Scientific Research and Essays Vol. 6(28), pp. 5976-5985, 23 November, 2011
Available online at http://www.academicjournals.org/SRE
DOI: 10.5897/SRE11.836
ISSN 1992-2248 ©2011 Academic Journals

Full Length Research Paper

Prediction of nickel-based super alloy surface roughness in CNC end milling operation using connectionist models

Vedat TOPUZ[1]*, Mustafa AY[2] and Gültekin BASMACI[1]

[1]Vocational School of Technical Sciences, Marmara University, Göztepe, 34722, Istanbul, Turkey.
[2]Technical Education Faculty, Marmara University, Göztepe, 34722, Istanbul, Turkey.

Accepted 16 August, 2011

This paper outlines a comparative study of different Artificial Neural Network models and 'adaptive-networks-based fuzzy inference systems' approach for predicting the surface roughness (Ra) using cutting speed (v), feed rate (fr) and cutting depth (d) when machining Nickel-based super alloys with uncoated carbide tool under milling conditions. Nickel-based super alloys are generally known to be one of the most difficult materials to machine because of hardness, high strength at high temperature, affinity to react with the tool materials and low thermal diffusivity. In this work, it was develop an accurate and reliable model for prediction of surface roughness for Nickel-based super alloys in end milling operations. In this study, to ensure the effectiveness of connectionist techniques, different models such as multi-layered perceptron, Elman recurrent neural network, radial basis function network and Adaptive-Networks-based Fuzzy Inference Systems were designed and optimal networks parameters were found. All the designed networks modeling and prediction ability were found using root mean square errors and correlations analysis. Finally, it was concluded that Elman recurrent neural network and adaptive-networks-based fuzzy inference systems could be used to predict the surface roughness for nickel-based super alloys in end milling process compared to the other connectionist models.

Key words: Surface roughness, nickel-based super alloy, artificial neural networks, adaptive-networks-based fuzzy inference systems.

INTRODUCTION

Nickel-based super alloys play an extremely important role in gas turbine engines. In addition to their use in aircraft, marine, industrial and vehicular gas turbines, Nickel-based super alloys are also used in space vehicles, rocket engines, experimental aircrafts, nuclear reactors, submarines, steam power plants, petrochemical equipments and other high-temperature applications (Bradley, 1988). Hence, nowadays the ability to machine Nickel-based super alloy is increasingly demanded. Nickel-based super alloys are known as some of the most difficult super alloys to machine in order to satisfy production and quality requirement (Choudhury, 1998). Metal machining has a long history and good cutting tools have been developed, however, advanced materials like Nickel-based super alloys are difficult to machine. The basic reasons for the difficulty in machining this alloy are, high work hardening rates at machining strain rates, abrasiveness, low thermal properties leading to high cutting temperatures, tough, gummy and strong tendency to weld to tool and to form a built up edge, tendency for maximum tool-face temperature to be close to the tool tip (Bradley, 1988). End milling is one of the most widely used (after turning) material cutting operations in industry. The aerospace industry places heavy demand on the end milling process due to both the shape and complexity of the finished dimensions. According to

*Corresponding author. E-mail: vtopuz@marmara.edu.tr.

theoretical models, the cutting force depends on the area of the chip (feed and depth of cut), the material, cutting tool properties and experimentally determined constants.

Restrictions concerning the characteristics of the workpiece-cutting tool-machine system have an impact on cutting conditions which is reflected on the surface texture and dimensional deviations of the product. ANN was used to predict the surface roughness in turning operation using mild steel work-pieces and high-speed steel as the cutting tool (Pal, 2005). They found performance of trained ANN satisfactory. Radial basis function network (RBFN) was used for predicting the upper and lower estimates of the surface roughness in a turning process. They found performance of the RBFN was inferior compared to multi layer perceptron (MLP) network (Sonar et al., 2006). In addition, ANN was used in grinding operations (Maksoud et al., 2003). They used ANN to control the grinding process for desired roughness. They found that ANN controller showed good performance in achieving the required workpiece surface roughness in real grinding operations. ANN trained with genetic algorithm (GA) and particle swarm optimization (PSO) was also developed for the prediction of surface roughness (Jesuthanam et al., 2007). They found that ANN-GA-PSO approach requires less number of epochs and RMS error than the classical ANN applications. On the other hand, ANFIS was also used to predict the workpiece surface roughness for milling process (Lo, 2003). He used triangular and trapezoidal membership functions and they found that both membership functions achieved very satisfactory accuracy.

Most of the ANN applications use simple MLP network training backpropagation algorithm. In this paper, it also used RBFN, Elman recurrent neural network (ERNN) and ANFIS, which are well-established techniques for function approximation. Generalization performances of all these approaches are assessed by comparing with experimental data.

MATERIALS AND METHODS

Milling is carried out on a milling machine that provides the power to turn the cutting tool at a given rotational speed and feed to the work piece at a specified rate and depth of cut. Therefore, three cutting parameters (cutting speed, feed rate and depth of cut) need to be determined in end milling operation and surface roughness strongly correlated with these parameters (Wang, 2004). The purpose of end milling operations is producing a low surface roughness of the machined work piece. Surface roughness is a measure of the technological quality of a product and a factor that greatly influences manufacturing cost. It describes the geometry of the machined surface and combined with the surface texture, which is process dependent, can play an important role on the operational characteristics of the part. The achievement of a desirable value is a repetitive and empirical process that can be very much time consuming. The part must be machined more than once until an acceptable value is obtained. The mechanism behind the formation of surface roughness is very complicated and process dependent, therefore it is very difficult to calculate its value through analytically.

