Review of ANFIS and its Application in Control of Machining Processes

Kabini S.K., Ikua B.W. and Nyakoe G.N.

Abstract—Adaptive Neural Fuzzy Inference Systems (ANFIS) are increasingly becoming popular in the modern world. This is due to their ability to model or represent vagueness in day to day activities or processes. These systems have the potential to adaptively control processes that present a difficulty to the conventional control techniques due to their ability to predict the likely outcome given a set of conditions or inputs. This paper looks at ANFIS and its applications in multidisciplinary fields and specifically, its use in control of machining processes. Some potential areas of application ANFIS are highlighted.

Keywords—ANFIS, Adaptive, Control, Machining.

I. Introduction

THe design of modern control systems is characterized by stringent performance and robustness requirements and therefore relies on model-based design methods. This introduces a strong need for effective modeling techniques. Many systems are not amenable to conventional modeling approaches due to strongly nonlinear behaviour and lack of precise knowledge of the process under study [1]. Nonlinear identification is therefore becoming an important tool which can lead to improved control systems along with considerable time saving and cost reduction. Among the different nonlinear identification techniques, methods based on fuzzy sets are gradually becoming established [2].

Adaptive control is a method of designing a controller with some adjustable parameters and an embedded mechanism for adjusting these parameters. Adaptive methods have been used mainly to improve the controllers performance online [3]. For each control cycle, the adaptive algorithm is normally implemented in three basic steps, namely,

1) Observable data is collected to calculate the controllers performance.
2) The controller's performance is used to calculate the adjustment to a set of controller parameters.
3) The controller's parameters are then adjusted to improve the performance of the controller in the next cycle.

Normally, an adaptive controller is designed based on one of the available techniques. Each

Kabini S.K. Department of Mechatronic Engineering, JKUAT (phone: +254720980179; e-mail: kkabini@eng.jkuat.ac.ke).

Ikua B.W, Department of Mechatronic Engineering, JKUAT (e-mail: ikua_bw@eng.jkuat.ac.ke).

Nyakoe G.N., Department of Mechatronic Engineering, JKUAT (e-mail: nyakoegn@eng.jkuat.ac.ke

technique is originally designed for a specific class of dynamic systems. The controller is then adjusted, as data is collected during run time to extend its effectiveness to control a larger class of dynamic systems.

ANFIS, which is derived from the term Adaptive Network Fuzzy Inference System, was first proposed by Jang in 1993, [4], and later changed to Adaptive Neural Fuzzy Inference System. This system is designed to allow IF-THEN rules and membership functions (fuzzy logic) to be constructed based on the historical data and also includes the adaptive nature for automatic tuning of the membership functions [5].

ANFIS refers to an inference system that integrates the best features of neural network and fuzzy logic. It is a system that predicts input/output relationship of given set of data [6]. It consists of nodes and directional links through which the nodes are connected. Part or all of the nodes are adaptive, which means that their outputs depend on the parameter(s) pertaining to these nodes and the learning rule specifies how these parameters should be changed to minimize the error measure [4].

The basic learning rule of adaptive networks is based on gradient descent and the chain rule which was proposed by Werbos [7] in the 1970's. However, since the basic learning rule is based the gradient method which is slow and has tendency to become trapped in local minima, Jang proposed a hybrid learning rule which sped up the learning process substantially. Both the batch learning and the pattern learning of the hybrid learning rule are discussed below.

II. Structure of ANFIS

ANFIS is composed of five functional blocks as shown in Fig. 1. These are:

- a *rule base* containing a number of fuzzy IF-THEN rules
- a *database* which defines the membership functions of the fuzzy sets used in the fuzzy rules
- a *decision-making unit* which performs the inference operations on the rules
- a *fuzzification interface* which transforms the crisp inputs into degrees of match with linguistic values
- a *defuzzification interface* which transform the fuzzy results of the inference into a crisp output.

Usually, the rule base and the database are jointly referred to as the knowledge base.

