V. Poor and O. Hadjiliadis, *Quickest Detection* (Cambridge University Press, Cambridge, 2009).
66. D. C. Montgomery, *Introduction to Statistical Quality Control*, 4th edn. (Wiley, New York, 2001).
67. B. E. Brodsky and B. S. Darkhovsky, *Nonparametric Methods in Change-Point Problems*, Kluwer, Dordrecht (2000).
68. S. Kritchman and B. Nadler, Non-parametric detection of the number of signals: Hypothesis testing and random matrix theory, *IEEE Transactions on Signal Processing* **57**(10) (2009) 3930–3941.
69. H. Chen, P. K. Varshney, S. Kay and J. H. Michels, Noise enhanced nonparametric detection, *IEEE Transactions on Information Theory*, ISSN:0018-9448, IEEE Press Piscataway, NJ, USA, **55**(2) (2009) 499–506.

70. H. Adeli and S. L. Hung, An adaptive conjugate gradient learning algorithm for effective training of multilayer neural networks, *Applied Mathematics and Computation* **62**(1) (1994) 81–102.

71. A. Samant and H. Adeli, Enhancing neural network incident detection algorithms using wavelets, *Computer-Aided Civil and Infrastructure Engineering* **16**(4) (2001) 239–245.

72. A. Karim and H. Adeli, Comparison of the fuzzy — wavelet RBFNN freeway incident detection model with the California algorithm, *Journal of Transportation Engineering* **128**(1) (2002) 21–30.

73. H. Adeli and X. Jiang, Neuro-fuzzy logic model for freeway work zone capacity estimation, *Journal of Transportation Engineering* **129**(5) (2003) 484–493.

74. H. Adeli and X. Jiang, Dynamic fuzzy wavelet neural network model for structural system identification, *Journal of Structural Engineering* **132**(1) (2006) 102–111.

75. A. M. A. Haidar, A. Mohamed, M. Al-Dabbagh, A. Aini Hussain and M. Masoum, An intelligent load shedding scheme using neural networks and neuro-fuzzy, *International Journal of Neural Systems* **19**(6) (2009) 473–479.

76. A. Fatehi and K. Abe, Flexible structure multiple modeling using irregular self-organizing maps neural network, *International Journal of Neural Systems* **18**(3) (2008) 233–256.

77. S. L. Hung and H. Adeli, Parallel backpropagation learning algorithms on cray Y-MP8/864 supercomputer, *Neurocomputing* **5**(6) (1993) 287–302.

78. S. L. Hung and H. Adeli, Object-oriented back propagation and its application to structural design, *Neurocomputing* **6**(1) (1994) 45–55.

79. H. Adeli and A. Karim, Fuzzy-wavelet RBFNN model for freeway incident detection, *Journal of Transportation Engineering* **126**(6) (2000) 464–471.

80. A. Karim and H. Adeli, Comparison of the fuzzy — wavelet RBFNN freeway incident detection model with the california algorithm, *Journal of Transportation Engineering* **128**(1) (2002) 21–30.

81. A. Karim and H. Adeli, Radial basis function neural network for work zone capacity and queue estimation, *Journal of Transportation Engineering* **129**(5) (2003) 494–503.

82. R. V. Mayorga and J. Carrera, A radial basis function network approach for the computational of inverse continuous time variant functions, *International Journal of Neural Systems* **17**(3) (2007) 149–160.

83. H. Liu, X. Wang and W. Qiang, A fast method for implicit surface reconstruction based on radial basis functions network from 3D scattered points, *International Journal of Neural Systems* **17**(6) (2007) 459–465.

84. S. Ghosh-Dastidar, H. Adeli and N. Dadmehr, Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection, *IEEE Transactions on Biomedical Engineering* **55**(2) (2008) 512–518.

85. W. Pedrycz, R. Rai and J. Zurada, Experience-consistent modeling for radial basis function neural networks, *International Journal of Neural Systems* **18**(4) (2008) 279–292.

86. P. Anand, B. V. N. Siva Prasad and C. Venkateswarlu, Modeling and optimization of a pharmaceutical formulation system using radial basis function network, *International Journal of Neural Systems* **19**(2) (2009) 127–136.

87. R. Savitha, S. Suresh and N. Sundararajan, A fully complex-valued radial basis function network and its learning algorithm, *International Journal of Neural Systems* **19**(4) (2009) 253–267.

88. X. Li, C. L. Ang and R. Gay, An intelligent business forecaster for strategic business planning, *Journal of Forecasting* **18** (1999) 181–204.

89. L. Rutkowski, *Computational Intelligence: Method and Techniques* (Springer, 2008).

90. T. Kohonen, *Self-Organization and Associative Memory* (Springer-Verlag, Berlin, 1988).

91. E. P. Carden and J. M. W. Brownjohn, Fuzzy clustering of stability diagrams for vibration-based structural health monitoring, *Computer-Aided Civil and Infrastructure Engineering* **23**(5) (2008) 360–372.

92. N. Gujarati Damodar, *Basic Econometrics*, 3rd edn. (McGraw-Hill International Editions, 1995).

93. S. Makridakes, S. C. Wheelwright and R. J. Hyndman, *Forecasting Methods and Applications*, 3rd edn. (John Wiley & Sons Inc., New York, 1998).

94. X. Li, J. H. Zhou, H. Zeng, Y. S. Wong and G. S. Hong, An intelligent predictive engine for milling machine prognostic monitoring, *CD Proceeding of the 4th IEEE International Conference on Industrial Informatics*, Singapore (2006) 1075–1080.

95. Y. Altintas *et al.*, In process detection of tool failure in milling using cutting force models, *J. Engng Ind.* **111** (1989) 149–157.

96. 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*), Gold Coast, Australia (2006) 1114–1123.