Research Article

Prediction of Surface Roughness in End Milling Process Using Intelligent Systems: A Comparative Study

Abdel Badie Sharkawy

Department of Mechanical Engineering, Faculty of Engineering, Assiut University, Assiut 71516, Egypt

Correspondence should be addressed to Abdel Badie Sharkawy, ab.shark@aun.edu.eg

Received 5 May 2011; Revised 30 June 2011; Accepted 16 August 2011

Academic Editor: Serafin Moral

Copyright © 2011 Abdel Badie Sharkawy. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A study is presented to model surface roughness in end milling process. Three types of intelligent networks have been considered. They are (i) radial basis function neural networks (RBFNs), (ii) adaptive neurofuzzy inference systems (ANFISs), and (iii) genetically evolved fuzzy inference systems (G-FISs). The machining parameters, namely, the spindle speed, feed rate, and depth of cut have been used as inputs to model the workpiece surface roughness. The goal is to get the best prediction accuracy. The procedure is illustrated using experimental data of end milling 6061 aluminum alloy. The three networks have been trained using experimental training data. After training, they have been examined using another set of data, that is, validation data. Results are compared with previously published results. It is concluded that ANFIS networks may suffer the local minima problem, and genetic tuning of fuzzy networks cannot insure *perfect* optimality unless suitable parameter setting (population size, number of generations etc.) and tuning range for the FIS, parameters are used which can be hardly satisfied. It is shown that the RBFN model has the best performance (prediction accuracy) in this particular case.

1. Introduction

End milling is one of the most common metal removal operation-encountered in industrial processes. It is widely used in a variety of manufacturing industries including the aerospace and automotive sectors, where quality is an important factor in the production of slots, pockets, precision molds, and dies. The quality of the surface plays a very important role in the performance of milling as a good-quality milled surface significantly improves fatigue strength, corrosion resistance, and creep life. Surface roughness also affects several functional attributes of parts, such as contact causing surface friction, wearing, light reflection, heat transmission, ability of distributing and holding of lubricant, coating, and resisting fatigue.

Conventionally, the setup parameters for the end milling operation are usually selected with the aid of trial cutting experiments, which are both time consuming and costly. The mechanism behind the formation of surface finish is very complicated and process dependent, therefore it is very difficult to calculate its value through analytical formula.

Moreover, the surface finish of the product depends on the experience of an operator and the machining environment. Therefore, there is a need for the development of a simulation system which is capable of predicting the surface finish of a workpiece and optimizing cutting conditions.

Modeling techniques for the prediction of surface roughness can be classified into three groups which are experimental models, analytical models, and artificial intelligence-based models. Experimental and analytical models can be developed by using conventional approaches such as statistical regression techniques which are usually called the response surface method (RSM) [1–3]. On the other hand, artificial-intelligence-based models are developed using non-conventional approaches such as artificial neural networks [4–8], fuzzy logic, genetic algorithms [9], and hybrid systems [10–13]. References [8, 14] present reviews for the various methodologies and practices that are being employed for the prediction of surface roughness.

Function approximation for a set of input-output pairs has numerous scientific and engineering applications such as signal processing, image restoration, pattern recognition,

control systems, and system identifications [15, 16]. The meaning of function approximation is to model a desired function or an input-output relation from a set of input-output sample data. Recently, artificial-intelligence-based models have become the preferred trend, and these are applied by most researchers to develop a model for near optimal conditions in machining. Both fuzzy inference systems (FISs) and neural networks are universal function approximators. They can get good performance for nonlinear functions, provided that there are sufficient rules in FISs or hidden neurons in neural network [17].

Karayel [6] has implemented the artificial neural network (ANN) in the prediction and control of surface roughness in CNC lathe. A feedforward multilayered neural network was developed and trained using the scaled conjugate gradient algorithm, which is a type of backpropagation. The number of iterations was stopped at 8,000 after trial and error procedure. Using some selected data from the experimentation results, the average absolute predicting error was 0.0229. Topal [7] has studied the prediction of surface roughness in flat end milling. He used a 3-layered feed forward multilayer perceptron network which is trained with backpropagation (BP) technique. The inputs are the cutting speed, feed rate, depth of cut, and the stepover ratio. The hidden layer has 10 neurons. The achieved average root mean squared prediction error (RMSE) is 0.04 after 65,000 iterations. Oktem et al. [11] have proposed an ANN model coupled with genetic algorithm (GA). The prediction error is less than 0.0534. However, to obtain the successful model of ANN, it depends totally on the process of trial and error with some factors to consider. Until now, there are no clear rules that could serve as a basis to be followed in producing the perfect ANN.

Roy [10] has designed an expert system using fuzzy inference system (FIS) and genetic algorithm (GA) so that the surface roughness in ultraprecision turning of metal matrix composite can be modeled for a set of given cutting parameters, namely, spindle speed, feed rate, and depth of cut. The maximum absolute value of prediction error for several case studies is less than 0.053. Colak et al. [9] have proposed a gene expression programming method for predicting surface roughness of milling surface relation to cutting parameters in CNC milling machines. Cutting speed, feed rate, and depth of cut of end milling operations were collected for predicting surface roughness. In their study, there is no quantification for the prediction error, however there are some figures presented there.

Few studies have been done related to the use of radial basis function networks (RBFNs) in the prediction of surface roughness in end milling. Lu [8] has used RBFN in the prediction of surface roughness in turning operation. The least mean squared error reached is 0.0439 in the training phase. The prediction error, however, is not clear from the contribution.

Radial basis function networks (RBFNs), as a special class of single hidden-layer feed forward neural networks, have been proved to be universal approximators [18–20]. One advantage of RBFNs compared with multilayer perceptrons is that the linearly weighted structure of RBFNs, where

parameters in the units of the hidden layer can often be prefixed, can easily be trained with a fast speed without involving nonlinear optimization. Another advantage of RBFNs, compared with other basis function networks, is that each basis function in the hidden units is a nonlinear mapping which maps a multivariable input to a scalar value, and thus the total number of candidate basis functions involved in an RBFNs model is not very large and does not increase when the number of input variables increases. With these attractive properties, RBFNs are an important and popular network model for function approximation [21].

Otherwise, various neurofuzzy inference systems (FISs) have also been used to determine the surface roughness in machining operations. An important issue in application of FISs in predicting problems is to extract the structure and type of fuzzy if-then rules from available input-output data. Given an FIS whose number and structure of fuzzy rules are known, the optimization techniques in ANN and genetic programming can be used to tune the shape of membership functions of fuzzy variables and other parameters of the fuzzy rule base.

Lo [12] has studied the implementation of an adaptive-network-based fuzzy inference system (ANFIS) to predict the workpiece surface roughness after the end milling process. Two different membership functions, triangular and trapezoidal, were adopted during the training process of ANFISs in order to compare the prediction accuracy of surface roughness by the two networks. When a triangular membership function was adapted, the prediction accuracy of ANFISs reached is as high as 96.5%. However, he has studied only the first-order (FO) Sugeno fuzzy inference system, and the learning mechanism used in his work is not clear.

More recently, Ho et al. [13] have proposed a genetic fuzzy inference system (G-FIS). The premise and consequent parameters of the fuzzy system have been optimally determined using genetic algorithms. They used the root mean squared error as the optimality criterion. In their network, they used Gaussian memberships to represent the input variables to the network. The trained fuzzy system can be considered as first-order (FO) Sugeno fuzzy inference system (FIS). The achieved prediction accuracy RMSE (root mean square of error) is 3.32%.

There are various machining parameters that can affect surface roughness. Beside feed rate, spindle speed, and depth of cut, there are many other parameters like tool geometry, vibration, workpiece hardness, surface temperature, the material being processed, cutting time, cutting forces, chip width and thickness, and even the machine with which the experiments are performed. More details can be found in [14]. In the present work, an attempt has been made to design approximating networks, so that the surface finish in end milling can be modeled for a set of input cutting parameters, namely, spindle speed, feed rate, and depth of cut. They are listed in Table 1. Several neural, neurofuzzy, and genetic fuzzy networks are developed and compared to each other in order to determine the most effective method for predicting the surface roughness in end milling process. The aim is to determine a network which can best capture the nonlinear mapping between the cutting parameters and the resulting

TABLE 1: End milling parameters used in the study.

