ANN BASED MECHANISTIC FORCE MODEL FOR FACE MILLING PROCESSES

A Thesis in Mechatronics
Master of Science in Mechatronics Engineering

Presented to the Faculty of the American University of Sharjah
College of Engineering
in partial fulfillment of
the requirements for the degree

Master of Science

by:
AMAL A. KHATTAB
B.S.2006

Sharjah, UAE
June 2011

We approve the thesis of Amal A. Khattab | Date of Signature

_____ _____

Dr. Ibrahim Deiab
Associate Professor
Department of Mechanical Engineering
Thesis Co-Advisor

_____ _____

Dr. Hany El Kadi
Associate Professor,
Department of Mechanical Engineering
Thesis Co-Advisor

_____ _____

Dr. Mamoun Abdel-Hafez
Assistant Professor
Department of Mechanical Engineering
Graduate Committee

_____ _____

Dr. Basil Darras
Assistant Professor
Department of Mechanical Engineering
Graduate Committee

_____ _____

Dr. Rached Dhaouadi
Director
Mechatronics Engineering Graduate Program

_____ _____

Dr. Hany El Kadi
Associate Dean
College of Engineering

_____ _____

Dr. Yousef Al Assaf
Dean
College of Engineering

_____ _____

Dr. Gautam Sen
Vice Provost
Research and Graduate Studies

ANN BASED MECHANISTIC FORCE MODEL FOR FACE MILLING PROCESSES

Amal A. Khattab, Candidate for the Master of Science Degree
American University of Sharjah, 2011

ABSTRACT

Due to increased global competition and increased calls for environmentally benign machining processes, there has been more focus and interest in making processes more lean and agile to enhance efficiency, reduce emissions and increase profitability. One approach to achieving lean machining is to develop a virtual simulation environment that enables fast and reasonably accurate predictions of machining scenarios, process output and provide access to needed information.

Investigation on the utilization of artificial intelligence is carried out. Artificial Neural networks (ANNs) are employed to develop a smart data base that can provide fast prediction of cutting forces resulting from various combinations of cutting parameters. The data base can also predict the cutting coefficient usually predicted to calibrate the force models. The data base would be highly beneficial to the growing manufacturing industry in the United Arab Emirates (UAE), as it can be used to decide upon optimum parameters prior to carrying out cutting tests. With time, the force model can expand to include different materials, tools, fixtures and machines and would be consulted prior to starting any job.

Predictions are compared to measured experimental results and are shown to be in good agreement. To address some of the difficulties encountered when using ANNs to predict cutting forces, the use of Polynomial classifiers (PCs) was also investigated to predict the cutting forces. A comparison between the predictions obtained using the PCs were found to be in good agreement compared to experimental results.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am heartily thankful to my supervisors, Dr. Ibrahim Deiab & Dr. Hany El Kadi, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject. Many thanks go to Mrs. Salwa for her cooperation and support throughout.

I owe my deepest gratitude to my parents and husband Mohamed El Assadi for their patience and support.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of this thesis.

CHAPTER 1

INTRODUCTION

1.1 Motivation & Background

Manufacturing is an added value process that had always been of significant importance to human civilization. Machining operations comprise a substantial portion of the world's manufacturing infrastructure, making the enhancement and control of metal removal processes one of the main concerns of the industry [1]. Means of achieving the said enhancement is to estimate the product and process quality, tool life and stability of the machining process because they facilitate effective planning of machining operations, optimum performance, quality and cost and hence can lead to the proper selection of machining conditions, optimal fixture design and avoidance of tool failure.

An accurate indicator of the mentioned factors is the cutting force resulting from the cutting process. Numerous attempts and methods have been proposed to predict the forces. In this study, ANNs, mechanistic models, polynomial classifiers and a hybrid/fusion of the techniques shall be investigated.

1.2 Scope of the Research

The current study is aimed at investigating the optimum technique of predicting the cutting forces during the face milling operation. Three techniques have been used, ANNs, ANNs fused with mechanistic modeling and polynomial classifiers.

The intention of fusing the technique of mechanistic with ANNs is to allow the prediction of cutting coefficients hence the cutting forces at conditions other than the ones encountered in training. This will greatly cut down costs and calibration time needed to find the cutting coefficients for each different set of cutting parameters. The ANN-based mechanistic force model will however be more global than mechanistic models, where it will be able to predict forces for cutting parameters it has not seen. The aim is to mix the models and capture the advantages of both models and incorporate it into one ANN based mechanistic force model.

The use of different polynomial classifiers architecture shall also be investigated to find the optimum model for force prediction.

The three techniques are investigated in this study for the purpose of obtaining more accurate prediction of the cutting forces. Figure 1 shows the various approaches addressed in this work

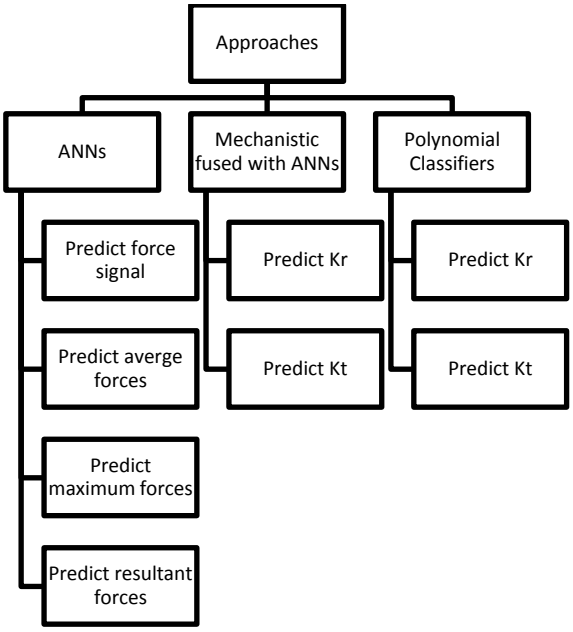

Figure 1 Approaches used in this work to predict cutting forces in end milling

1.3 Thesis Organization

Chapter 2 covers a summary of literature survey of force prediction techniques employed earlier. Chapter 3 describes the experiment design, its setup and the experiments carried out to populate the data used to build the model. Chapter 4 discusses force prediction results using ANNs as well as the results using a hybrid model. Chapter 5 presents results using Polynomial Classifiers. Whereas, Chapter 6 compares results presented in Chapters 4 and 5, Chapter 7 concludes the study and proposes future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Force Prediction