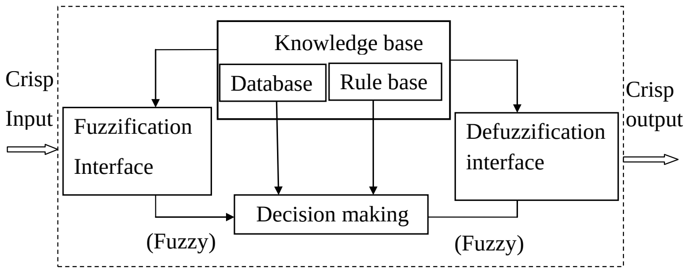

Fig. 1. Block representation of ANFIS

III. Learning algorithms

A. Architecture of ANFIS and its basic learning rule

An adaptive neural network (see Fig. 2) is a multilayer feedforward network in which each node performs a particular function called node function on incoming signals as well as a set of parameters pertaining to this node [8]. The formulas for the node functions may vary from node to node, and the choice of each node function depends on the overall input-output function which the adaptive network is required to carry out [4]. The links in an adaptive network only indicate the flow direction of signals between nodes; no weights are associated with the links. To reflect different adaptive capabilities, circle

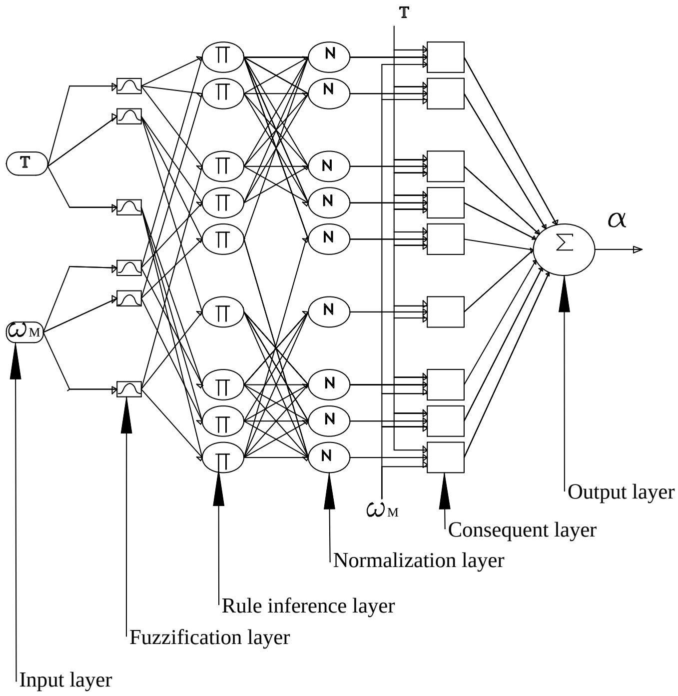

Fig. 2. An Adaptive network

and square nodes are used in an adaptive network. A square node which is an adaptive node has parameters while a circle node which is fixed node has none. The parameter set of an adaptive network is the union of the parameter sets of each adaptive node. In order to achieve a desired input-output mapping, these parameters are updated according to given training data and a gradient-based learning procedure [9].

As an example, suppose that a given adaptive network has L layers and the k^{th} layer has k nodes. The node in the i^{th} position of the k^{th} layer can be denoted by k, i and its node function (or node output) by O_i^k. Since a node output depends on its incoming signals and its parameter set, then;

$$O_i^k = O_i^k(O_i^{k-1}, ...O_{\sharp(k-1)}^{k-1}, a, b, c, ...) \tag{1}$$

where a, b, c, etc., are the parameters pertaining to this node, and $\sharp$ represents a number indicating the position of the node in the layer. O_i^k is used as both the node output and node function.

Assuming that the given training data set has P entries, the error measure for the p^{th} ($1 \leq p \leq P$) entry of training data can be defined as the sum of squared errors:

$$E_p = \sum_{m=1}^{\sharp(L)} (T_{m,p} - O_{m,p}^L)^2 \tag{2}$$

where $T_{m,p}$ is the m^{th} component of p^{th} target output vector, and $O_{m,p}^L$ is the m^{th} component of actual output vector produced by the presentation of the p^{th} input vector. The overall error measure is;

$$E = \sum_{p=1}^{P} E_p \tag{3}$$

In order to develop a learning procedure that implements gradient descent in E over the parameter space, first the error rate $\frac{\partial E}{\partial O}$ for p^{th} training data and for each node output O is calculated. The error rate for the output node at (L, i)can be calculated from [4]:

$$\frac{\partial E_p}{\partial O_{i,p}^L} = -2(T_{i,p} - O_{i,p}^L) \tag{4}$$

For the internal node at (k, i), the error rate can be derived by the chain rule:

$$\frac{\partial E_p}{\partial O_{i,p}^k} = \sum_{m=1}^{\sharp}(k+1)\frac{\partial E_p}{\partial O_{m,p}^{k+1}}\frac{\partial O_{m,p}^{k+1}}{\partial O_{i,p}^k} \tag{5}$$