Parameter	Meaning
S_m	Spindle speed (rpm)
F_m	Feed rate (ipm)
D_m	Depth of cut (in)
R_m	Roughness (μin)

surface roughness. Throughout this paper, the root mean square error (RMSE) is used as a measure of the prediction accuracy.

This paper proceeds as follows. In Section 2, the RBFNs used in this work is developed. Section 3 introduces the ANFIS structure and its implementation to construct four different fuzzy networks. Section 4 illustrates the use of genetic algorithms to build up two other optimized fuzzy systems. The experimental data sets used in this and previous studies are given in Section 5. Section 6 demonstrates the results of comparing the performance of the seven networks and previously published results. Section 7 offers our concluding remarks.

2. Radial Basis Function Network (RBFN) Approach

The different types of artificial neural network are, in practice back propagation neural networks (BPNNs), counter propagation neural networks (CPNNs), and radial basis function neural networks (RBFNs), and so forth. Even though BPNNs is widely used for a variety of systems, especially in the field of surface roughness prediction as it appears in these recent articles [22–26], it suffers from a number of drawbacks. First, it is very slow to converge because of the use of sigmoid nonlinear transformation functions. Second, it is not always simple or straightforward to design the topology of the network that would accurately represent the system being modeled.

RBFN has been chosen in this work because it has been proven that it is a very efficient network when function approximation is needed [5]. This artificial neural network has the following advantages:

- (a) it is very fast in comparison to backpropagation;
- (b) it has the ability of representing nonlinear functions;
- (c) it does not experience local minima problems of back-propagation.

The following four points give the main characteristics.

- (i) An RBFN is an ANN which uses radial basis functions as activation functions instead of sigmoid functions.
- (ii) A radial basis function is a function whose value depends only on the distance from a center point c

$$F(x) = f(|x - c|). \quad (1)$$

- (iii) The ANN output is a linear combination of radial basis functions.

- (iv) RBFNs are used in many applications like time series predictions and function approximation.

The theory of *multivariable interpolation* in high-dimensional space has a long history [19]. The interpolation (approximation), in its *strict* sense, may be stated as follows

Given a set of n different points $\{x_i \in R^m \mid i = 1, 2, \dots, n\}$ and a corresponding set of n real numbers $\{d_i \in R^1 \mid i = 1, 2, \dots, n\}$, find a function $F : R^n \rightarrow R^1$ that satisfies the interpolation condition:

$$F(x_i) = d_i, \quad i = 1, 2, \dots, n. \quad (2)$$

As mentioned earlier, the three end milling parameters considered in this study are spindle speed (S_m), feed rate (F_m), and depth of cut (D_m). They are contained in the vector x_i , that is, $m = 3$. The output is the surface roughness R_m contained in d_i , that is, $d_i = R_m$. For strict interpolation as specified here, the interpolating surface (i.e., the function F) is constrained to pass through all the training data points.

An RBFN is a multidimensional nonlinear function mapping that depends on the distance between the input vector and the center vector. An RBFN with an n -dimensional input $x \in R^n$ and a single output R^1 can be represented, as shown in Figure 1, by the weighted summation of a finite number of radial basis functions as follows [20].

The radial basis functions (RBFNs) technique consists of choosing a function F that has the following form:

$$F(x) = \sum_{i=1}^n w_i \phi(\|x - x_i\|), \quad (3)$$

where $\{\phi(\|x - x_i\|) \mid i = 1, 2, \dots, n\}$ is a set of n arbitrary functions (generally nonlinear) known as *radial basis functions*, and $\|\cdot\|$ denotes a *norm* that is usually Euclidean. The known data points $\{x_i \in R^m \mid i = 1, 2, \dots, n\}$ are taken to be the *centers* $c_i \in R^n$ of the radial basis functions, and w_i is a weight parameter vector to be determined.

Inserting the interpolation conditions of (2) in (3), we obtain the following set of simultaneous equations for the unknown coefficients (weights) of the expansion w_i :

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1n} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{n1} & \phi_{n2} & \cdots & \phi_{nn} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}, \quad (4)$$

where

$$\phi_{ji} = \phi(\|x_j - x_i\|), \quad (j, i) = 1, 2, \dots, n. \quad (5)$$

Let

$$d = [d_1, d_2, \dots, d_n]^T, \quad w = [w_1, w_2, \dots, w_n]^T. \quad (6)$$

The n -by-1 vectors d and w represent the *desired response vector* and *linear weight vector*, respectively, where n is the size

FIGURE 1: Block diagram representation of radial basis function network (RBFNs) with input $x \in R^m$ and output $F(x)$.

of the training sample. Let Φ denote an n -by- n matrix with elements ϕ_{ji} :

$$\Phi = \{\phi_{ji} \mid (j, i) = 1, 2, \dots, n\}. \quad (7)$$

This matrix is called the *interpolation matrix* [19]. We may then rewrite (4) in the compact form

$$\Phi w = d. \quad (8)$$

Assuming that Φ is nonsingular and therefore that the matrix Φ^{-1} exists, we may go to solve (8) for the weight vector w as shown by

$$w = \Phi^{-1}d. \quad (9)$$

The vital question is how we can be sure that the interpolation matrix Φ is nonsingular. Fortunately, the previous results due to Micchelli [19, 20] have shown that for n distinct points $x_1, x_2, \dots, x_n \in R^m$, a large class of radial basis functions may guarantee the nonsingularity of Φ . The term radial basis function is derived from the fact that these functions are radially symmetric; that is, each node produces an identical output for inputs that lie at a fixed radial distance from the center. Among those radial basis functions are the Gaussian, multiquadratic, inverse multiquadratic, thin plate splines, cubic splines, and linear splines. The later is used in this work.

The above RBFN, used in this study, is not unique. Several other RBFNs can be found in the literature such as regularization network [19] and generalized RBFN [19, 27]. However, with the above RBFN, no trial-and-error procedure is required, and the computational aspects are simple and straight forward. It is also more suitable for this study since the number of data pairs involved in the training phase is not very large. Furthermore, in the training phase, one can expect zero training error, which is usually difficult (if not impossible) to achieve, at least by the other networks discussed in this work; that is, ANFIS and G-FIS.

3. ANFIS-Based Fuzzy Networks

Fuzzy inference systems (FISs) are usually used as mathematical tools for approximating ill-defined nonlinear functions.

They can import qualitative aspects of human knowledge and reasoning process by data sets without employing precise quantitative analysis using the following five functional components, as shown in Figure 2 [28].

- (i) A rule base containing a number of fuzzy if-then rules.
- (ii) A database defining the membership functions of the fuzzy sets.
- (iii) A decision-making unit as the inference engine.
- (iv) A fuzzification interface which transforms crisp inputs to linguistic variables.
- (v) A defuzzification interface converting fuzzy outputs to crisp output.

As mentioned above, fuzzy inference systems (FISs) are composed of a set of *If-Then* rules. A typical first-order (FO) Sugeno fuzzy model for the problem under consideration has the following rule set:

$$R^l : \text{ If } S_m \text{ is } A_i, F_m \text{ is } B_j, D_m \text{ is } C_k, \quad (10)$$

$$\text{Then } f_l = p_l S_m + q_l F_m + r_l D_m + t_l,$$

where R^l ($l = 1, 2, \dots, N$) denotes the l th rule number, A_i , B_j , and C_k are the linguistic terms of the inputs, p_l , q_l , r_l , and t_l are the consequent parameters. Similar to the work of [12, 13] (for the sake of comparison), we choose $i, j, k = 1, 2, 3$, so that the number of rules N is 27.