Knowledge of cutting forces beforehand is valuable as it leads to an efficient and automated process through the proper selection of machining parameters, fixture design and appropriate machines and tools used. The challenge in accurately modeling the cutting forces lies in the fact that the milling process is very complex due to the many variables influencing these forces. These variables are highly interlinked and a change of a single parameter will result in different cutting forces [2]. In addition, the machining process is nonlinear and time-dependent.

Different techniques have been used to predict cutting forces. These vary in their generality, accuracy and amount of data required as an input into the model. Analytical methods are hindered by their low accuracy in predicting forces and by their lack of generality as well as the large amount of experimental data needed for each work piece and tool material under various cutting conditions [1]. This renders their use expensive and time consuming. Whereas, mechanistic methods have a higher accuracy in predicting cutting forces; their main drawback is their lack of generality [1].Therefore; traditional identification methods fail to provide a general, accurate force prediction using minimal data. For the above mentioned reasons, ANNs can be an alternative option to use as a substitute or as a supporting tool for the prediction of the cutting forces. Their use will be investigated in this study to develop a model that gives a general, more accurate prediction that requires less experimental data.

Ehmann [1] traced the historical evolution of research in process modeling and have found that in general, all the analytical models did not accurately predict the dynamic forces. They then investigated the experimental methods and found that deriving the dynamic cutting force coefficients, although not trivial, delivers values close to experimental results. Mechanistic and numerical methods are of more recent origin and rely on empirical models and computer simulation

techniques; the latter includes both mechanistic and finite element methods. The authors concluded that a combination of the above listed methods is typically needed to obtain a working model and stated that the mechanistic models showed the most predictive power compared to other methods. For this reason, most of the current research is steered towards the mechanistic force models.

Koeringsberger et al. [3] presented a method in which the cutting forces are assumed to be proportional to the chip cross sectional area with the constants of proportionality called the specific cutting pressures. These pressures depend on the cutter geometry, cutting conditions, insert grade and work piece material properties. Calibration is performed by running tests at different combinations of spindle speed, feed rate and depth of cut. In face milling, these tests are conducted using a single cutting insert to avoid the effect of run out. In addition, these tests are conducted on a work piece with no surface discontinuities to simplify the correlation between the cutting forces and the angular position of the cutting insert.

Coelho [4] used an end mill fitted with a single point indexable insert to perform an orthogonal milling operation. An average curve of force was obtained using 5 consecutive rotations. A simple force model as a function of the undeformed chip thickness was fitted into the force data. Although the results agreed with experimental data, huge differences were observed between the specific cutting coefficients and friction coefficients, depending on cutting speed, feed rate and cutting width.

Jayaram [5] has presented a method for estimating the specific cutting pressures for mechanistic cutting force models in face milling. His research is considered untraditional because it uses multiple cutting inserts. Moreover, the specific cutting pressures are estimated from data collected during actual production and hence there is no production interruption. Results obtained using both the empirical method and the author's method were validated through simulation and experimental tests carried out on a rigid 2024 aluminum work piece using an uncoated carbide insert. Results have shown that both the empirical method and the authors' proposed method are of similar magnitude.

Saffar [6] has proposed a finite element method based model to simulate the cutting forces as well as tool deflection in the end milling operation. An advantage of the proposed model is that the material properties are defined based on the Johnson Cook theory whereas in theoretical relationships the material properties are constants. Another accuracy enhancer is that the simulation allows for the use of nonlinear geometric boundaries. The author concluded that using a finite element based simulations gives more accurate results compared to theoretical relationships.

Wan [7] proposed a new and simplified mechanistic method for the calibration of cutting force coefficients and cutter runout for cylindrical end milling using the measured instantaneous cutting forces instead of the average ones. The author proposed a simplified method for calibrating the cutting force coefficients, which will be valid over a wide range of cutting conditions. The cutting force coefficients were expressed as power functions of the instantaneous uncut chip thickness. The method is achieved by following two steps: first, mathematical relationships between the instantaneous cutting forces and the uncut chip thickness are established. Then, nonlinear algorithms are used to solve the established nonlinear contradiction equations. The method has been validated and was found to be in good agreement with the measured results in both shape and magnitude of the force signal.

2.2 Definition of Cutting Forces in Milling

The cutting process in milling yields three forces of different magnitude and direction that depend on the cutting parameters and the cutter geometry. The forces that can be measured by a dynamometer, are the global forces in the X, Y and Z directions. These forces are then transformed into the axial, radial and tangential force components, which are geometrically related. The axial, radial and tangential forces continuously change in order and magnitude. Figure 2 shows the global X, Y and Z axis and the transformed force system:

F_A : Axial component of the cutting force

F_T : Tangential component of the force

F_R : Radial component of the force

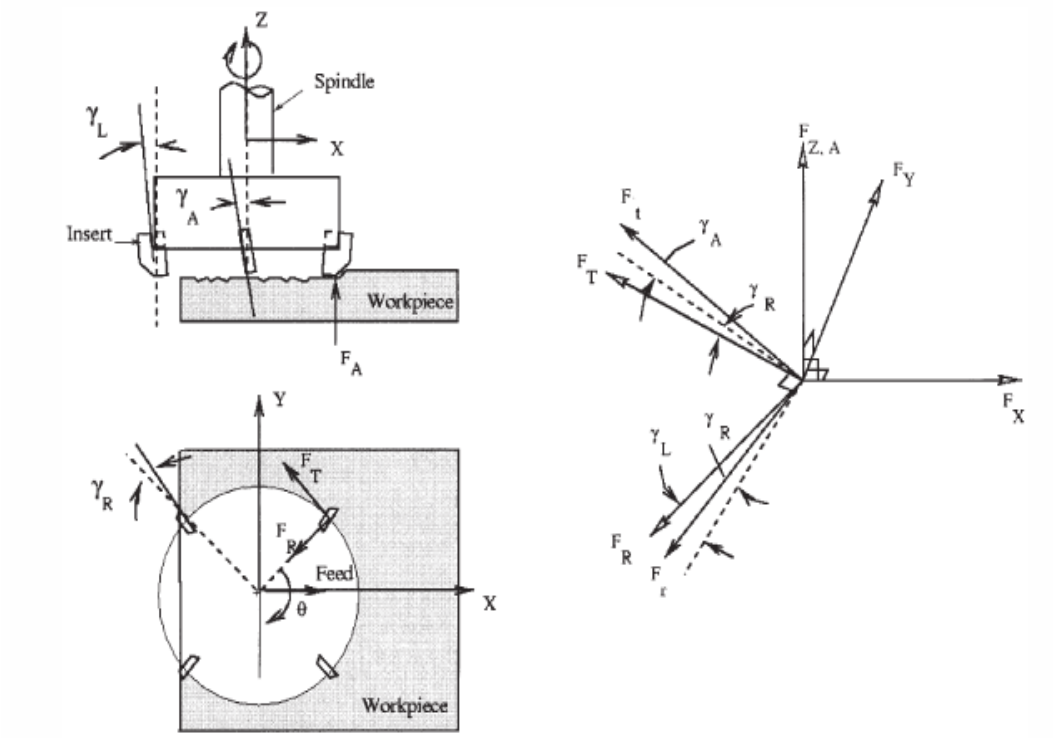

Figure 2 Directions of Cutting Forces in the Milling Process [7]

The transformation of the global forces has been done in a variety of ways: Mechanistic equations have been used to model the cutting–thrust forces, F_C–F_T, the tangential–radial forces, F_T–F_R, and the normal–friction forces, F_N–F_F. The third force component can be obtained from the other two forces and from the geometry of the cutting insert. For this research, the tangential–radial cutting force model will be employed. This model is a slightly modified combination of the models developed by Deiab [8], Fu [9] and Kim [10].

In the tangential–radial mechanistic cutting force model, the specific cutting pressures, K_t and K_r, are represented by the following equations [8]:

$$\ln(K_t(t))=a_0+a_1\ln(t_c(t)), \quad (1)$$

$$\ln(K_r(t))=b_0+b_1\ln(t_c(t)), \quad (2)$$

where:

a_0, a_1, b_0, and b_1: constants for a given combination of tool and work piece material

t_c : the instantaneous chip thickness, see Fu [9].

The constants a_0, a_1, b_0 and b_1 are called the specific cutting pressure constants. The cutting forces in the tangential and radial directions are obtained by multiplying the specific cutting pressures by the chip area, $A_C(t)$:

$$F_t(t)=K_t(t)A_c(t), \quad (3)$$

$$F_r(t)=K_r(t)A_c(t). \quad (4)$$

The tangential, F_t, and radial, F_r, cutting forces described in the above equation depend on the rake face geometry of the cutting insert. The axial rake angle, γ_A, radial rake angle, γ_R and lead angle, γ_L are then used to transform these forces to the global tangential, F_t, radial, F_r, and axial, F_A, directions [10].This transformation, (see Fig. 2), can be described by the following equation:

$$\begin{bmatrix} F_T(t) \\ F_R(t) \\ F_A(t) \end{bmatrix} = T(.) \begin{bmatrix} F_t(t) \\ F_r(t) \end{bmatrix} \quad (5)$$

where:

$$T(\,) = \begin{pmatrix} 1 & \tan(\gamma_R)\cos(\gamma_L)/\cos(\gamma_A) \\ -\tan(\gamma_R) & \cos(\gamma L)/\cos(\gamma A) \\ \tan(\gamma_A)/\cos(\gamma_R) & \sin(\gamma_L)/\cos(\gamma_A))\cos(\gamma_R) \end{pmatrix} \quad (6)$$

Figure 2 shows the directions of the axial, radial and tangential cutting forces and global *X*, *Y*,and *Z* directions for the face milling process. The positive direction of the forces shown in the figure corresponds to the cutting forces acting on the cutter. The angle of the cutting insert, θ, is used to transform the cutting forces from the tangential, radial and axial forces to the global cutting forces in the *X*, *Y* and *Z* directions using the rotational transformation matrix, *R*(*t*). In addition, if an insert is located on a workpiece surface discontinuity, the cutting forces are zero. This is handled using a function, *W*(*t*), which is equal to 1.0 when the insert is cutting the work piece and 0.0 when the insert is outside the work piece. The cutting forces in the *X*, *Y* and *Z* directions can be represented by the following equation:

$$\begin{bmatrix} F_X(t) \\ F_Y(t) \\ F_Z(t) \end{bmatrix} = R(t)W(t) \begin{bmatrix} F_T(t) \\ F_R(t) \\ F_A(t) \end{bmatrix} \quad (7)$$

$$R(t) = \begin{pmatrix} \cos(\Theta) & -\sin(\Theta) & 0 \\ \sin(\Theta) & \cos(\Theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (8)$$