Various theoretical models are not accurate enough and applied only to a limited range of processes and cutting conditions or must be used in conjunction with obscure diagrams and statistical tables.

Cutting performance measurement

The machined surface roughness was measured (according to ISO 4287/1) by a Marh profilometer. The average surface roughness (Ra), being the arithmetic average of the absolute value of the heights of roughness irregularities from the mean value measured within a sampling length of 5.6 mm. A machining center “VCM 550 John FORD” with 16 kW power and a maximum spindle speed of 6000 rpm was used. A Kistler piezoelectric dynamometer 9265B with a load amplifier was used to acquire the forces (Fx, Fy, Fz). Data acquisitions were made through piezoelectric dynamometer by interface RS 232 to load amplifier and to PC using to appropriate software Dynoware Kistler (Basmaci, 2002). Realized measurement system block diagram is shown in Figure 1.

Selection of the cutting parameters

The cutting experiments were carried out on CNC milling machine using uncoated carbide end milling with the grade of K10 for the machining of Nickel-based super alloys (the chemical composition of the work piece material confirms to the following specification: (wt%): 0.02 C; 0.09 Mn; 0.104 Si; 1.26 Ti; 0.259 Al; 0.265 Co; 6.73 Mo; 17.95 Fe; 22.75 Cr; 47.34 Ni and the hardness of the workpiece material was measured and found to be 280 BHN). Using this machine and super alloy totally, 60 experiments were carried out. Cutting speed (v) 30, 40, 50, 60 and 70 m/min, feed rate (fr) between 80 and 200 mm/min, and the cutting depth (*d*) 0.5, 1, and 1.5 mm were selected. Among them, 36 train, 12 test and 12 validation data sets randomly selected which are given in Table 1.

Network models

Artificial neural networks (ANNs) and adaptive-networks-based fuzzy inference system (ANFIS) have been applied to a large number of problems because of their non-linear system modeling capacity. Given a sample vectors, ANNs and ANFIS are able to map the relationship between input and output; they “learn” this relationship, and store it into their parameters. As these two characteristics suggest, they should prove to be particularly useful when little a priori knowledge about the laws that govern the system that generated the data ability by their learning ability. They offer parallel, methodology predicts the surface roughness accurately by utilizing small sized training and testing data sets. ANNs were designed to mimic the characteristics of the biological neurons in the human brain and nervous system. When the network is adequately trained, it is able to generalize relevant output for a set of input data. Learning typically occurs through the training. The training algorithm adjusts the connection weights (synapses) iteratively. On the other hand, fuzzy logic uses the experiences of a human expert working with linguistic expressions. It also compensates for inadequate and uncertain knowledge about the system. Once a fuzzy inference system (FIS) is equipped with learning capability, all the design methodologies for ANN become directly applicable to FIS, called ANFIS.

Multi layer perceptron (MLP)

Typical MLP network is arranged in layers of neurons, where each neuron in a layer computes the sum of its inputs $\mathbf{x} = [v\ fr\ d]^{\mathrm{T}}$ and passes this sum through an activation function (f). The output of the network ($\mathbf{o}$) is defined as a matrix form:

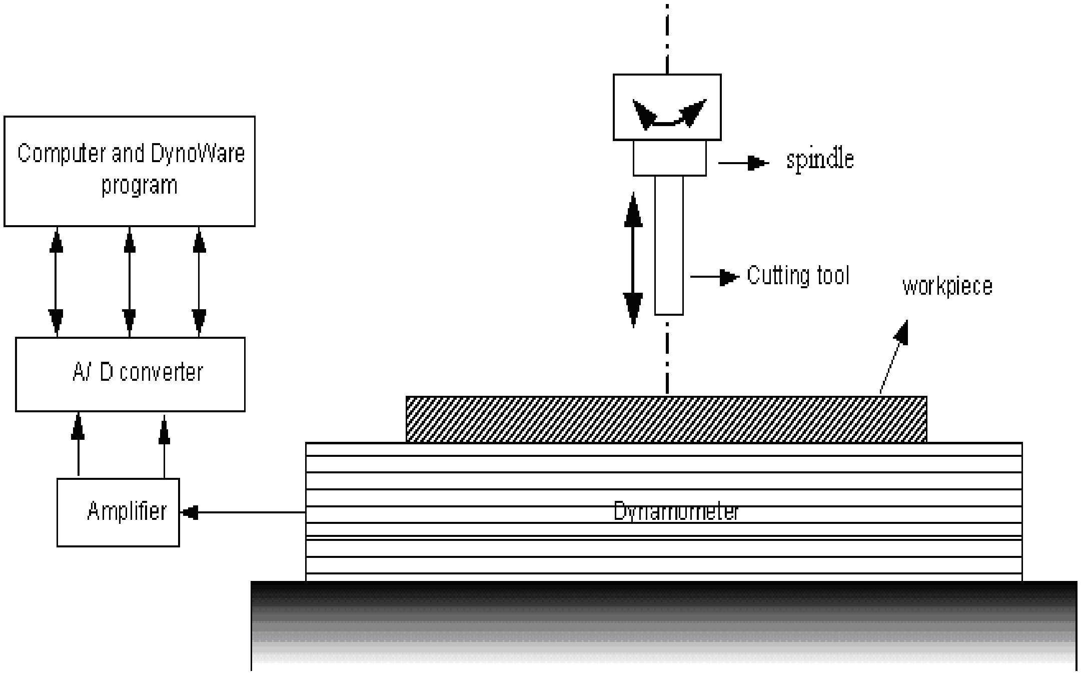

Figure 1. Block diagram of roughness measurement system.