where $1 \leq k \leq L$ - 1. That is, the error rate of an internal node can be expressed as a linear combination of the error rates of the nodes in the next layer. Therefore for all $1 \leq k \leq L$ and $1 \leq i \leq \sharp(k)$, we can find $\frac{\partial E_p}{\partial O_{i,p}^k}$ by Eqns. 4 and 5. If α is a parameter of the given adaptive network, then Eqn. 5 becomes;

$$\frac{\partial E_p}{\partial \alpha} = \sum_{O^* \in S}\frac{\partial E_p}{\partial O^*}\frac{\partial O^*}{\partial \alpha}, \tag{6}$$

where S is the set of nodes whose outputs depend on α. Then the derivative of the overall error measure E with respect to α is,

$$\frac{\partial E_p}{\partial \alpha} = \sum_{p=1}^{P}\frac{\partial E_p}{\partial \alpha} \tag{7}$$

Accordingly, the update formula for the generic parameter α is,

$$\Delta\alpha = -\eta\frac{\partial E}{\partial \alpha} \tag{8}$$

in which η is a learning rate. The learning rate can be further expressed as [4];

$$\eta = \frac{k}{\sqrt{\sum_{\alpha}(\frac{\delta E}{\partial \alpha})^2}} \tag{9}$$

where k is the step size, the length of each gradient transition in the parameter space. Usually, the value of k is changed to vary the speed of convergence.

There are two learning algorithms for adaptive networks. With the *batch learning* or *off-line learning*, the update formula for parameter α is based on Eqn. 7 and the update action takes place only after the whole training data set has been presented, i.e., only after each *epoch* or *sweep*. On the other hand, if the parameters are to be updated immediately after each input-output pair has been presented, then the update formula is based on Eqn. 6 and is referred to as the *pattern learning* or *on-line learning*.

B. Hybrid learning rule: Batch (off-line) learning

Hybrid learning rule combines the gradient method and the least squares estimate (LSE) to identify parameters [10]. If the adaptive network under consideration has only one output, then

$$output = F(\vec{I}, S) \tag{10}$$

where $\vec{I}$ is the set of input variables and S is the set of parameters. If there exists a function H such that the composite function $H \circ F$ is linear in some of the elements of S, then these elements can be identified by the least squares method. More formally, if the parameter set S can be decomposed into two sets

$$S = S_1 \oplus S_2 \tag{11}$$

where $\oplus$ represents sum such that $H \circ F$ is linear in the elements of S_2, then upon applying H to Eqn. 10, we have;

$$H(output) = H \circ F(\vec{I}, S) \tag{12}$$

which is linear in the elements of S_2. Now given values of elements of S_1, training data which can be denoted by P, can be input into Eqn. 12 and a matrix equation obtained:

$$AX = B \tag{13}$$

where X is an unknown vector whose elements are parameters in S_2. Let $|S_2|$ = M, then the dimensions of A, X and B are $P \times M, M \times 1$ and $P \times 1$, respectively. Since P, the number of training data pairs is usually greater than M, the number of linear parameters, this is an overdetermined problem, and generally there is no exact solution to Eqn. 13. Instead, LSE of X and X^*, is sought to minimize the squared error $\|AX - B\|^2$. The most well-known formula for X^* uses the pseudo-inverse of X [6].

$$X^* = (A^T A)^{-1} A^T B \tag{14}$$

While Eqn. 14 is concise in notation, it is expensive in computation when dealing with the matrix inverse and, moreover, it becomes ill defined if $A^T A$ is singular. As a result, sequential formulas are employed to compute the LSE of X. This sequential method of LSE is more efficient, especially when M is small and can be easily modified to an on-line version i.e., for systems with changing characteristics. Specifically, let the ith row vector of matrix A defined in Eqn. 13 be a_i^T and the ith element of B be b_i^T, then X can be calculated iteratively using the following sequential formulas in [6], [11];

$$X_{i+1} = X_i + S_{i+1} a_{i+1} (b_{i+1}^T - a_{i+1}^T X_i) \tag{15}$$

$$S_{i+1} = S_i - \frac{S_i a_i + 1 a_{i+1}^T S_i}{1 + a_{i+a}^T S_i a_{i+1}} \tag{16}$$

for i=0,1,...,P-1, where S_i is the covariance matrix, and the least squares estimate X^* is equal to X_p. The initial conditions to bootstrap Eqn. 15 and 16 are $X_0 = 0$ and $S_0 = \gamma I$, where γ is a positive large number and I is the identity matrix of dimension $M \times M$. When dealing with multi-output adaptive networks, output in Eqn. 10 is a column vector and Eqns. 15 and 16 still apply except that b_i^T is the i^{th} row of matrix B.