Hence, the cutting force equation can be simplified to the following form:

$$\begin{bmatrix} F_X(t) \\ F_Y(t) \\ F_Z(t) \end{bmatrix} = R(t)W(t)T(\,) \begin{bmatrix} F_t(t) \\ F_r(t) \end{bmatrix} \quad (9)$$

2.3 Calibration Techniques

In order to predict the cutting forces, using Eqs (3) and (4), the specific cutting pressures, K_t and K_r are required. The estimation of the specific cutting pressures requires the determination of the constants (a_0, a_1, b_0 and b_1). These constants are typically estimated from experiments for a given combination of work piece and insert material. The single insert off-line method is commonly used to calibrate the mechanistic cutting force models and determine the specific cutting pressure. In this method, the specific cutting pressure constants are obtained from linear regression of the specific cutting pressures with respect to the chip thickness in the logarithmic space [11]. Figure 3 shows a block diagram of the off-line calibration procedure. The global cutting forces, F_X, F_Y and F_Z, are first transformed to the tangential, F_T, radial, F_R, and axial, F_A, cutting forces using the angular location of the cutting insert, θ [Eqs. (7) and (8)]. Eqs. (3)–(6) are then used to compute the specific cutting pressures, K_t and K_r using the cutter geometry angles (radial rake angle, γ_R, axial rake angle, γ_A, and lead angle γ_L) and the chip cross-sectional area A_c. The specific cutting pressure constants are obtained from the least squares fit of Eqs. (1) and (2) as follows:

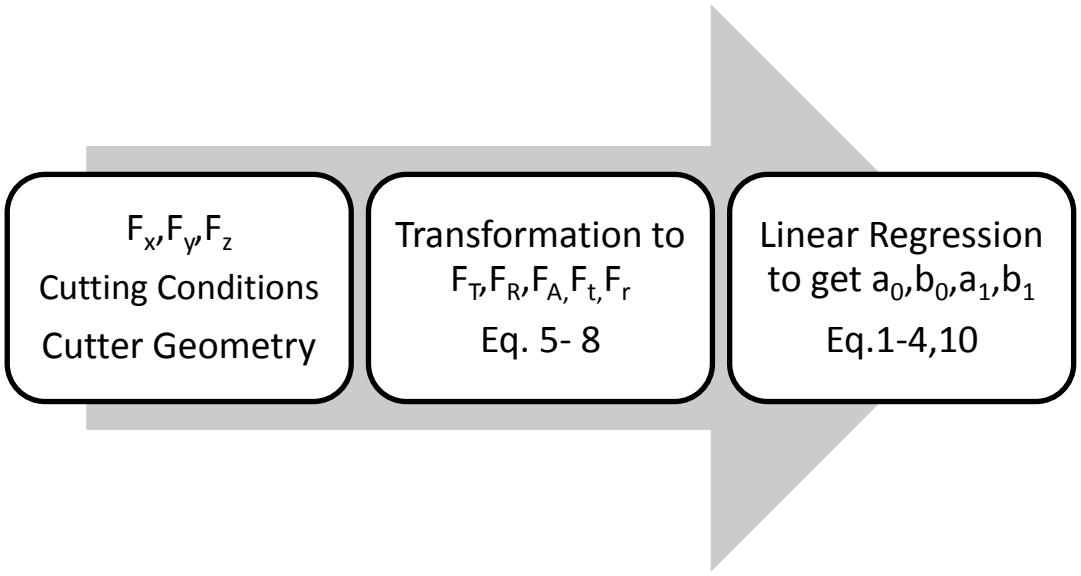

Figure 3 Traditional Methods of Mechanistic Force Modeling

Mathematically, this can be presented as:

$$\begin{pmatrix} a_0 & b_0 \\ a_1 & b_1 \end{pmatrix} = \begin{pmatrix} N & \sum \ln(t_c^i) \\ \sum \ln(t_c^i) & \sum (\ln(t_c^i))^2 \end{pmatrix}^{-1}$$

$$\begin{pmatrix} \sum \ln(K_t^i) & \sum \ln(K_r^i) \\ \sum \ln(K_t^i)\ln(t_c^i) & \sum \ln(K_r^i)\ln(t_c^i) \end{pmatrix} \quad (10)$$

where:

N: total number of calibration tests

t_c^i , K_t^i and K_r^i : are the chip thickness and specific cutting pressures for the ith calibration test.

Typically, calibration tests are conducted at different combinations of feed rate and depth of cut. However, in the face milling process, the chip thickness varies continuously with the rotation of the cutter. Hence a single cutting force profile can provide cutting force information about different values of chip thickness. If tests at different values of feed rate and depth of cut are used to calibrate the cutting force model, then a least squares fit that is valid for a wide range of conditions can be obtained. In addition, the single insert off-line method can also be modified to include the effects of cutting velocity, V and the normal rake angle, γ, [11]. Hence, the single insert off-line method provides the specific cutting

pressures which are valid for a wide range of cutting conditions, cutting velocities and cutter geometry.

2.4 Mechanistic Method - Literature Review

Kang [12] has developed a mechanistic model of cutting forces in the micro end milling process. The analytical model proposed took into account the tool edge radius effect. This effect is a characteristic of the micro end milling process. Experiments have been carried out on aluminum with a 0.2 mm end mill cutter. The author demonstrated that the predicted cutting forces were consistent with the experimental cutting forces. The author however, recommended that studies must be done to develop a better model that can predict cutting forces under a wider variety of conditions while yielding more accurate results.

Yun [13] proposed a different method of accurately predicting cutting forces using cutting condition independent coefficients. He considered that the coefficients are affected only by the uncut instantaneous chip thickness, which is estimated by following the movement of the position of the center of the cutter. The test results indicate that the proposed method is effective and accurate. This was confirmed by comparing the predicted results to the experimental results. Moreover, the cutter run out offset was confirmed by measuring it with a dial gauge. The author stated that the measured value was in the range of 5-6μm, where the estimated value using the developed model was 6 μm.