The overall output y of this first order (FO) Sugeno fuzzy system (10) is

$$y = \frac{\sum_{l=1}^N \tau_l f_l}{\sum_{l=1}^N \tau_l} = \frac{\sum_{l=1}^N \tau_l (p_l S_m + q_l F_m + r_l D_m + t_l)}{\sum_{l=1}^N \tau_l}, \quad (11)$$

where τ_l is the firing strength of R^l , which is defined as

$$\tau_l = A_i(S) \times B_j(F) \times C_k(D), \quad (12)$$

where $i, j, k = 1, 2, 3$.

A zero-order (ZO) Sugeno fuzzy model has the following form

$$R^l : \text{ If } S_m \text{ is } A_i, F_m \text{ is } B_j, D_m \text{ is } C_k, \quad (13)$$

$$\text{Then } f_l = u_l,$$

where u_l , $l = 1, 2, \dots, N$ is constant. The overall output of this zero-order model is

$$y = \frac{\sum_{l=1}^N \tau_l f_l}{\sum_{l=1}^N \tau_l} = \frac{\sum_{l=1}^N \tau_l u_l}{\sum_{l=1}^N \tau_l}. \quad (14)$$

In this and the coming sections, Gaussian membership functions are used to define the membership grade of the input variables. A Gaussian membership function is defined by

$$\mu_{ij}(x_i) = \exp\left\{-\frac{1}{2}\left(\frac{x_i - c_j}{\sigma_j}\right)^2\right\}, \quad (15)$$

FIGURE 2: The structure of the fuzzy inference system used in this work.

where x_i , $i = 1, 2, 3$ are the input variables, that is, S_m , F_m , D_m , c_j , and σ_j are, respectively, the center and spread (width) of the j th membership function, $j = 1, 2, 3$.

With the above structure of the FO-FIS, the number of parameters in the premise part is 18 and in the consequent part is 4×27 , so the total number of parameters is 126. With respect to ZO-FIS, the number of parameters in the premise part is also 18 and in the consequent part is 1×27 , so the total number of parameters is 45. In this work, the two FISs are tuned and optimized by two learning algorithms; the ANFIS as discussed in this section and the genetic algorithms in Section 4.

Fundamentally, ANFIS is about taking an initial fuzzy inference system (FIS) and tuning it with back propagation algorithm based on the collection of input-output data. With tuning, we mean optimal selection of the parameters of the input membership functions and parameters of the consequent part of the FIS. ANFIS may use the back propagation (BP) or the hybrid learning algorithm (HL) to identify the FIS parameters. In the hybrid learning method (HL), a combination of least squares and back propagation gradient descent methods is used for training FIS membership function parameters to model a given set of input/output data. The coming two subsections discuss the architecture and the learning algorithms of ANFIS.

3.1. ANFIS Architecture. For simplicity, we assume that the fuzzy inference system under consideration has two inputs x and y and one output z . For a first-order (FO) Sugeno fuzzy model [17], a common rule set with two fuzzy if-then rules is the following

$$\text{Rule 1 : If } x \text{ is } A_1 \text{ and } y \text{ is } B_1, \quad (16)$$

$$\text{Then } f_1 = p_1x + q_1y + r_1.$$

$$\text{Rule 2 : If } x \text{ is } A_2 \text{ and } y \text{ is } B_2, \quad (17)$$

$$\text{Then } f_2 = p_2x + q_2y + r_2.$$

Figure 3 illustrates the reasoning mechanism for this Sugeno model; the corresponding equivalent ANFIS architecture is as shown in Figure 4, where nodes of the same layer have similar functions, as described next. Here, we denote the output of the i th node in the layer l as $O_{l,i}$.

Layer 1. Every node i in this layer is an adaptive node with a node function

$$\begin{aligned} O_{1,i} &= \mu_{A_i}(x), \quad \text{for } i = 1, 2, \text{ or} \\ O_{1,i} &= \mu_{B_{i-2}}(y), \quad \text{for } i = 3, 4, \end{aligned} \quad (18)$$

where x (or y) is the input to node i and A_i (or B_{i-2}) is a linguistic label (such as “small” or “large”) associated with this node. In other words, $O_{1,i}$ is the membership grade of a fuzzy set (μ_{A_i} or $\mu_{B_{i-2}}$) and it specifies the degree to which the given input x (or y) satisfies the quantifier A_i (or B_{i-2}). Here, the membership function for the inputs x and y can be any appropriate parameterized membership function such as Gaussian, triangular, trapezoidal, or any other appropriate function. Parameters in this layer are referred to as premise parameters.

Layer 2. Every node in this layer is a fixed node labeled Π , whose output is the product of all the coming signals:

$$O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \quad i = 1, 2. \quad (19)$$

Each node represents the firing strength of a rule. In general, any other T-norm operators that perform fuzzy AND can be used as the node function in this layer.

Layer 3. Every node in this layer is a fixed node labeled S . The i th node calculates the ratio of the i th rule’s firing strength to the sum of all rules’ firing strengths:

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2. \quad (20)$$

The outputs of this layer are called normalized firing strengths.

Layer 4. Every node i in this layer is an adaptive node with a node function:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i), \quad (21)$$

where $\bar{w}_i$ is a normalized firing strength from layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set of this node. Parameters in this layer are referred to as the consequent parameters.

FIGURE 3: A two-input first-order Sugeno fuzzy model with two rules.

FIGURE 4: ANFIS architecture.

Layer 5. The single node in this layer is a fixed node labeled Σ , which computes the overall output as the summation of all incoming signals:

$$O_{5,1} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}. \quad (22)$$

The above statements conclude the ANFIS architecture which is equivalent to a Sugeno FO fuzzy model. If the consequent part contains only r_i (p_i and q_i are set to zeros), $i = 1, 2$, then we get the ZO Sugeno fuzzy model. For comparison purposes, the study examines the two models.

3.2. The ANFIS Learning Algorithms. The task of the learning algorithm is to modify all the modifiable parameters of the adaptive layers. In this study, two learning algorithms are

considered; the back propagation (BP) and the hybrid learning (HL) algorithms. In both cases, the initial FIS is generated from the training data set, using a grid partition on the data (no clustering). The central part of BP concerns how to recursively obtain a gradient vector in which each element is defined as the derivative of an error measure with respect to a parameter. This is done by means of the chain rule, a basic formula for differentiating composite functions, that is covered in every textbook on elementary calculus. Once the gradient is obtained, a number of derivative-based optimization and regression techniques are available for updating the parameters like gradient methods, steepest descent, Newton's methods, conjugate gradient methods, and nonlinear least-squares. In particular, if we use the gradient vector in a simple steepest descent method, the resulting learning paradigm is referred to as the BP learning rule [17]. The BP is used

to modify the consequent parameters with the forward pass training method. The training method optimizes the consequent parameters with the premise parameters fixed. The estimation method of the optimum modulation is done using the following formula:

$$E = \frac{1}{2}(R_m - F)^2, \quad (23)$$

where R_m is the expected surface roughness, F is the ANFIS output, and E is the mean square error value. When E reaches the convergence condition, it will produce the inference results. Otherwise, the consequent parameters are fixed and the premise parameters are modified with the BP method.

The second training method considered in this study is based on hybrid learning (HL) algorithm which has been proposed by Jang [29]. The algorithm consists of a combination of the least square estimator (LSE) and the gradient descent (GD) method. More specifically, in the forward pass of the hybrid learning algorithm, node outputs go forward until Layer 4 and the consequent parameters are identified by the least-square method. In the backward pass, the error signals propagate backward and the premise parameters are updated by the GD method. The LSE is used to modify the consequent parameters with the forward pass training method. The training method optimizes the consequent parameters with the premise parameters fixed. When E in (23) reaches the convergence condition, it will produce the inference results. Otherwise, the consequent parameters are fixed and the premise parameters are modified with the GD method. The GD method is a backward pass training method which adjusts the optimum premise parameters. These optimum premise parameters are modified corresponding to the fuzzy sets in the input domain. After the new parameters of the premise part are obtained, the output of ANFIS is calculated again by employing the consequent parameters found by the forward pass training method. Table 2 summarizes the activities in each pass.