The gradient method and the least squares estimate can be combined to update the parameters in an adaptive network. Each epoch of this hybrid learning procedure is composed of a forward pass and a backward pass. In the forward pass, input data is supplied and functional signals go forward to calculate each node's output until the matrices A and B in Eqn. 13 are obtained. The parameters in S_2 are identified by the sequential least squares formulas in Eqns. 15 and 16. After identifying parameters in S_2, the functional signals keep going forward until the error measure is obtained. In the backward pass, the error rates or the differential error measure with respect to each node output, (see Eqns. 4 and 5) propagate from the output end toward the input end, and the parameters in S_1 are updated by the gradient descent method in Eqn. 8.

For given fixed values of parameters in S_1, the parameters in S_2 thus found are guaranteed to be the global optimum point in the S_2 parameter space due to the choice of the squared error measure. Not only does this hybrid learning rule decrease the dimension of the search space in the gradient method, but, in general, it also cuts down substantially the convergence time.

Taking an example of one-hidden-layer back-propagation neural network with sigmoid activation functions. If this neural network has p output units, then the output in Eqn. 10 is a column vector. Let $H(\circ)$ be the inverse sigmoid function given by,

$$H(x) = \ln(\frac{x}{1-x}) \tag{17}$$

then Eqn. 12 becomes a linear function such that each element of H is a linear combination of the parameters pertaining to layer 2. In other words, S_1 and S_2 are the thresholds of hidden and output layers, respectively. Therefore, the back-propagation learning rule can be applied to tune the parameters in the hidden layer, and the parameters in the output layer can be identified by the least squares method. However, by using the least squares method on the data transformed by $H(\circ)$, the parameters that are obtained are optimal in terms of the transformed squared error measure instead of the original one.

C. Hybrid learning rule: Pattern (on-line) learning

If the parameters are updated after each data presentation, then this is referred to as pattern learning or on-line learning. This learning paradigm is vital to the on-line parameter identification for systems with changing characteristics. To modify the batch learning rule to its on-line version, the gradient descent should be based on E_p (see Eqn. 5) instead of E.

For the sequential least squares formula to account for the time-varying characteristics of the incoming data, there is need to decay the effects of old data pairs as new data pairs become available. This problem is well studied in the adaptive control and system identification literature and a number of solutions are available [11]. The simplest method is to formulate the squared error measure as a weighted version that gives higher weighting factors to more recent data pairs. This amounts to the addition of a *forgetting factor* λ to the original sequential formula:

$$X_{i+1} = X_i + S_{i+1}a_{i+1}(b_{i+1}^T - a_{i+1}^T X_i) \quad (18)$$

$$S_{i+1} = \frac{1}{\lambda}[S_i - \frac{S_i a_i + 1 a_{i+1}^T S_i}{\lambda + a_{i+a}^T S_i a_{i+1}}] \quad (19)$$

for i=0,1,...,P-1, where the value of λ is between 0 and 1. The smaller λ is, the faster the effects of old data decay become.

IV. Application of ANFIS in the control of machining processes

ANFIS has been applied in control of some machining processes. It is mainly employed in processes that present a difficulty to the conventional control processes. This could be due to unpredictable variation of the process parameters or unavailability of complete information on the parameters [12]. Some cases where ANFIS has been employed in the control of machining processes include wire electro discharge machining (WEDM, grinding, tool condition monitoring, and boring operations).

An ANFIS based Fuzzy Logic Controller for the control of chatter vibration in cylindrical grinding process was developed in [13]. ANFIS was used to generate membership functions and rule base for a fuzzy logic controller. The ANFIS was trained by data obtained from a dynamic model for the process. The ANFIS based FLC controller was validated through a series of experiments. The controller was found to effectively reduce vibrations from a range of $10^{-1}\mu$m to a range of $10^{-2}\mu$m. The controller was implemented as shown in Fig. 3