Yussefian and Imani [14] showed that cutting force is related to chip area and cutting edge contact length by mechanistic cutting force coefficients. B-spline parametric curves are used to model the process geometry. The chip load boundary is first modeled using analytical parametric curve intersection approach. The chip area is then segmented into elements for which feed and approach angle can be considered constant. Cutting force coefficients extracted from an orthogonal database are used to correlate the geometric modeling to cutting force components. In this research, experimental machining tests have been conducted for a wide range of cutting conditions. The results confirm the validity of the proposed method within the deviation of $\pm$10%. Due to the fact that the proposed

approach incorporates orthogonal cutting database, cutting force prediction can be implemented for different cutting edge geometries.

Wan and Zhang [15] systematically studied the cutting force modeling methods in peripheral milling process in the presence of cutter runout. Emphasis was put on how to efficiently calibrate the cutting force coefficients and cutter runout. Mathematical derivations and implementation procedures were carried out based on the measured cutting force or its harmonics from Fourier transformation. Five methods were presented: in the first three methods the cutting force coefficients are assumed to be constants whereas in the last two they are taken as functions of instantaneous uncut chip thickness. The first and fifth methods were taken from the literature for comparison. The second method proceeds using the first and N_k^{th} harmonic forces as the source signal while the third and the fourth are derived based on the measured cutting forces and their first harmonics. Methods 4 and 5, where a mathematical model employing varying cutting coefficients (as a function of the uncut chip thickness) was used to relate the cutting forces and the instantaneous uncut chip thickness, gave best results.

2.5 Use of ANN in force modeling- ANNs Background & Literature

The theory of artificial neural networks was first proposed in 1940's to simulate the work of the human brain [16]. ANN can generally be defined as a structure composed of a number of interconnected units [17]. Each unit has an input/output (I/O) characteristic and implements a local computation or function. The output of each unit is determined by its I/O characteristic, its interconnection to other units and (possibly) external inputs, and its internal function. The network usually develops an overall functionality through one or more forms of training. The fundamental unit or building block of the ANN is called artificial neuron (called neuron from here on) [18]. The neuron has a set of inputs (X_i) weighted before reaching the main body of the processing element. In addition, it has a bias term, a threshold value that has to be reached or exceeded for the neuron to produce a signal, a non-linearity function (f_i) that acts on the produced signal (R_i), and an output (O_i). The basic model of the artificial neurons is illustrated in Figure 4.

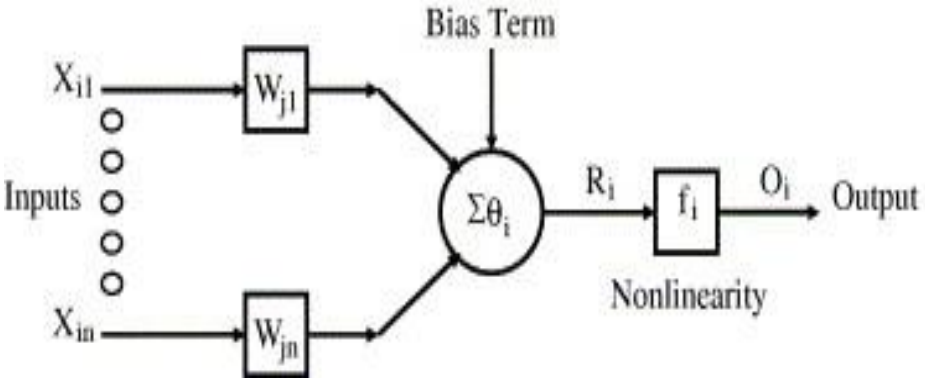

Figure 4 Basic model of artificial neuron [19]

Artificial neural networks consist of an input layer, one or more hidden layers and an output layer as shown in Figure 5.

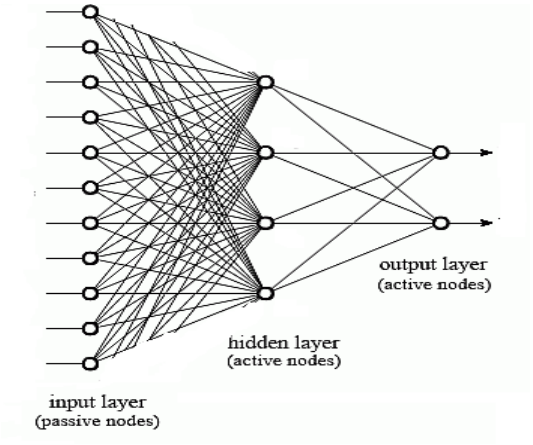

Figure 5 ANN Configuration

The greatest advantage of artificial neural networks is their ability to model complex non-linear, multi-dimensional functional relationships without any prior assumptions about the nature of the relationships and the network is built directly from experimental data by its self-organizing capabilities [20].

Several neural network architectures can be used to address the problem at hand. In this work we will be using the following ANN structures:

1- Feedforward Neural Networks (FNN)
2- Cascade Feedforward Neural Networks (CFFN)
3- Elman Networks (ELM)
4- Layer Recurrent Network (LRN)
5- Radial Basis Networks (RDN)

More details about the ANNs are shown in Appendix F.

Tandon and Al Monayri [21] have presented an ANN-based force model for the end milling process. The process depends on a number of parameters that are strongly interlinked. The authors designed a feed forward back propagation neural network with one or two hidden layers and sigmoid activation functions. The maximum force and the mean force are the output parameters. Their predictions compared well with the experimental results. The error obtained was attributed to: inaccuracies in the computation of the immersion geometry as it varies during cornering and to the wear condition of the tool which results in an increase in the cutting force.