The hybrid learning (HL) algorithm causes the error E to converge to the convergence condition. The reader is referred to [29] for further mathematical derivation of this HL algorithm. Previous results have proven that this hybrid learning (HL) algorithm is highly efficient for optimally tuning Sugeno FISs [12, 17, 29].

Here, we have examined four ANFIS networks. They are FO and ZO Sugeno FISs as defined in (11) and (14), respectively; each of them was trained by the BP and HL algorithms. The four ANFIS networks have been initiated using grid partition to the input variables, that is, in the antecedent part, the membership functions are equally spaced inside the range of each input variable. The consequent parameters have been initiated with zeros. The coming section discusses tuning the FO and ZO Sugeno FISs using an alternative method, the genetic algorithms (GAs).

4. Genetically Evolved Fuzzy Inference Systems (G-FISs)

Genetic algorithms (GAs) are derivative-free stochastic optimization methods based loosely on the concepts of natural

TABLE 2: Two passes in the hybrid learning procedure for ANFIS.

	Forward pass	Backward pass
Premise parameters	Fixed	Gradient descent
Consequent parameters	Least-square estimator	Fixed
Signals	Node outputs	Error signals

selection and evolutionary processes. Their popularity can be attributed to their freedom from dependence on functional derivatives, and they are less likely to get trapped in local minima, which inevitably are present in any practical optimization application (including ANFIS). Eventually, GAs can be used to determine the optimal parameters of a fuzzy system given some optimality criteria.

The solution of an optimization problem with GAs begins with a set of potential solutions (FISs) or chromosomes (usually in the form of bit strings) that are randomly selected. The entire set of these chromosomes comprises a population. The chromosomes evolve during several iterations or generations. New generations (offspring) are generated utilizing the crossover, mutation, and elitism technique. Crossover involves splitting two chromosomes and then combining one-half of each chromosome with the other pair. Mutation involves flipping a single bit of a chromosome. Elitism is a policy of always keeping a certain number of best members when each new population is generated. The chromosomes are then evaluated employing a certain fitness criterion, and the best ones are kept, while the others are discarded. This process repeats until one chromosome has the best fitness and is taken as the optimum solution of the problem. Figure 5 is a schematic diagram illustrating how a fuzzy system can be trained using GAs. A comprehensive review about GAs can be found in [30].

In this section, the values of the premise and consequent parameters of FO and ZO FISs are learned by minimizing the root mean squared error (RMSE) defined by

$$J = \left[\sum_{m=1}^{\alpha} \frac{(R_m - F)^2}{\alpha} \right]^{1/2}, \quad (24)$$

where α denotes the number of training data, R_m is the actual experimental surface roughness (training data sets), and F denotes the predicted surface roughness, which is an output of FIS. This performance index has been also adapted in [13].

Because GA endeavors to maximize the fitness function, the fitness function of each gene (chromosome) is calculated as follows:

$$F = \frac{1}{1 + J}, \quad (25)$$

where J is the performance index defined in (24) and 1 is introduced at the denominator to prevent the fitness function from becoming infinitely large.

Based on the aforementioned concepts, a genetic algorithm for maximization problems can be described as follows [17].

FIGURE 5: Genetic fuzzy inference system (G-FIS).

Step 1. Initialize a population with randomly generated individuals and evaluate the fitness value of each individual using (25).

Step 2.

- (a) Select two members from the population with probabilities proportional to their fitness values.
- (b) Apply crossover with a probability equal to the crossover rate.
- (c) Apply mutation with a probability equal to the mutation rate.
- (d) Repeat from (a) to (c) until enough members are generated to form the next generation.

Step 3. Evaluate each member of the new generation using the fitness function (25).

Step 4. Repeat steps 2 and 3 until a stopping criterion is met.

Here, the GA performs only parameter learning of the fuzzy model. The structure of the genetic fuzzy inference systems (G-FIS) is completely determined in advance by determining the number of memberships of each input variable and choosing the function of the consequent part whether it is ZO or FO. That is, the number of rules is the product of the memberships of the input variables (full interconnection between Layers 2 and 3, Figure 4). Gaussian membership functions (15) have been utilized for the input variables. As it is the case with ANFIS, the number of parameters to be determined (optimally tuned) is 126 for the FO-FIS and 45 for the ZO-FIS. Similar to the work of Ho et al. [13], the data base of the antecedent and consequent parameters has been randomly initiated.

The GA is used to tune the membership functions at the precedent part of the fuzzy rules and the consequent part (whether it is ZO or FO) within prespecified ranges. These ranges determine the search space of the optimization problem. Choosing large ranges, increases the search space, and the optimal solution may not be reachable. Choosing low ranges, restricts the search space which may lead to an under-determined optimization problem. A compromise has to be found.

Otherwise, the critical parameters of the GA are the size of the population, crossover rate, mutation rate, number of iterations, that is the number of generations (the stopping criterion used in this work), and so forth. These parameters are problem dependent [31]. A parametric study is introduced in Section 6.3 to determine the best possible numeric values of these parameters. The obtained values are heavily depending on trial and error.

5. Experimental Data Sets

In this study, we use the experimental data published in [12, 13]. A high-speed steel (HSS) four-flute end milling cutter with a diameter 3/4" was used to machine 6061 aluminum alloy. Spindle speed, feed rate, and depth of cut were selected as the machining parameters to analyze their effect on surface roughness. A total of 48 sets were utilized as the training data for all the above algorithms. Among them, the settings of spindle speed include 750, 1000, 1250, and 1500 rpm; those of the feed rate include 6, 12, 18, and 24 ipm; and the depth of cut is set at 0.01, 0.03, and 0.05 in. They are listed in Table 3. The testing (validation) data sets are listed in Table 4. They are 24 sets which implement different feed rate sittings of 9, 15, and 21 ipm. The settings for the other parameters are the same as those of the training sets.

The experimental training and testing data have been normalized in order to make them suitable for the training and validation processes [5]. This was done by mapping each term to a value between 0 and 1 by simply dividing each column in Tables 2 and 3 by the corresponding maximum value. This approach avoids the complications of other normalization criterion which can be found in [4], and the predicted surface roughness value can be easily transformed back to its true value.

6. Results and Discussion

In this section, results are demonstrated and discussed. The training and testing RMSE of the seven examined networks are summarized in Table 5. Findings show that the RBFN has achieved the least training error (RMSE = 0.0) and least testing error (RMSE = 0.0295). ANFIS network of type ZO Sugeno fuzzy model trained with BP exhibited the worst predicting results (RMSE = 0.1069). The two G-FIS networks give mixed signals about their prediction accuracy.

Table 6 gives more details about the prediction accuracy of three selected networks; the RBFN, the FO Sugeno fuzzy model trained by ANFIS using HL algorithm, and the ZO Sugeno fuzzy model trained by GA. The coming subsections demonstrate the performances of the three types of networks implemented in this study.