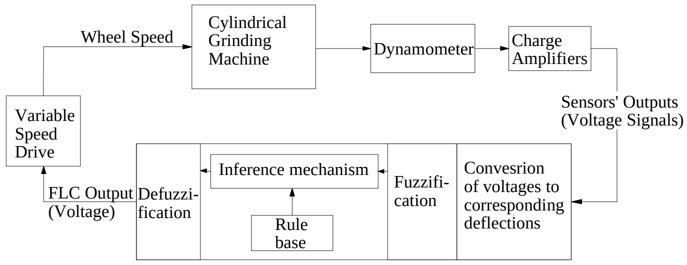

Fig. 3. Schematic diagram for the control of cylindrical grinding process using FLC

An adaptive neuro-fuzzy inference system (ANFIS) model for the prediction of the white layer thickness (WLT) and the average surface roughness in wire electrical discharge machined parts which are functions of the process parameters was developed in [14]. This was due to two facts. One is that wire electrical discharge machined surfaces are characterized by their roughness and metallographic properties and, two is that, surface roughness and white layer thickness (WLT) are the main indicators of quality of a component for WEDM. Pulse duration, open circuit voltage, dielectric flushing pressure and wire feed rate were taken as model's inputs. The model combined modeling function of fuzzy inference with the learning ability of artificial neural network. A set of rules was generated directly from the experimental data. The model's predictions were verified with experimental work.

An ANFIS for tool condition monitoring (TCM) of twist drill wear was developed in [15]. A multilayer feed-forward neural network and a back propagation training algorithm for ANFIS were used. The algorithm utilized vibration signature analysis as the main and only source of information from the machining process.

A neural-fuzzy inference scheme was applied to predict the flank wear from cutting force signals during end-milling process in [16]. In this work, construction of an ANFIS based system that would provide a linguistic model for the estimation of tool wear from the knowledge embedded in the neural network contribution was proposed. Machining experiments conducted using the proposed method indicated that, by using appropriate maximum force signals, the flank wear could be predicted within 4% of the actual wear for various end-milling conditions.

An ANFIS based on-line system for monitoring boring tools was developed in [17]. This work involved investigation of precision and quality control in boring operations. Fourteen features obtained by processing cutting force signals using virtual instrumentation were used. A Sequential Forward Search (SFS) algorithm was employed to select the best combination of features. Backpropagation neural networks (BPNs) and adaptive neuro-fuzzy inference systems (ANFIS) were used for on-line classification and measurement of tool wear.

The input vectors consisted of selected features. For the on-line classification, the outputs was boring tool condition, which was either usable or worn out. For the on-line measurement, the outputs was estimated value of the tool wear. Using BPN, five features were needed for the on-line classification of boring tools. The features were, average longitudinal force, average value of the ratio between the tangential and radial forces, skewness of the longitudinal force, skewness of the tangential force, and kurtosis of the longitudinal force.

Three features were used for the on-line classification of boring tools by the ANFIS. The features were, average longitudinal force and average of the ratio between the tangential and radial forces, and kurtosis (measure of whether the data are peaked or flat relative to a normal distribution) of the longitudinal force. Only one feature, kurtosis of the longitudinal force, was used for the on-line measurement of tool wear using ANFIS. The study showed that, both $5 \times 20 \times 1$ BPN and 3×5 ANFIS could achieve a 100% success rate for the on-line classification of boring tool conditions .

An Adaptive Neuro-Fuzzy Inference System (ANFIS) technique for modeling and simulation of the material removal rate in stationary ultrasonic drilling of sillimanite ceramic was developed in [18]. In the model, depth of penetration, time for penetration and penetration rate were taken as inputs. The model combined modeling function of fuzzy inference with the learning ability of artificial neural network; and a set of rules that had been generated directly from experimental data. The modeling approach developed was verified by comparing the predicted results with experimental results. Results showed that the values of material removal rate predicted by the proposed model were similar to the experimental values with 0.1% level of significance.

A novel transductive neuro-fuzzy inference method to control force in a high-performance drilling process was presented in [19]. The method was verified by analysis and verification of the behavior of a transductive neuro-fuzzy inference system for controlling the process which is usually complex. The verification addressed issues such as dynamic modeling, computational efficiency, and viability of the real-time application of the algorithm. It also included assessment of the topology of the neuro-fuzzy system (e.g., number of clusters, number of rules).

V. Suggested application areas of ANFIS

ANFIS based control, being a relatively new control technique that does not employ a process model and has the potential to represent vague data as well as the capability of self tuning, i.e., learning and adapting, can be widely used in control of machining processes. Following are suggested possible areas of application of ANFIS in control of machining processes.