Aykut et al. [22] developed a model for predicting the cutting forces as function of cutting parameters for the face milling of Stellite6 (a cobalt based alloy) using ANNs. Asymmetric milling was chosen over symmetric milling because the cutting tool diameter is larger than the milling width which extends tool life. A feed-forward ANN with back-propagation was initially adopted in order to reach the optimal network architecture, the authors experimented with various network architectures, different training algorithms and altering the number of neurons per hidden layer and transfer functions in the hidden layers/output layer. The selected ANN model consisted of three input parameters corresponding to: cutting speed, feed rate and depth of cut, 35 hidden neurons and three outputs corresponding to the cutting forces in the x-, y- and z-directions. The authors concluded that the ANNs can be accurately used to predict the effects of machinability on chip removal cutting parameters for face milling of stellite 6 in asymmetric milling processes.

Cus et al. [23] proposed a neural network based approach to select optimized cutting parameters in turning. The need arises because optimization of machining parameters is a non linear optimization with constraints making it difficult to use the conventional optimization algorithms due to problems with convergence speed and/or accuracy. The authors compared predictions obtained using feed forward back propagation and radial basis networks. They found that the radial basis network (RBN) needed more neurons than the standard feed forward ANNs with the back propagation learning rule, but conceiving of RBNs lasts only part of the

time (8 sec) necessary for training of the feed forward neural networks. The feed forward ANNs give more accurate results, but they require more time (~25min) for training and testing.

Zuperl et al. [24] modeled the end milling machining process of hardened die steel with a ball end mill cutter using neural networks to predict the effect of machining variables (spindle speed, feed rate, axial/radial depth of cut, number of flutes, tool geometry and flank wear). Back propagation and radial basis networks where used. It was shown that a radial basis network is best used as the cutting force modeler.

Zain [25] has used ANNs to predict the surface roughness in the end milling operation. Twenty four samples of data were collected and divided to 18 for training and 6 for testing. The feed forward back propagation is used as the algorithm, with traingdx, learnndx, MSE, logsig as the training, learning, performance and transfer functions, respectively. Three nodes were used for the input layer corresponding to the cutting speed, feed rate and rake angle and a single node was used for the output corresponding to the surface roughness value. The authors carried the investigation by varying the number of hidden layers and the number of neurons in the hidden layers. They found that a 3-1-1 network structure gave the best results in predicting the surface roughness.

Uros [26] used adaptive network based interference systems (ANFIS) for the estimation of flank wear in end milling processes. The author made use of cutting force signals as an input to their ANFIS system, in addition to spindle speed and depth of cut. It was found the ANFIS system could predict flank wear within 0.5 seconds during milling. Moreover, the system was able to predict flank wear for different cutting conditions with an accuracy of 93.64%. The authors concluded that the ANFIS system is a system that can be efficiently used to optimize the end milling process.

Ozel et al. [27] used both regression models and neural networks to predict the surface roughness and tool wear in hard turning. They found that neural networks modeled surface roughness and flank wear better than regression models. Two

different approaches and networks were used in the prediction: the first, using direct process parameters: tool edge geometry, hardness of work piece, cutting speed, feed rate, and cutting length as inputs. The second approach is designed for chamfered and honed tool edge geometries; this approach uses indirect inputs: the mean values of cutting forces. The motive behind using the second approach was to decrease the size of the neural network and hence enable for faster convergence and better predictions. The results showed good agreement with experimental data.

Al-Ahmari [28] studied different predictive machinability models: multiple linear regression analysis techniques (RA), response surface methodology (RSM) and computational neural networks (CNN) to predict tool life, surface roughness and cutting forces in turning. He found out that the CNN models provide better prediction capabilities due to their ability to offer better predictions of complex nonlinearities and interactions than RA and RSM. Furthermore, the RSM models yielded better results than the RA models.

Davim et al. [29] carried out an investigation into the effect of cutting conditions on surface roughness using artificial neural networks. The ANN used is an error back propagation training algorithm with three neurons in the input layer (corresponding to the feed rate, cutting speed and depth of cut), 16 neurons in the hidden layer and two neurons in the output layer (corresponding to the average roughness R_a and the maximum peak to valley height R_t). They concluded that the surface roughness is highly sensitive to feed rate and cutting speed. On the other hand, depth of cut has less effect on surface roughness. Surface roughness had a tendency to reduce with the increase in cutting speed and the decrease in feed rate.

Briceno et al [30] have employed two supervised neural networks to estimate the forces developed during milling. The first one is a back propagation (BP) neural network with log-sigmoid transfer functions in the hidden layers and linear transfer function in the output layer; the second is a radial basis function network (RBF) with Gaussian activation functions. Three inputs were used in this study: feed rate, spindle speed and radial depth of cut. The output parameters were the maximum, minimum, mean and standard deviation of the force. A single hidden

layer was always used, and the number of neurons in the hidden layer was varied to find the "best" network, which was found to be 5 neurons. A cost function is used as a basis of comparison between networks.

Liu and Wang [31] also proposed a back propagation (BP) ANN for on-line modeling of the milling process. However, this study has several limitations, the most important of which is the use of a single machining parameter as the variable input.