6.1. Performance of The RBFN. The RBFN described in Section 2 has been trained using the experimental data sets listed in Table 3. As mentioned earlier, we have used the linear splines radial basis function, Figure 6. This type of radial basis functions does not require any trial-and-error procedure, which is necessary in Gaussian RBFN to determine

TABLE 3: Experimental results for training data (Lo [12] and Ho et al. [13]).

m	Experimental results			
	S_m	F_m	D_m	R_m
1	750	6	0.01	65
2	750	6	0.03	63
3	750	6	0.05	72
4	750	12	0.01	144
5	750	12	0.03	102
6	750	12	0.05	94
7	750	18	0.01	185
8	750	18	0.03	147
9	750	18	0.05	121
10	750	24	0.01	187
11	750	24	0.03	170
12	750	24	0.05	172
13	1000	6	0.01	58
14	1000	6	0.03	78
15	1000	6	0.05	62
16	1000	12	0.01	130
17	1000	12	0.03	84
18	1000	12	0.05	92
19	1000	18	0.01	138
20	1000	18	0.03	124
21	1000	18	0.05	86
22	1000	24	0.01	163
23	1000	24	0.03	153
24	1000	24	0.05	142
25	1250	6	0.01	50
26	1250	6	0.03	63
27	1250	6	0.05	71
28	1250	12	0.01	101
29	1250	12	0.03	99
30	1250	12	0.05	85
31	1250	18	0.01	115
32	1250	18	0.03	92
33	1250	18	0.05	95
34	1250	24	0.01	155
35	1250	24	0.03	109
36	1250	24	0.05	121
37	1500	6	0.01	37
38	1500	6	0.03	56
39	1500	6	0.05	56
40	1500	12	0.01	88
41	1500	12	0.03	82
42	1500	12	0.05	94
43	1500	18	0.01	119
44	1500	18	0.03	87
45	1500	18	0.05	104
46	1500	24	0.01	119
47	1500	24	0.03	103
48	1500	24	0.05	109

TABLE 4: Experimental results for testing data (Lo [12] and Ho et al. [13]).

m	Experimental results			
	S_m	F_m	D_m	R_m
1	750	9	0.01	109
2	750	9	0.05	95
3	750	15	0.03	122
4	750	15	0.05	104
5	750	21	0.01	178
6	750	21	0.03	163
7	750	21	0.05	150
8	1000	9	0.01	92
9	1000	15	0.03	108
10	1000	21	0.01	149
11	1000	21	0.03	145
12	1000	21	0.05	112
13	1250	15	0.01	106
14	1250	15	0.03	96
15	1250	21	0.01	125
16	1250	21	0.03	100
17	1250	21	0.05	105
18	1250	9	0.03	73
19	1500	15	0.01	106
20	1500	15	0.03	83
21	1500	15	0.05	99
22	1500	21	0.01	118
23	1500	21	0.03	102
24	1500	21	0.05	113

TABLE 5: A summary of the training and testing results.

	RBFN	ANFIS			G-FIS		
		Learning method	ZO	FO	Initiation method	ZO	FO
Training error (RMSE)	0.0	BP	0.1067	0.0449	Random	0.0485	0.0409
		HL	0.0315	0.0016			
Testing error (RMSE)	0.0295	BP	0.1069	0.0542	Random	0.0395	0.0488
		HL	0.0434	0.0339			

the suitable slopes of the Gaussian functions [19, 20]. Since the number of training data sets is 48, the resulting interpolation matrix Φ defined in (5) is 48×48 symmetric positive definite. Elements of the main diagonal are zeros. The weight vector has been computed using (9). Values of w_i , $i = 1, 2, \dots, 48$ are listed in Table 7.

When we examined the inverse problem, that is,

$$J = \left[\sum_{m=1}^{\alpha} \frac{(\Phi w - R_m)^2}{\alpha} \right]^{1/2}, \quad \alpha = 48, \quad (26)$$

we got $J = 0$. Afterward, the network has been examined using the testing data of Table 4. The new interpolation matrix Φ has been computed using the testing data as the input vectors. The training input data sets are used as the centers. The resulting Φ is 24×48 . This matrix and the weight vector in Table 7 have been used to compute the predicted

surface roughness and then compared with testing counterpart. We obtained RMSE = 0.0295. Errors of the 24 sets of testing data after training are plotted in Figure 7, and the scatter diagram is given in Figure 8. The later shows that the predicted data is distributed in a narrow range around the 45° line. This means that the proposed RBFN can capture the nature of the experimental data with accuracy close to 97%.

Referring to Tables 5 and 6, results show that the prediction error of the RBFN is less than that of the two G-FIS networks (FO and ZO), despite the fact that genetic algorithms are often seen as global optimizers. This may be explained as follows. The G-FISs are optimal under the conditions of using certain parameter settings (population size, number of generations, etc) and within the specified ranges for the fuzzy system parameters (slope, width of the Gaussian membership functions, and parameters of the consequent part). For other parameter settings and tuning ranges, other FISs

TABLE 6: A comparison of measured and predicted surface roughness of the test data.

Test no.	Parameters			Measured	R_m		
	S_m	F_m	D_m		RBFN predicted	ANFIS (HL,FO) predicted	GA (ZO) predicted
1	750	9	0.01	109	103.98	95.78	109.18
2	750	9	0.05	95	82.96	82.36	73.63
3	750	15	0.03	122	124.15	116.79	122.78
4	750	15	0.05	104	105.94	96.93	112.12
5	750	21	0.01	178	183.14	187.44	176.66
6	750	21	0.03	163	159.31	161.19	161.39
7	750	21	0.05	150	143.67	148.65	149.19
8	1000	9	0.01	92	91.64	87.13	95.89
9	1000	15	0.03	108	105.85	96.46	106.78
10	1000	21	0.01	149	153.56	149.17	153.48
11	1000	21	0.03	145	135.09	141.20	136.73
12	1000	21	0.05	112	116.47	113.28	127.24
13	1250	15	0.01	106	112.68	104.36	112.46
14	1250	15	0.03	96	94.96	99.06	93.34
15	1250	21	0.01	125	133.28	135.37	127.79
16	1250	21	0.03	100	103.12	98.65	112.17
17	1250	21	0.05	105	105.84	108.58	107.48
18	1250	9	0.03	73	78.73	77.50	70.95
19	1500	15	0.01	106	103.49	101.79	105.17
20	1500	15	0.03	83	86.07	86.17	89.61
21	1500	15	0.05	99	97.27	98.47	92.45
22	1500	21	0.01	118	118.87	120.44	120.64
23	1500	21	0.03	102	93.04	94.37	105.34
24	1500	21	0.05	113	104.50	106.92	101.99
RMSE%					2.95%	3.39%	3.95%

FIGURE 6: Linear splines radial basis function, $\phi(r) = r$.

are obtained and probably better results may be achieved. However, a trial-and-error procedure should be followed to select them. For instance, determining the suitable range

for each parameter of the FIS (126 parameters for FO and 45 parameters for ZO) is a tedious and time-consuming task.

Also, RBFN has showed better results than that of FO Sugeno fuzzy model tuned by ANFIS using HL algorithm, despite the powerfulness of this algorithm as discussed by earlier works [12, 17, 29]. In general, this may be referred to the local minima problems of derivative-based optimization schemes.

6.2. Performance of ANFIS Networks. In this subsection, we discuss the performance of four ANFIS networks. They are the FO and ZO Sugeno fuzzy models trained with BP and HL algorithms. As mentioned in Section 3, they have been initiated using grid partition to the input variables, that is, equally spaced membership functions in the antecedent part. The consequent parameters have been initiated with zeros. In training the four ANFIS networks, the training data sets in Table 3 were used to conduct 500 cycles of learning. The prediction RMSE of the training and testing is shown in Table 5. The ZO Sugeno fuzzy model trained with BP algorithm has demonstrated the worst performance; RMSE = 0.1067. Better

TABLE 7: Values of the members of the weight vector.

$w_1 = 0.6001$	$w_{17} = 0.5471$	$w_{33} = -0.0011$
$w_2 = 0.4402$	$w_{18} = -0.3502$	$w_{34} = -0.5903$
$w_3 = 0.0998$	$w_{19} = 0.4575$	$w_{35} = 0.5141$
$w_4 = -0.3603$	$w_{20} = -0.2180$	$w_{36} = 0.0060$
$w_5 = -0.0066$	$w_{21} = 0.8256$	$w_{37} = 0.5385$
$w_6 = 0.0790$	$w_{22} = 0.2043$	$w_{38} = 0.0499$
$w_7 = -0.6617$	$w_{23} = -0.4272$	$w_{39} = 0.4842$
$w_8 = -0.1991$	$w_{24} = -0.0579$	$w_{40} = -0.1173$
$w_9 = -0.0474$	$w_{25} = 0.0928$	$w_{41} = -0.0202$
$w_{10} = -0.0044$	$w_{26} = 0.0781$	$w_{42} = -0.3623$
$w_{11} = 0.0820$	$w_{27} = -0.3322$	$w_{43} = -0.4317$
$w_{12} = -0.4343$	$w_{28} = -0.0664$	$w_{44} = 0.0590$
$w_{13} = 0.2373$	$w_{29} = -0.6556$	$w_{45} = -0.2374$
$w_{14} = -0.6222$	$w_{30} = 0.0913$	$w_{46} = 0.5349$
$w_{15} = 0.3334$	$w_{31} = 0.4232$	$w_{47} = 0.0204$
$w_{16} = -0.5701$	$w_{32} = 0.3740$	$w_{48} = 0.2359$

FIGURE 7: The prediction error of the testing data, RBFN.

results with BP algorithm have taken place when FO Sugeno fuzzy model has been used; RMSE = 0.0449.