$$\mathbf{o} = f^2(\mathbf{W}^2 f^1(\mathbf{W}^1\mathbf{x}+\mathbf{b}^1)+\mathbf{b}^2) \tag{1}$$

Where superscript defines the layer number, S0 is input number, S1 is hidden layer neuron number, n is activation function input, $\mathbf{W}$ is weight matrices defined as:

$$\mathbf{W^1} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,S0} \\ w_{2,1} & w_{2,2} & \dots & . \\ . & . & \dots & w_{2,S0} \\ w_{S1,1} & w_{S1,2} & \dots & w_{S1,S0} \end{bmatrix},$$

$$\mathbf{W^2} = [w_{1,1}\ w_{1,2} \dots w_{1,S1}]),$$

b is bias vector defined as:

$$\mathrm{b}^1 = [b_1\ b_2\ ..\ b_{S1}]^{\mathrm{T}},\ \mathrm{b}^2 = [b_1],$$

f are activation functions defined as:

$$f^1 = \frac{exp^n - exp^{-n}}{exp^n + exp^{-n}},\ f^2 = n.$$

Figure 2 shows a realized one hidden layer MLP network for this work. MLP networks learn any input output relation adjusting the weights using back propagation approach. The back propagation algorithm is generalization of the least mean square (LMS) algorithm (Hagan et al., 1996.). This algorithm adjusts the weights in order to minimize the mean square error as follows:

$$e = \frac{1}{2}\sum_{\gamma=1}^{p}(t^\gamma - o^\gamma)^2 \tag{2}$$

Where: t is target, o is MLP output, γ is the sample instant in q size.

The steepest descent algorithm iteratively decreases network error during learning phase at each epoch as follows:

$$w^m(k+1) = w(k) - \eta\frac{\partial e}{\partial w}$$

$$b^m(k+1) = b(k) - \eta\frac{\partial e}{\partial b} \tag{3}$$

Where η is learning rate.

Elman recurrent neural networks (ERNN)

ERNN (also known as partially recurrent neural network) are a subclass of recurrent networks is shown in Figure 3. It is MLP network augmented with additional context layers (W^0), storing output values ($\mathbf{y}$), of one of the layers delayed (z^{-1}) by one-step and used for activating this other layer in the next time (t) step:

$$\mathbf{y}_{(\mathbf{t+1})} = f^1(\mathbf{W}^1\mathbf{x}+\mathbf{b}^1)+\mathbf{y}_{(\mathbf{t})}\mathbf{W}^0 \tag{4}$$

While ERNN use identical training algorithm as MLP, context layer

Table 1. Training, testing and validation data sets.

Training data set					Test data set				
No	**v**	**fr**	**d**	**Ra**	**No**	**v**	**fr**	**d**	**Ra**
1	30	140	0.5	0.533	1	30	90	0.5	0.504
2	30	80	1.0	0.601	2	30	200	0.5	0.781
3	30	130	1.0	0.789	3	30	80	1.5	0.697
4	30	140	1.0	0.813	4	30	140	1.5	0.982
5	30	200	1.0	1.112	5	30	200	1.5	1.231
6	30	180	1.5	1.056	6	40	80	0.5	0.368
7	40	110	0.5	0.413	7	40	140	1.0	0.715
8	40	200	0.5	0.673	8	40	200	1.0	0.929
9	40	80	1.0	0.554	9	40	140	1.5	0.723
10	40	80	1.5	0.592	10	40	200	1.5	0.967
11	40	105	1.5	0.658	11	60	140	0.5	0.584
12	50	80	0.5	0.506	12	70	80	1.5	0.509
13	50	200	0.5	0.713					
14	50	80	1	0.652					
15	50	120	1	0.734					
16	50	200	1	0.978					
17	50	80	1.5	0.732	**Validation data set**				
18	50	95	1.5	0.851	**No**	***v***	***fr***	***d***	***Ra***
19	50	140	1.5	1.049	1	30	140	0.5	0.533
20	60	110	0.5	0.489	2	40	110	0.5	0.413
21	60	200	0.5	0.627	3	40	80	1.5	0.592
22	60	80	1	0.598	4	50	80	0.5	0.506
23	60	140	1	0.715	5	50	170	0.5	0.685
24	60	80	1.5	0.881	6	50	80	1	0.652
25	60	90	1.5	1.047	7	50	140	1	0.814
26	60	140	1.5	1.179	8	50	200	1	0.978
27	60	200	1.5	1.653	9	50	200	1.5	1.205
28	70	140	0.5	0.36	10	60	140	0.5	0.584
29	70	200	0.5	0.581	11	70	140	1	0.662
30	70	80	1	0.417	12	70	140	1.5	0.735
31	70	120	1	0.614					
32	70	140	1	0.662					
33	70	200	1	0.801					
34	70	170	1.5	0.829					
35	70	140	1.5	0.735					
36	70	200	1.5	0.938					

weight (W^0) is not updated as in Equation 3.