- For avoidance of thermal damage to the product and tool in machining processes such as grinding, milling, turning, drilling and jig boring
- In the control of cutting forces machining processes such as milling, grinding and turning for example, to avoid tool breakages
- In controlling the size and shape of a components in machining processes such as turning and milling
- In focusing a laser beam in laser machining
- Adjustment of spark gap in Electro discharge machining
- Control of vibrations in machining processes
- Focusing of water jet in water jet machining

- Automatic feeding and positioning of workpieces in automated machining processes

Acknowledgment

This work was sponsored by Jomo Kenyatta University of Agriculture and Technology.

References

[1] Rowe W.B., Allanson D.R., Pettit J.A., Moruzzi J.L., Kelly S., "Intelligent CNC for grinding," *Proceedings of the Institute of Mechanical Engineers*, vol. 205, pp. 233–239, 1991a.

[2] Robert B., "An overview of nonlinear identification and control with fuzzy systems," in *Intelligent control systems using computational intelligence technique*, 2005.

[3] Jos C. P., Neil R. E., Curt L. W., *Neural and Adaptive Systems: Fundamentals through Simulations*. Wiley, 1999.

[4] Jyh-Shing R. J., "ANFIS : Adaptive-network-based fuzzy inference system," *IEEE Transactions on systems, man, and cybernetics*, vol. 23 No.3, pp. 665–685, 1993.

[5] Jang J.S., Sun C.T., Mizutani E., *Neuro-Fuzzy and Soft Computing A computational approach to learning and machine intelligence*. Prentice-Hall, Inc., Upper Saddle River, New Jersey, 1997.

[6] Piero P. B., "Adaptive neural fuzzy inference systems (ANFIS): Analysis and applications." Lecture notes, 2000.

[7] Paul J. W., *The roots of backpropagation: from ordered derivatives to neural networks*. John Wiley and Sons Inc., 1994.

[8] Takagi T., Sugeno M., "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. on System, Man and Cybernetics*, vol. 4 , No. 9, pp. 116–132, 1985.

[9] Hou Z. X., LI H.Q., "Nonlinear system identification based on adaptive neural fuzzy inference system," *IEEE*, vol. 1, pp. 2067–2069, 2002.

[10] Roger J. S. R., "Fuzzy modeling using generalized neural networks and Kalman filter algorithm," in *In Proceedings of the Ninth National Coference on Artificial Intelligence*, 1991.

[11] Graham C. G., *Adaptive filtering prediction and control*. Prentice-Hall (Englewood Cliffs, N.J), 1984.

[12] Mohammad A., "ANFIS based soft-starting and speed control of AC voltage controller fed induction motor," *IEEE*, vol. 3, pp. 106–110, 2006.

[13] Kabini S. K., "Design of an adaptive controller for cylindrical plunge grinding process," Master's thesis, Jomo Kenyatta University of Agriculture and Technology, 2011.

[14] Ulas Caydasa, Ahmet Hascal?ka and Sami Ekicib, "An adaptive neuro-fuzzy inference system (ANFIS) model for wire-EDM ," *Expert Systems with Applications*, vol. 36, Issue 3, Part 2, pp. 6135–6139, 2009.

[15] Issam Abu-Mahfouz, "Drilling wear detection and classification using vibration signals and artificial neural network," *International Journal of Machine Tools and Manufacture*, vol. 43, Issue 7, pp. 707–720, 2003.

[16] Zuperl Uros, Cus Franc and Kiker Edi, "Adaptive network based inference system for estimation of flank wear in end-milling ," *Journal of Materials Processing Technology*, vol. 209, Issue 3, pp. 1504–1511, 2009.

[17] Liu T.I., Kumagai A. , Wang Y.C., Song S.D., Fu Z. and Lee J., "On-line monitoring of boring tools for control of boring operations ," *Robotics and Computer-Integrated Manufacturing*, vol. 26, Issue 3, pp. 230–239, 2010.

[18] Simranpreet Singh Gill, and Jagdev Singh, "An Adaptive Neuro-Fuzzy Inference System modeling for material removal rate in stationary ultrasonic drilling of sillimanite ceramic ," *Expert Systems with Applications*, vol. 37, Issue 8, pp. 237–246, 2010.

[19] Gajate A. Haber R.E., Vega P.I. and Alique J.R. , "A Transductive Neuro-Fuzzy Controller: Application to a Drilling Process," *IEEE Transactions on Neural Networks*, vol. 21, no.7, pp. 1158–1167, 2010.