Here, in order to save space, we only present the learning results of the ANFIS network which has achieved the best predicting performance. This network is the FO Sugeno model and has been trained using the HL algorithm; RMSE = 0.0339. The evolution of the RMSE during the learning phase is shown in Figure 9(a). As it can be noticed, this error has reached to a steady-state value of 0.0016 after nearly 300 epochs. Figure 9(b) shows the resulting error of the testing data. The numerical value of the antecedent part given in Tables 8 and 9 gives the values of the parameters of the consequent part.

Figure 10 shows the membership functions before and after training. In this Figure, the initial and the final membership functions of the spindle speed and feed rate have experienced relatively large changes in comparison with the membership functions of the depth of cut. This remark indicates that the depth of cut has the least impact on the surface roughness of the end milling process.

FIGURE 8: Scatter diagram of the testing and predicted data, RBFN.

In Lo's work [12], the same experimental data in Tables 3 and 4 has been used to train and test two ANFIS networks. The two networks are FO Sugeno models trained with the HL algorithm, and similar to this work, the number of rules is 27. In the first network, he utilized triangular membership functions and trapezoidal membership functions in the second network. The average error has been used to compare the prediction accuracy of the two networks. The network which used triangular membership functions has performed better. The author of this paper has used Table 3 in Lo's work [12] to compute the prediction RMSE when triangular membership function is used. The RMSE there is 0.0347, which is very close to FO Sugeno model when Gaussian membership functions trained by HL as we did in this work. It means that the Gaussian and triangular membership functions can equally capture the nature of the experimental data.

The conclusion that can be withdrawn from the above results is that the FO Sugeno model trained with ANFIS using HL algorithm can achieve accuracy of around 96.6% whether triangular or Gaussian membership functions have been used.

However, this conclusion cannot be generalized. The author of reference [32] uses ANFIS to predict the surface roughness in turning process and compares his results with a proposed response surface method (RSM). Similar to this work, the input parameters are spindle speed, federate, and depth of cut. The achieved prediction accuracy of the ANFIS network is better than the proposed RSM. He examined two kinds of membership functions, the triangular and Gaussian membership functions. According to his results, better results have been achieved with triangular membership functions.

A more recent article is presented in [33]. This reference uses the same experimental data (training and testing data) as used in this work. The authors implement an ANFIS network with Gaussian membership functions for the inputs, that is, spindle speed, federate, and depth of cut. The training algorithm, however, is different from the presented here. It is called leave-one-out cross-validation algorithm and used to obtain an optimal ANFIS network. Then, they use a "top-down" rule reduction approach to decrease the number of

TABLE 8: The optimal premise parameters of ANFIS FO Sugeno model trained with HL algorithm.

Input variable	Optimal premise parameters	Input variable	Optimal premise parameters	Input variable	Optimal premise parameters
S	$C_{S_1} = 0.5361$	F	$C_{F_1} = 0.2185$	D	$C_{D_1} = 0.1999$
	$\sigma_{S_1} = 0.0947$		$\sigma_{F_1} = 0.0957$		$\sigma_{D_1} = 0.1696$
	$C_{S_2} = 0.7516$		$C_{F_2} = 0.5487$		$C_{D_2} = 0.6007$
	$\sigma_{S_2} = 0.0347$		$\sigma_{F_2} = 0.0958$		$\sigma_{D_2} = 0.1670$
	$C_{S_3} = 0.9715$		$C_{F_3} = 0.9518$		$C_{D_3} = 1.0000$
	$\sigma_{S_3} = 0.0923$		$\sigma_{F_3} = 0.2032$		$\sigma_{D_3} = 0.1696$

TABLE 9: The consequent parameters of ANFIS FO Sugeno model trained with HL algorithm.

Rule no.	p_i	q_i	r_i	t_i
1	-0.3475	0.1141	0.09476	0.4712
2	0.4908	0.01827	0.03587	0.06537
3	-0.5326	0.082	0.3158	0.3164
4	-0.142	0.6265	0.1154	0.508
5	-0.8807	0.3169	0.3492	0.5938
6	0.3319	-0.05292	0.1754	0.1892
7	-1.996	-0.05069	0.4055	1.978
8	-0.3329	0.3839	0.2737	0.5203
9	-1.28	1.041	0.2528	0.2626
10	0.2578	0.07892	0.05012	0.2143
11	0.2012	0.08151	0.1871	0.3131
12	0.2403	0.05555	0.1689	0.1762
13	0.673	-1.131	0.1542	0.7394
14	0.4164	0.01532	0.1737	0.3085
15	0.1619	-0.5922	0.2246	0.231
16	-1.471	5.782	-0.4443	-2.232
17	0.008048	0.6415	-0.0189	-0.08517
18	-0.4688	2.999	-0.7868	-0.7847
19	-0.215	0.09277	0.07536	0.367
20	0.1457	0.03197	0.06851	0.1165
21	-0.3208	0.07498	0.2976	0.2996
22	-0.3354	0.3712	0.1312	0.5879
23	-0.4769	0.2513	0.3433	0.588
24	0.1199	0.1176	0.1524	0.1659
25	0.3456	-0.0816	0.07721	0.3656
26	-0.1232	0.3988	0.1172	0.1992
27	0.008071	0.06087	0.2541	0.2582

rules from 27 to 20. Although the achieved RMSE = 0.0040 in training and 0.0319 in testing outperform the work of Lo [12], the proposed RBFN in this work still performs better; see Table 5.

6.3. Performance of Genetic Fuzzy Inference Systems (G-FIS).

As the performance of a GA depends on its parameters, a parametric study has been carried out to determine the optimal set of GA parameters. These parameters are the population size, the number of generations, the number of bits of each variable, the crossover probability, and the mutation

FIGURE 9: Evolution of the RMSE during the learning phase of FO Sugeno FIS using ANFIS trained with HL algorithm (a) and the resulting prediction error of testing data (b).

probability. They are problem dependent and should be selected carefully in order to achieve *good* results. We have started the parametric study from the parameter set used in [13]; that is, population size is 200, the number of generation is 200, the crossover rate is 0.9, and the mutation rate is 0.1. The number of bits which represent each variable is not mentioned there. Here, the parametric study consists of five stages and has been performed only on the ZO Sugeno fuzzy model.

TABLE 10: Ranges of the premise and consequent parameters of FO Sugeno fuzzy model.

Parameters	Range
Premise parameters	$C_{S_1}, C_{F_1}, C_{D_1}$
	$C_{S_2}, C_{F_2}, C_{D_2}$
	$C_{S_3}, C_{F_3}, C_{D_3}$
	$\sigma_{S_i}, \sigma_{F_i}, \sigma_{D_i}, (i = 1, 2, 3)$
Consequent parameters	$p_l, q_l, r_l, t_l, (l = 1, 2, \dots, 27)$

TABLE 11: Optimal premise and consequent parameters of ZO Sugeno model tuned by GA.