Radial basis function network (RBFN)

RBFN has a feed-forward structure consisting of two layers, nonlinear hidden layer and linear output layer as depicted in Figure 4. The proposed model uses Gaussian kernel (Ψ) as the hidden layer activation function. The output layer implements a linear combiner of the basis function responses defined as:

$$\mathbf{o} = b + \sum_{j=1}^{q} w_{i,j} \Psi_j \tag{5}$$

Where, q is the sample size, Ψ_j is response of the jth hidden neuron described as:

$$\Psi j = \exp\left[-\frac{\left\| \mathbf{x} - c_j \right\|^2}{2\sigma_j{}^2} \right] \tag{6}$$

Where, c_j is Gaussian function center value and σ_j is its variance.

RBFN training has two-stage procedure. In the first stage, the

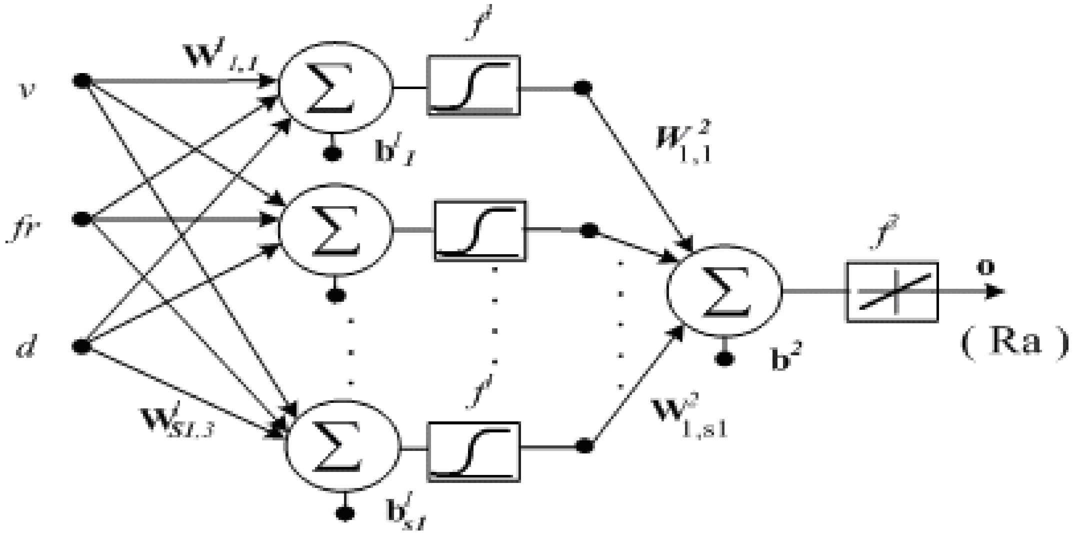

Figure 2. Realized MLP network structure.

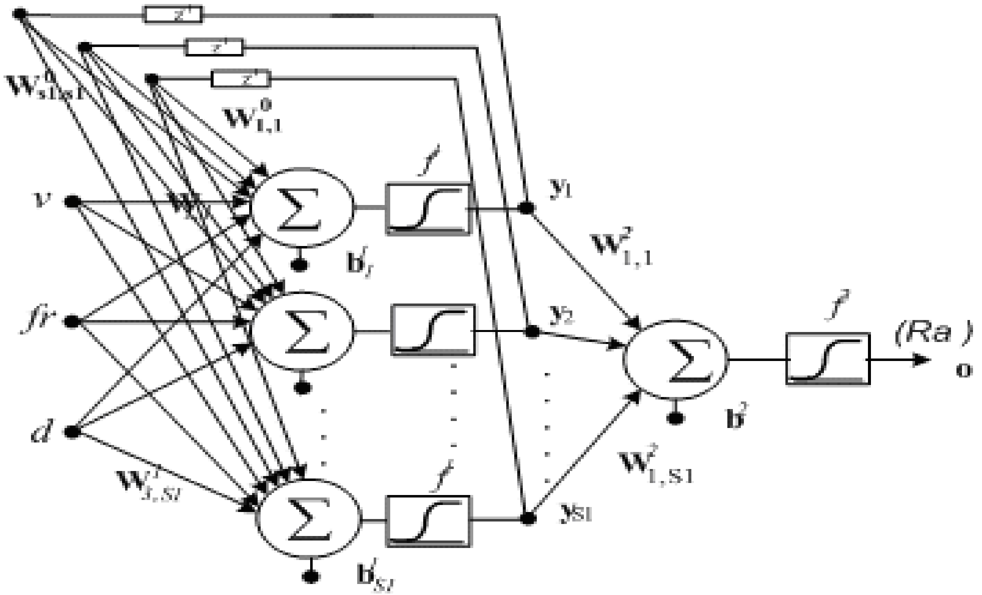

Figure 3. Realized ERNN structure.

input data set is used to determine the center locations (c_j) using unsupervised clustering algorithm such as the *K*-means algorithm and choose the radii (σ_j) by the *k*-nearest neighbor rule. The second step is to update the weights (W) of the output layer while keeping the (c_j) and (σ_j) are fixed.