2.6 Polynomial Classifiers

The polynomial classifiers are learning algorithms proposed and adopted in recent years for classification, regression, and recognition with remarkable properties and generalization ability [17, 32]. Due to their need for less training examples and far less computational requirements, PCs are used in this work for composite life predictions. In the training phase, the elements of each training feature vector, $x = [x_1, x_2 ..., x_N]$, are combined with multipliers to form a set of basis functions, p(x). The elements of p(x) are the monomials of the form:

$$\prod_{j=1}^{N} x_j^{k_j} \tag{11}$$

where k_j is a positive integer and

$$0 \leq \sum_{j=1}^{N} k_j \leq K \tag{12}$$

For example if the vector x consists of two coefficients, $\mathbf{x}=[x_1\ x_2]$ and a second degree polynomial (i.e. K=2) is chosen, then:

$$P(X) = [1 \quad x_1 \quad x_2 \quad x_1^2 \quad x_1x_2 \quad x_2^2]^T \tag{13}$$

Once the training feature vectors are expanded into their polynomial basis terms, the polynomial network is trained to approximate an ideal output using mean-squared error as the objective criterion. The polynomial expansion for all of the training set features vectors (L vectors) is defined as:

$$M = [P(x_1) \quad P(x_2) \ldots\ldots\ldots\ldots\ldots\ldots P(x_L)]^T \tag{14}$$

The training problem reduces to finding an optimum set of weights, w, that minimizes the distance between the ideal outputs and a linear combination of the polynomial expansion of the training data such that [32]:

$$w^{opt} = arg_w min\|Mw - O\|_2 \quad (15)$$

where O represents the ideal output comprised of the column vector whose entries are the cutting forces under consideration. The weights of the identification models, $\mathbf{w}_{opt}$, can be obtained explicitly by applying the normal equations method [32] such as

$$w^{opt} = M^{+}O \quad (16)$$

where M+ is the Moore-Penrose pseudo-inverse of matrix M [32]

In the prediction stage when an unknown feature vector, x, is presented to the network, the vector is expanded into its polynomial terms ***p(x)*** and its associated logarithmic prediction is determined such that

$$\log(N_f) = w^{opt}P(X) \quad (17)$$

Deiab et al. [33] employed cutting tool wear using statistical signal analysis, pattern recognition, and sensor fusion to predict cutting tool wear. To investigate pattern recognition, polynomial classifiers and neural networks were used. ANNs and polynomial classifiers have been used to predict and classify different tool wear states based on statistical features extracted from sensory information. For polynomial classifiers, a multi-input single-output (MISO) system was modeled via polynomial classifiers. This involves finding the system parameters that best map the multidimensional input sequence (training feature vectors) to the corresponding one-dimensional output sequence (target). As for ANNs, a 3 layer feed forward back propagation network was used, They have demonstrated the effectiveness of both models and found that the predicted experimental test results and the measured tool wear are in good agreement and that the prediction accuracies of the two approaches are comparable. However, polynomial classifiers have shown lower training time over neural networks, where experiments showed that polynomial classifiers require a maximum of 2 s for training compared to few minutes required by neural networks.

2.7 Summary

Analytical methods are hindered by their low accuracy in predicting forces. Their lack of generality and the large amount of experimental data needed for each work material and tool material under various cutting conditions renders their use expensive and time consuming. Mechanistic methods have a high accuracy of predicting cutting forces; however, their main drawback is their lack of generality. Modeling cutting forces using artificial neural networks is another approach used in the literature. ANNs are being used now because milling processes are very complex and many variables influence the forces, these variables are highly interlinked and a change of one parameter would alter the resulting forces. In addition, the machining process is nonlinear and time-dependent, causing traditional identification methods to fail in providing accurate prediction, for the above reasons ANNs are a competitive alternative and maybe a substitute. Moreover, ANNs are robust and global when compared to traditional methods. Available models are either mechanistic based or ANN based. Each model has its advantages and disadvantages. Including AI in the proposed mechanistic force model for face milling process will allow prediction of pressure coefficients hence cutting forces at conditions other than the ones encountered in training the system will be predicted, which will greatly cut down costs and time of running calibration tests to find the pressure coefficients for each different set of cutting parameters. The aim is to mix both models and capture the advantages of both models and incorporate it into one ANN based mechanistic force model. The ANN based mechanistic force model will be more global than mechanistic models, where it will be able to predict forces for cutting parameters it has not seen.

First, the cutting pressures for each set of cutting conditions need to be obtained using neural network schemes rather than the power law (as is the case in mechanistic models). The second step is to obtain a global set of cutting pressures that applies to a range of cutting parameters than to specific values. The proposed model will be tested for face milling process with aluminum alloy 6061 work piece and shall be validated against data that the network has not seen.

CHAPTER 3

EXPERIMENTAL INVESTIGATION

The experimental data used in this work is extracted from the work by Deiab [8]

3.1 Gathering Experimental Data

In order to create a force model, a large amount of force signal data has to be collected. This is to be done using a tool dynamometer; (a force measuring device). Cutting is done on an Aluminum 6061 work piece using an OKUMCadetV4020 face milling machine. Moreover, a cutter diameter of 3 inches (which is less than the workpiece width of 4 inches) is chosen.

The tool holder can carry 4 inserts, but 1 insert will be used as to study the force signal from a single insert. The force resulting from one insert can be superimposed to mimic the 4 inserts placed in the tool.

The Details of the experimental setup are shown in Table 1 and Figure 6.

Table 1 Experiment Setup

Milling Machine	**OKUMCadetV4020**
Tool Dynamometer	Kistler type 9225A
Workpiece	Aluminum 6061
Milling type	Face milling
Work piece width	4 inches
Cutter diameter	3 inches
Insert specifications	Uncoated Carbide H10
Number of inserts	4
Sampling frequency	2005Hz