Input variables	Optimal premise parameters	Optimal consequent parameters
S	$C_{S_1} = 0.3000$	$u_1 = -1.0000$
	$\sigma_{S_1} = 0.1312$	$u_2 = -0.5000$
	$C_{S_2} = 0.2868$	$u_3 = 0.9998$
	$\sigma_{S_2} = 0.4438$	$u_4 = 0.1261$
	$C_{S_3} = 0.9000$	$u_5 = 0.7499$
	$\sigma_{S_3} = 0.2250$	$u_6 = 0.9993$
		$u_7 = 1.0000$
		$u_8 = 1.0000$
		$u_9 = 1.0000$
F	$C_{F_1} = 0.2508$	$u_{10} = 0.0000$
	$\sigma_{F_1} = 0.1322$	$u_{11} = 0.8732$
	$C_{F_2} = 0.3500$	$u_{12} = 0.0000$
	$\sigma_{F_2} = 0.3500$	$u_{13} = 0.7500$
	$C_{F_3} = 0.7250$	$u_{14} = -0.3750$
	$\sigma_{F_3} = 0.4750$	$u_{15} = 0.3750$
		$u_{16} = 1.0000$
		$u_{17} = 0.9375$
		$u_{18} = 0.5313$
D	$C_{D_1} = 0.2009$	$u_{19} = -0.1875$
	$\sigma_{D_1} = 0.5708$	$u_{20} = 0.6250$
	$C_{D_2} = 0.7051$	$u_{21} = -0.8139$
	$\sigma_{D_2} = 0.2875$	$u_{22} = 0.3281$
	$C_{D_3} = 0.8375$	$u_{23} = 0.2500$
	$\sigma_{D_3} = 0.1078$	$u_{24} = 1.0000$
		$u_{25} = 0.6562$
		$u_{26} = 0.5000$
		$u_{27} = 0.5000$

In the first stage, crossover probability is varied from 0.8 to 0.99, keeping the other parameters, namely, mutation probability, population size, number of bits, and maximum number of generations fixed to 0.1, 200, 16, and 200, respectively. Like the works of [10, 13], the best result is observed with crossover probability of 0.90. In the second stage, mutation probability is changed in the range of 0.001 to 0.1, keeping crossover probability, population size, number of bits, and maximum generation fixed to 0.90, 200, 16, and 200, respectively. The best result is found with the mutation probability

of 0.016. In the third stage, crossover probability, mutation probability, number of bits, and population size are kept fixed to 0.90, 0.016, 16, and 200, respectively, and the maximum number of generation is varied from 200 to 600. The best result takes place when the number of generation is 500. After this number of generation, no improvement has been noticed in the fitness function (25). In the fourth stage, the number of bits has been changed in the range of 16–64. The crossover probability, mutation probability, population size, and the maximum number of generation are kept to 0.90, 0.016, 200, and 500, respectively. It has been noticed that the number of bits has little or no impact on the results. So, we have selected the number of bits to be 32. In the last stage of the parametric study, the population size has been changed from 50 to 200, after keeping the other parameters, namely, crossover probability, mutation probability, number of bits, and maximum number of generation fixed to 0.90, 0.016, 32, and 500, respectively. The best result is observed when the population size is 80. Thus, the following GA parameters are found to give the best results during the GA-based training of ZO Sugeno fuzzy inference system:

- (i) single point crossover with a probability of 0.90,
- (ii) bitwise mutation with a probability of 0.016,
- (iii) maximum number of generation is 500,
- (iv) number of bits which represent each variable of the FIS to the chromosome is 32,
- (v) population size is 80.

In this work, both the ZO and FO Sugeno fuzzy models have been trained using these GA parameters. So, the number of bits which constitute the chromosome (gene or a possible solution) is $32 \times 126 = 4032$ bits for the FO Sugeno fuzzy model. With respect to the ZO Sugeno fuzzy model, the chromosome length is $32 \times 45 = 1440$ bits. For the two models, the number of examined solutions is the number of generations multiplied by the number of populations, that is 40,000 possible solutions. Of course, the optimal solution is the best one of these 40,000 solutions.

In order to complete the definition of the GA optimization problem, a range of variation should be specified for each parameter of the FIS. Ranges of variation of the FO Sugeno fuzzy model are listed in Table 10. They are similar to those used in [13] in order to optimize a FO Sugeno model. With respect to ZO Sugeno model, the same premise parameters ranges have been used. The consequent parameters of the ZO Sugeno fuzzy model, $u_i, i = 1, 2, \dots, 27$, have been also trained between -1 and 1 . These ranges have

FIGURE 10: Membership functions of FO Sugeno model before training (a) and after training by ANFIS using HL algorithm.

been selected intentionally in order to simulate a similar optimization problem to that which is found in [13] where the authors have used the same experimental data listed in Tables 3 and 4 for training and testing, respectively. However, different parameter settings for the GA are used here.

As it can be noticed from Table 5, the ZO Sugeno model (RMSE = 0.0395) has achieved better performance than the FO Sugeno model (RMSE = 0.0488). Surprisingly, the ZO G-FIS has achieved less testing error (RMSE = 0.0395) than the training one (RMSE = 0.0485). This remark has been noticed from several simulation experiments for both FO and ZO (but not consistently). The remark that should be noted here is that for the same simulation experiment, several runs do not always result in the same (exact) findings. This can be attributed to the stochastic nature of GAs. Results presented here are the best obtained results.

Parameters of the optimally tuned ZO Sugeno fuzzy model are shown in Table 11. In order to save space, parameters of the optimally tuned FO Sugeno fuzzy model are not listed here.

Membership functions of the two tuned Sugeno fuzzy models after tuning are given in Figure 11. They show large

changes in the membership functions of all the three end milling parameters. This remark is notable for the FO model relative to the corresponding one which has been trained with ANFIS using HL algorithm, Figure 10. This may be referred to the nature of the tuning process of GAs. They are derivative-free algorithms and, more importantly, tuning the membership functions is done randomly. Nevertheless, referring to Figure 11, the least changes have taken place in the membership functions of the depth of cut. This remark gives the impression that the depth of cut has the least influence on the surface roughness. It was also a concluding remark in the work of [12, 13].

As mentioned earlier, in the work of Ho et al. [13], the authors used the GAs to optimally tune a FO-FIS which, similar to this work, uses Gaussian membership functions for the three inputs. The number of rules (27 rules), the fitness function (24), and the tuning ranges (Table 10) are also similar to this work. The optimal FIS there has produced a prediction RMSE of 0.0332 (computed by the author of this work from Table 5 in [13]). The only difference between this work and the work of Ho et al. [13] is that we used here different parameter settings for the GA. Accordingly,

FIGURE 11: Membership functions of FO (a) and ZO (b) Sugeno fuzzy models after training with GA.

different prediction error for the same data has been resulted in RMSE = 0.0488. These findings reveal the fact that the parameter settings of GA (number of generations, number of population, etc.) have a considerable influence on the resulting optimal solution and the prediction accuracy.

7. Conclusions

The ability of predicting the surface roughness of end milling process without carrying actual experiment will help to develop automatic manufacturing system. In this work, three algorithms have been examined. The aim is to determine the most effective method for the prediction of surface roughness. From the results obtained from this work, a number of concluding remarks can be summarized as follows.

- (i) RBFN has been found to be the most successful technique to perform surface roughness prediction with RMSE of 2.95%. With comparison to the other predicting algorithms examined in this work, it is the simplest and the fastest method for the problem under consideration. This kind of artificial neural network has proved to be the most effective mean in capturing the nature of the training data, and the best

results have been achieved when examined with the testing data. Unlike other types of RBFN like the regularization and generalized RBFN, the implemented RBFN in this work does not require a trial-and-error procedure.