Adaptive-networks-based fuzzy inference systems (ANFIS)

Proposed first-order Sugeno type ANFIS architecture is shown in Figure 5. This architecture with typical fuzzy if–then rules can be expressed as:

$$\mathrm{R}_1 : \text{If } x_1 \text{ is } \mathrm{A}_{1,1} \text{ and } x_2 \text{ is } \mathrm{A}_{2,1} \text{ .... and } x_i \text{ is } \mathrm{A}_{i,1} \text{ then } \mathrm{o}_1 = \mathrm{k}_{1,1}x_1 + \mathrm{k}_{2,1}x_2 + \text{ .... } + \mathrm{k}_{i,1}x_3 + \mathrm{r}_1$$

$$\mathrm{R}_2 : \text{If } x_1 \text{ is } \mathrm{A}_{1,1} \text{ and } x_2 \text{ is } \mathrm{A}_{2,1} \text{ .... and } x_i \text{ is } \mathrm{A}_{i,2} \text{ then } \mathrm{o}_2 = \mathrm{k}_{1,2}x_1 + \mathrm{k}_{2,2}x_2 + \text{ .... } + \mathrm{k}_{i,2}x_3 + \mathrm{r}_2$$

$$\mathrm{R}_{iXj} : \text{If } x_1 \text{ is } \mathrm{A}_{1},_{j} \text{ and } x_2 \text{ is } \mathrm{A}_{2,j} \text{ .... and } x_i \text{ is } \mathrm{A}_{i,j} \text{ then } \mathrm{o}_{iXj} = \mathrm{k}_{1,j}x_1 + \mathrm{k}_{2,j}x_2 + \text{ .... } + \mathrm{k}_{i,j}x_3 + \mathrm{r}_{iXj}$$

Where R_{ixj} is rule set, i is crisp input numbers, j is fuzzy set

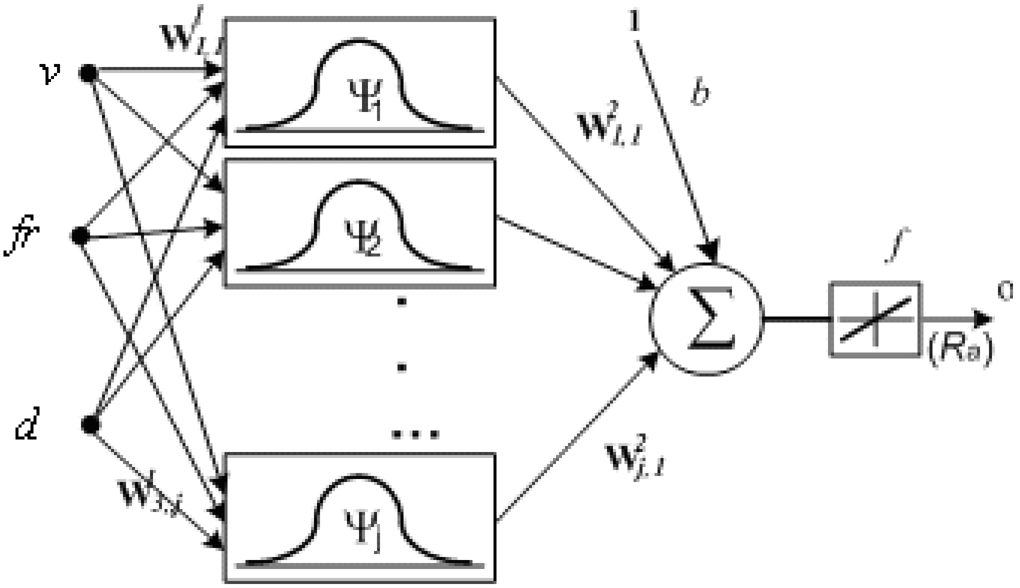

Figure 4. Realized RBFN structure.

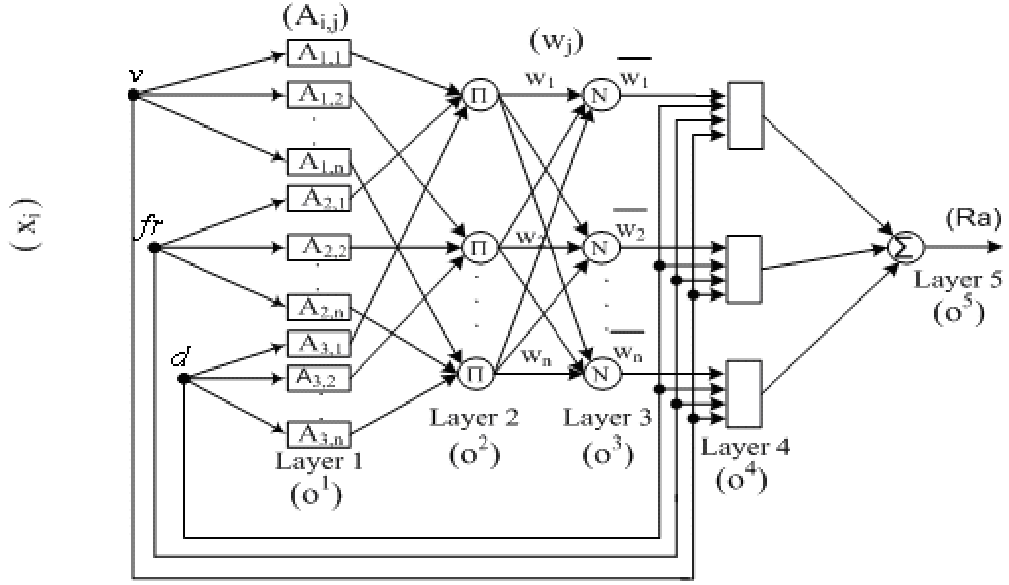

Figure 5. Realized ANFIS structure.

numbers for every inputs, x_i is crisp input, $A_{i,j}$ is the nonlinear parameter with linguistic labels, o_{ixj} is ANFIS output, $k_{i,j}$ is output polynomial coefficient, r_{ixj} is output constant.

The entire system architecture consists of five layers, the fuzzy layer (O^1), product layer (O^2), normalized layer (O^3), de-fuzzy layer (O^4) and total output layer (O^5). The membership relationship between the output and input functions of the fuzzy layer can be expressed as follows:

$$o^1_{i,j}=\mu_{A_{i,j}}(x_i) \qquad i=1....m, j=1....n \tag{7}$$

Where $\mu_{A_{i,j}}$ membership grade of a fuzzy set $A_{i,j}$.

In other words, $o_{i,j}$ is the membership grade of a fuzzy set $A_{i,j}$ and it specifies the degree to which the given input x_i satisfies the quantifier $A_{i,j}$. Parameters in this layer are referred to as premise parameters. Layer 2 is the product layer and is labeled with (Π).

The outputs of the layer (w_j) is the weight functions of the next layer as follows:

$$o^2{}_j = w_j = (\prod_{i=1} \mu_{A_{i,j}}(x_i)) \qquad j=1....n \tag{8}$$

The third layer is the normalized layer, whose nodes are labeled (N). Their function is to normalize the weight function in the following process:

$$o^3{}_j = \overline{w}_j = (\frac{w_j}{\sum_{j=1} w_j}) \qquad j=1....n \tag{9}$$

The fourth layer is the de-fuzzy layer, whose nodes are adaptive. The de-fuzzy relationship between the input and output of this layer can be defined as follows:

$$o^4{}_j = \overline{w}_j f_j = \overline{w}_j (\sum_i k_{i,j} x_i + r_j) \qquad j=1....n \tag{10}$$

Where $k_{i,j}$ and r_j are the linear parameters or consequent parameters of the node.

The fifth layer is the total output layer. The output of this layer is the total of input signals which represents the results of surface roughness. The results can be written as follows:

$$o^5 = \sum_j o_j{}^4 \tag{11}$$

Thus, it was constructed as an adaptive network that is functionally equivalent to a Sugeno fuzzy model using hybrid-learning algorithm. For this learning approach premise parameters, defining the membership functions, ANFIS employs gradient descent to fine-tune them. For consequent parameters, defining the coefficients of each output equations; ANFIS uses the least-squares method to identify them (Jang et al., 1997).

RESULTS

The number of hidden layer and neurons in the hidden layer(s) play very important roles for ANNs and choice of these numbers depends on the application. Influenced by theoretical works proved that single hidden layer is sufficient for ANNs to approximate any complex nonlinear function with any desired accuracy. In addition, determining the optimal number of hidden neurons is still a question to deal with. Although there is no theoretical basis for selecting these parameters, a few systematic approaches are also reported but the most common way of determining the number of hidden neurons is still trial and error approach. It was proposed the same approach for MLP and ERNN; therefore, 5, 10, 15, 20 and 25 neurons are selected in one hidden layer. It was also realized that to use more than 25 neurons in the hidden layer did not increase the generalization ability of designed networks. Because the learning rate (η) greatly affects the ANN performance, different coefficient has been tried and 0.05 was obtained as the best rate. Designed MLP and ERNN were trained using back propagation learning algorithms, which is described in this study. In addition, it also used RBFN with 36 neurons (which equals the number of training sample size) to demonstrate the efficiently of this type networks. Once trained, the network can be used for predicting the output for any input vector from the input space. This is called the "generalization property" of the network. To show this property of trained networks, same experiment was done with testing data set that are not involved in training data set. It also use the early stopping strategy to improve the generalization properties of designed network. In this strategy, the validation data set is used and the error on the validation set is monitored during the training process.

The validation error normally decreases during the initial phase of training. However, when the network begins to overfit the data, the error on the validation set typically begins to rise (Demut, 2002). When the validation error increases for a specified number of iterations, which chosen 10, the training is stopped. At the end of the training and testing experiments, obtained root mean square error (RMSE) values and correlations coefficients (R) are given in Table 2. In a conventional FIS, an expert who is familiar with the system determines the number of rules, types of membership functions (MF) and MF boundary. In addition to this approach, there are other techniques such as clustering methods. In this work, because no expert is available, the number and type of MFs are chosen by trial and error effort. For an ANFIS premise part triangle, trapezoid or Gaussian MF are chosen and linear or constant parameters are also chosen as consequent part. ANFIS architecture was trained with hybrid-learning algorithm. After the several experiments, it was realized that to use more than two MFs does not improve the generalization ability on the other hand dramatically increases training time. For this reason, only two MFs were used and results RMSE values and correlations coefficients were given in Table 2.

DISCUSSION

Consequently, the best models type was chosen for every architecture using Table 2. These results shows that realized ERNN and ANFIS architectures have similar and good prediction performance (R_{ERNN_25} = 0.9315, $R_{Gaussian\_2\_Linear}$ = 0.9229). MLP (R_{MLP_25} = 0.895) has no good prediction performance as ERNN and ANFIS but certainly better than RBFN (R_{RBFN} = 0.456). To understand the modeling capability of designed networks clearly, training results were given in Figure 6. This figure shows that while RBFN has perfect modeling ability (R_{RBF}

Table 2. Different ANN and ANFIS models RMSE values and correlations (R) coefficients.