- (ii) The achieved prediction error by the RBFN outperforms previous results achieved by previous works. In [12], the RMSE using ANFIS with triangular membership functions is 3.47%, and in [13], the FO Sugeno model tuned by G-FIS, the RMSE is 3.32%. Little achievement has been done by the training algorithm presented in [33], where the RMSE is 3.19%.
- (iii) To the best of knowledge of the author of this work, the proposed RBFN has not been examined before in relation to the problem of surface roughness prediction. The presented results here may open the door for other applications.
- (iv) With regard to ANFIS networks, the results of this work and previous results show that the type of membership function plays an important role in the prediction accuracy. Triangular and Gaussian membership functions result in similar prediction performance. Using different experimental data as in [32]

resulted in different results; that is, the triangular membership functions perform better. Lower prediction accuracy has taken place when trapezoidal membership functions are used [12]. This reveals the complicity of fuzzy systems as universal function approximators and the need for more mathematical rigorous to determine the proximity nature of fuzzy systems. Nevertheless, this kind of optimized networks suffers the problem of local minima.

- (v) Results of G-FIS (genetic-based fuzzy inference systems) show that ANFIS networks trained by the HL algorithm have performed better. This enforces our conclusion that using GAs in optimization cannot ensure the obtaining of *perfect* optimal solution, especially in complex systems, unless suitable parameter settings and tuning ranges have been known in advance, which is difficult, if not impossible, to satisfy. Determining the suitable range for all the parameters of a FIS using trial-and-error procedure is a tedious and time-consuming task. This is despite the fact that the obtained solution is the best one of the examined solutions; that is, number of generations multiplied by the number of genes in the population. It means that the obtained solution is the optimal one under some predefined conditions, but not necessarily the most *effective* solution.

References

- [1] K. Bouacha, M. A. Yallese, T. Mabrouki, and J.-F. Rigal, "Statistical analysis of surface roughness and cutting forces using response surface methodology in hard turning of AISI 52100 bearing steel with CBN tool," *International Journal of Refractory Metals and Hard Materials*, vol. 28, no. 3, pp. 349–361, 2010.
- [2] D. Singh and P. V. Rao, "A surface roughness prediction model for hard turning process," *International Journal of Advanced Manufacturing Technology*, vol. 32, no. 11–12, pp. 1115–1124, 2007.
- [3] S. Agarwal and P. Venkateswara Rao, "Modeling and prediction of surface roughness in ceramic grinding," *International Journal of Machine Tools and Manufacture*, vol. 50, no. 12, pp. 1065–1076, 2010.
- [4] A. M. Zain, H. Haron, and S. Sharif, "Prediction of surface roughness in the end milling machining using Artificial Neural Network," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1755–1768, 2010.
- [5] J. F. Briceno, H. El-Mounayri, and S. Mukhopadhyay, "Selecting an artificial neural network for efficient modeling and accurate simulation of the milling process," *International Journal of Machine Tools and Manufacture*, vol. 42, no. 6, pp. 663–674, 2002.
- [6] D. Karayel, "Prediction and control of surface roughness in CNC lathe using artificial neural network," *Journal of Materials Processing Technology*, vol. 209, no. 7, pp. 3125–3137, 2009.
- [7] E. S. Topal, "The role of stepover ratio in prediction of surface roughness in flat end milling," *International Journal of Mechanical Sciences*, vol. 51, no. 11–12, pp. 782–789, 2009.
- [8] C. Lu, "Study on prediction of surface quality in machining process," *Journal of Materials Processing Technology*, vol. 205, no. 1–3, pp. 439–450, 2008.
- [9] O. Çolak, C. Kurbanoglu, and M. C. Kayacan, "Milling surface roughness prediction using evolutionary programming methods," *Materials and Design*, vol. 28, no. 2, pp. 657–666, 2007.
- [10] S. S. Roy, "Design of genetic-fuzzy expert system for predicting surface finish in ultra-precision diamond turning of metal matrix composite," *Journal of Materials Processing Technology*, vol. 173, no. 3, pp. 337–344, 2006.
- [11] H. Oktem, T. Erzurumlu, and F. Erzincanli, "Prediction of minimum surface roughness in end milling mold parts using neural network and genetic algorithm," *Materials and Design*, vol. 27, no. 9, pp. 735–744, 2006.
- [12] S. P. Lo, "An adaptive-network based fuzzy inference system for prediction of workpiece surface roughness in end milling," *Journal of Materials Processing Technology*, vol. 142, no. 3, pp. 665–675, 2003.
- [13] 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*, vol. 36, no. 2, part 2, pp. 3216–3222, 2009.
- [14] P. G. Benardos and G.-C. Vosniakos, "Predicting surface roughness in machining: a review," *International Journal of Machine Tools and Manufacture*, vol. 43, no. 8, pp. 833–844, 2003.
- [15] H. L. Shieh, Y. K. Yang, P. L. Chang, and J. T. Jeng, "Robust neural-fuzzy method for function approximation," *Expert Systems with Applications*, vol. 36, no. 3, part 2, pp. 6903–6913, 2009.
- [16] A. B. Sharkawy, "Genetic fuzzy self-tuning PID controllers for antilock braking systems," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 7, pp. 1041–1052, 2010.
- [17] J.-S. Roger Jang and C.-T. Sun and Eiji Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice Hall, 1997.
- [18] S. A. Billings, H. L. Wei, and M. A. Balikhin, "Generalized multiscale radial basis function networks," *Neural Networks*, vol. 20, no. 10, pp. 1081–1094, 2007.
- [19] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, 2nd edition, 1999.
- [20] M. M. Gupta, L. Jin, and N. Homma, *Static and Dynamic Neural Networks: From Fundamental to Advanced Theory*, John Wiley & Sons, Hoboken, NJ, USA, 2003.
- [21] S. Chen, X. Hong, C. J. Harris, and P. M. Sharkey, "Sparse modeling using orthogonal forward regression with PRESS statistic and regularization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 2, pp. 898–911, 2004.
- [22] I. Asiltürk and M. Çunkaş, "Modeling and prediction of surface roughness in turning operations using artificial neural network and multiple regression method," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5826–5832, 2011.
- [23] M. R. Razfar, R. Farshbaf Zinati, and M. Haghshenas, "Optimum surface roughness prediction in face milling by using neural network and harmony search algorithm," *International Journal of Advanced Manufacturing Technology*, vol. 52, no. 5–8, pp. 487–495, 2011.
- [24] M. F. F. Ab. Rashid and M. R. Abdul Lani, "Surface roughness prediction for CNC milling process using artificial neural network," in *Proceedings of the World Congress on Engineering (WCE '10)*, vol. 3, London, UK, July 2010.
- [25] S. Z. Chavoshi and M. Tajdari, "Surface roughness modelling in hard turning operation of AISI 4140 using CBN cutting tool," *International Journal of Material Forming*, vol. 3, no. 4, pp. 233–239, 2010.

- [26] G. K. MohanaRao, G. Rangajanardhaa, D. HanumanthaRao, and M. SreenivasaRao, "Development of hybrid model and optimization of surface roughness in electric discharge machining using artificial neural networks and genetic algorithm," *Journal of Materials Processing Technology*, vol. 209, no. 3, pp. 1512–1520, 2009.
- [27] P. D. Wasserman, *Advanced Methods in Neural Computing*, Van Nostrand, New York, NY, USA, 1993.
- [28] M. Zanaganeh, S. J. Mousavi, and A. F. Etemad Shahidi, "A hybrid genetic algorithm-adaptive network-based fuzzy inference system in prediction of wave parameters," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 8, pp. 1194–1202, 2009.
- [29] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [30] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, Mass, USA, 1999.
- [31] A. K. Nandi and D. K. Pratihari, "Automatic design of fuzzy logic controller using a genetic algorithm—to predict power requirement and surface finish in grinding," *Journal of Materials Processing Technology*, vol. 148, no. 3, pp. 288–300, 2004.
- [32] N. C. Tsourveloudis, "Predictive modeling of the Ti6Al4V alloy surface roughness," *Journal of Intelligent Robot Systems*, vol. 60, pp. 513–530, 2010.
- [33] M. Dong and N. Wang, "Adaptive network-based fuzzy inference system with leave-one-out cross-validation approach for prediction of surface roughness," *Applied Mathematical Modelling*, vol. 35, no. 3, pp. 1024–1035, 2011.