Models	Training data set		Testing data set	
	RMSE	Correlation (R)	RMSE	Correlation (R)
MLP_5	0.0270	0.8752	0.0427	0.7970
MLP_10	0.0186	0.8782	0.0270	0.8090
MLP_15	0.0165	0.8854	0.0209	0.8264
MLP_20	0.0021	0.9851	0.0208	0.8482
MLP_25	0.0017	0.9897	0.0133	0.8950
ERNN_5	0.0024	0.9827	0.0199	0.8711
ERNN_10	0.0020	0.9876	0.0159	0.8867
ERNN_15	0.0017	0.9896	0.0110	0.9043
ERNN_20	0.0013	0.9916	0.0100	0.9176
ERNN_25	0.0006	0.9964	0.0080	0.9315
RBFN	$1.17.10^{-30}$	1	0.3517	0.4565
ANFIS triangle_2_linear	0.0025	0.9843	0.0111	0.8965
ANFIS triangle_2_constant	0.0186	0.8771	0.0232	0.8086
ANFIS gaussian_2_Linear	0.0009	0.9943	0.0116	0.9229
ANFIS gaussian_2_constant	0.0262	0.872	0.0193	0.7665
ANFIS trapezoid_2_linear	0.0007	0.9954	0.0172	0.8820
ANFIS trapezoid-2_constant	0.0215	0.8565	0.0241	0.7840

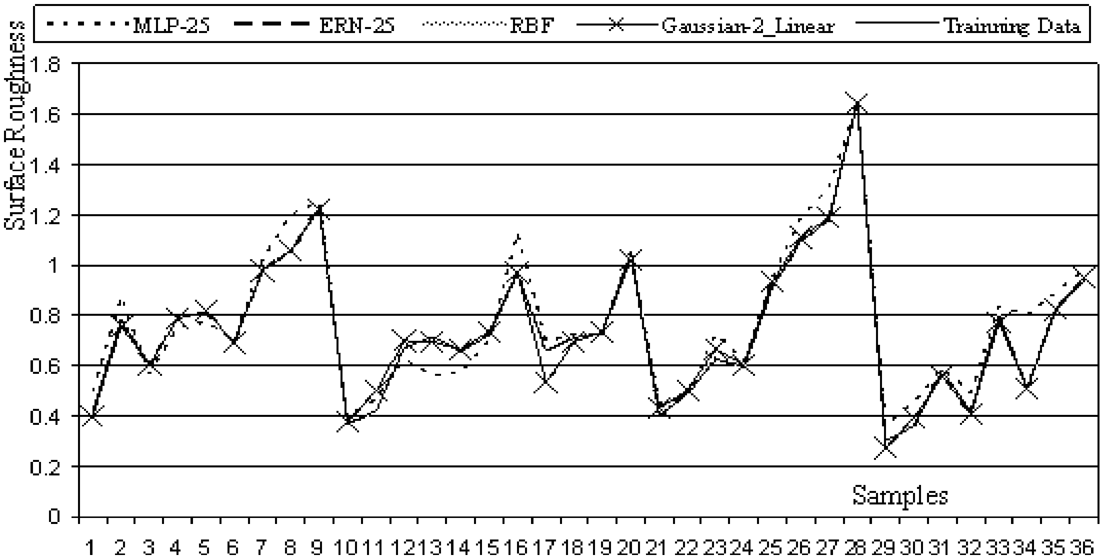

Figure 6. Modeling the surface roughness capability of different networks.

= 1); others (R_{MLP_25} = 0.9897; R_{ERNN_25} = 0.9964; $R_{Gaussian\_2\_Linear}$ = 0.9943) are also good enough to be modeling the surface roughness. In addition, to show the prediction ability of designed networks clearly testing results were given in Figure 7. This figure shows us that, ERNN_25 or Gaussian_2_Linear networks can be used to prediction of surface roughness of nickel-based super alloy in CNC end milling operation. On the other hand, RBFN is not good enough to use the prediction of surface roughness.

Conclusions

In this study, the performance of MLP, ERNN, RBFN and ANFIS based connectionist architectures for predicting

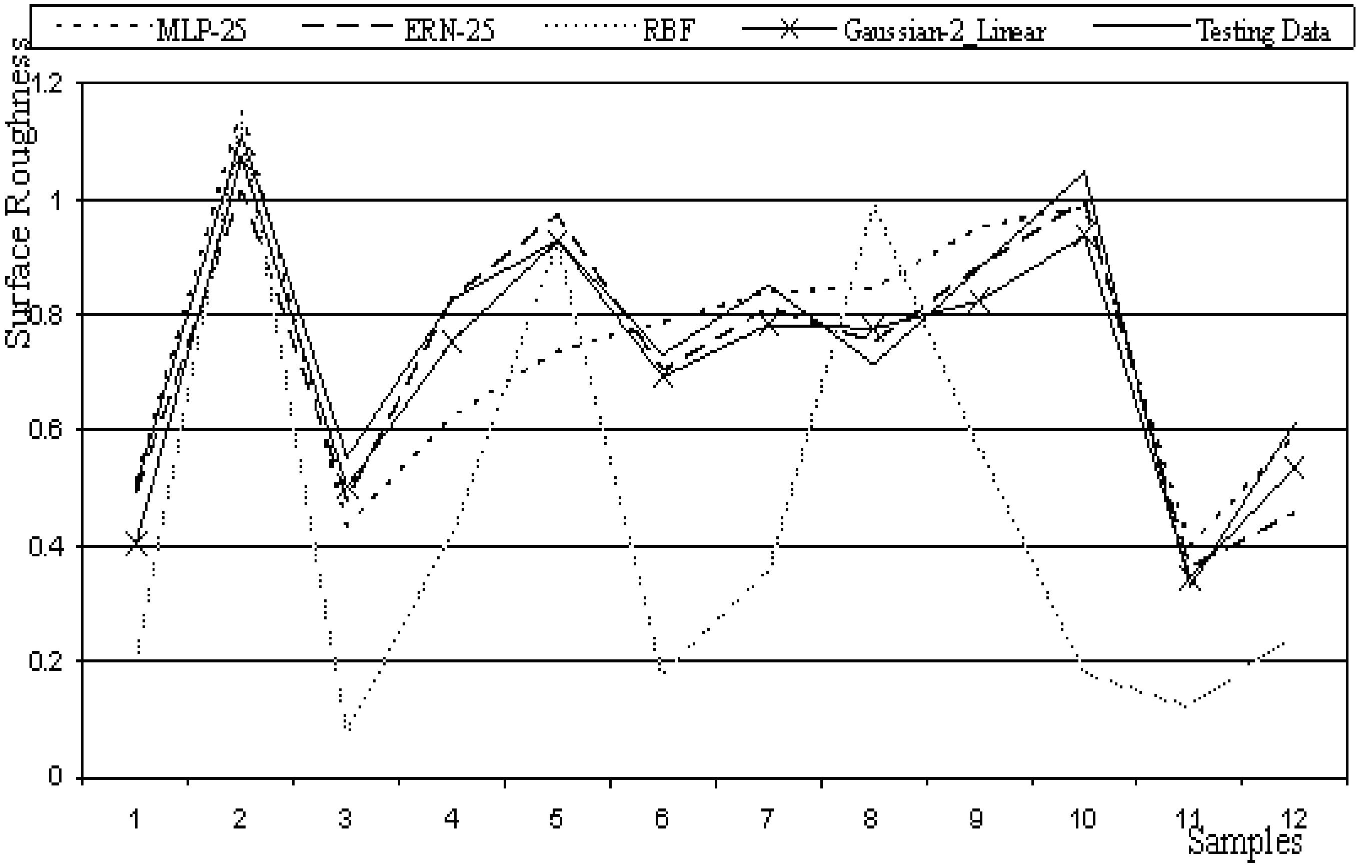

Figure 7. Prediction of the surface roughness ability of different networks.

the surface roughness (Ra) in end milling process for nickel based super alloy were compared. Cutting speed (v), feed rate (fr) and cutting depth (d) were selected for input parameters. Several experiments were performed to find the best models of every architecture and were compared the prediction ability of designed networks. In addition to results discussed earlier, the following conclusions can be summarized:

1) Ra values from experimental measurements trace a regular path in a wide range of cutting conditions. It can be observed that surface roughness is considerably affected by all cutting parameters. Surface roughness becomes particularly higher for lower values of machining tolerance.
2) MLP and ERNN performance were considerably affected by neurons number in the hidden layer. On the other hand, too many neurons did not improve generalization ability of networks.
3) RBFN has perfect modeling capability (exact matching) but very poor generalization ability. On the other hand, generalization ability of RBFN may be enhanced using different activation functions or learning algorithms. However, the performance was not tested of this proposition on this work.
4) Compared to the other models, ERNN and ANFIS could efficiently predict the surface roughness (Ra) in end milling process.
5) ERNN network takes more training time but it is dependent on the training data size and network parameters. It can be inferred that ERNN and ANFIS could yield more accurate results, if good data selection strategies, training paradigms and network input and output representations are determined properly.
6) Finally, ERNN is the best network model to predict the ‘nickel-based super alloy surface roughness’ in CNC end milling operation.

REFERENCES

Basmaci G (2002). Cutting tool parameters in a CNC machine by using Nickel based super alloys, Marmara University Institute of Pure and Applied Sciences Mechanical Education Branch MSc. Thesis., Turkey.

Bradley E (1988). Super alloys A Technical Guide, ASM International, USA.

Choudhury I A, El-Baradie M A. (1998). Machinability of Nickel-base super alloys: a general review. J. Mat. Proc. Techol.. 77: 278-284.

Demut H, Beale M (2002). Neural Network Toolbox User’s Guide, The

MathWorks, Inc. Natick, MA.

Hagan MT, Demuth HB, Beale M (1996). Neural Network Design, PWS Publishing, Boston, MA.

Jang J, SR. Sun, CT, Mizutani E (1997). Neuro-fuzzy and soft computing. A computational approach to learning and machine intelligence, Prentice Hall, New Jersey.

Jesuthanam CP, Kumanan S, Asokan P (2007). Surface Roughness Prediction Using Hybrid Neural Networks. Mach. Sci. Technol., 11(2): 271- 286.

Lo SP (2003). An adaptive-network based fuzzy inference system for prediction of workpiece surface roughness in end milling. J. Mater. Process. Technol., 142: 665-675.

Maksoud TMA, Atia MR, Koura MM (2003). Applications of Artificial Intelligence to Grinding Operations via Neural Network. Mach. Sci. Technol., 7(3): 361-387.

Pal SK, Chakraborty D (2005). Surface roughness prediction in turning using artificial neural network. Neural Comput. Appl., 14: 319-324.

Sonar DK, Dixit, US, Ojha D.K (2006). The application of a radial basis function neural network for predicting the surface roughness in a turning process. International J. Adv. Manuf. Technol., 27: 661-666.

Wang MY, Chang, HY (2004). Experimental study of surface roughness in slot end milling. International J. Mach. Tool Manuf., 44: 51-57.