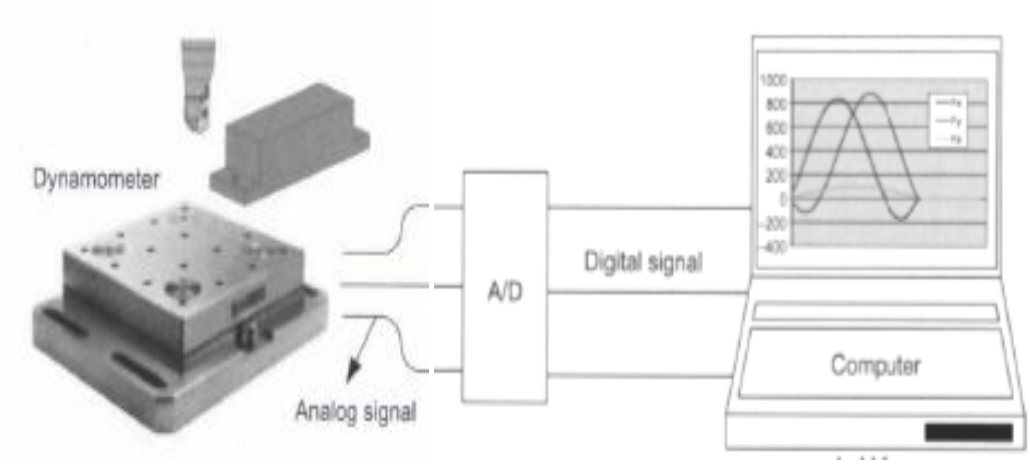

Figure 6 Experimental Setup

3.2 Test Matrix

Various Artificial Intelligence (AI) techniques will be implemented, tested and evaluated for the purpose of achieving accurate prediction of cutting forces. The different combinations of cutting conditions i.e. the test matrix that will be studied is shown in Figure 7.

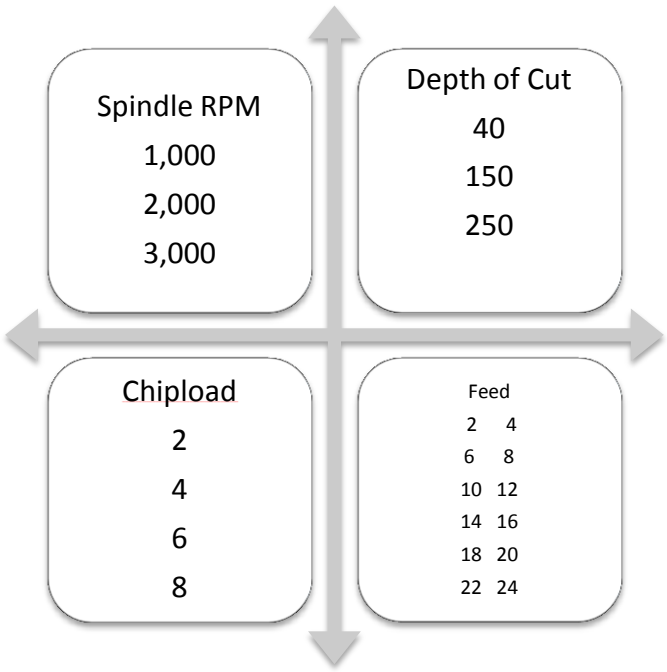

Figure 7 Test Matrix

Combinations of these cutting parameters create a machining data base which shall be used in modeling and simulating this process (Appendix A). The collected data are to be used later in training and validating the developed model. The data is grouped into 36 cutting conditions, denoted by C1, for the 1st cutting condition etc.

3.3 Experimental Procedure

The cutting test matrix was populated by only varying four parameters: spindle rpm, depth of cut, chipload and feed. The ranges of each parameter were taken to accommodate for a wide variety of setups. The workpiece material is Aluminum 6061 and the cutter was a four inch diameter and four insert face milling cutter, uncoated carbide H10. The sampling frequency of the dynamometer is 2005 Hz. Calibration cutting test were carried out with one insert. Force data used for training the ANN is obtained from an average of five revolutions of the same insert at the same cutting conditions to accommodate for any experimental error due inhomogeneous workpiece material and tool wear. Data was preprocessed by taking the average of the first five cutting cycles.

3.4 Data Preparation Methods and Processing

As shown in Figure 10, the output of the dynamometer reading is a voltage signal. The voltage signals from the 36 conditions should be first multiplied by the gain which is a constant equal to 44.96 N/V. Moreover, drift should be subtracted. The cutting time is one second and that result in numerous cycles, an average of 5 cycles is taken to arrive at the average cutting force. An example of the technique and code used is shown in Appendix B.

3.5 Normalization Techniques

The simplest normalization technique is the Min–max normalization. Min–max normalization is best suited for the case where the bounds (maximum and minimum values) of the scores produced are known. In this case, we can easily shift the minimum and maximum scores to 0 and 1, respectively. However, even if

the matching scores are not bounded, we can estimate the minimum and maximum values for a set of matching scores and then apply the min–max normalization. Given a set of matching scores

$$\{s_k\},\ k=1, 2, \ldots, n \tag{18}$$

The normalized scores are given by

$$s_k = (s_k - s_{min}) / (s_{max} - s_{min}) \tag{19}$$

When the minimum and maximum values are estimated from the given set of matching scores, this method is not robust (i.e., the method is highly sensitive to outliers in the data used for estimation). Min–max normalization retains the original distribution of scores except for a scaling factor and transforms all the scores into a common range [0, 1].

Min. and Max values for the input and output parameters used are shown in Table 2.

Table 2 Min & Max values

Input / Output	Min	Max
RPM	1000	3000
DOC (1/1000 inch)	40	250
Chipload	2	8
Feed (inch/min)	2	24
Time (sec)	0.0005	0.06
F_x	-161.79	99.1774
F_y	-228.892	139.5648
F_z	-30.7034	51.17689

An example of the normalization done for the RPM is as follows:

$$\text{Normalized RPM} = (\text{RPM}-1000)/(3000-1000) \tag{20}$$

3.6 Partitioning the Training & Testing data

The data has been divided into training and validation data, where the training data is 75% and the remaining 25% of the data is used for validation. The data consists of 36 sets as mentioned earlier. The data is randomly divided as shown in Appendix C.

CHAPTER 4

FORCE PREDICTION USING ANNS

In this chapter, ANNs will be employed to predict the full force signal, the average, resultant and maximum forces as highlighted in
Figure 8.

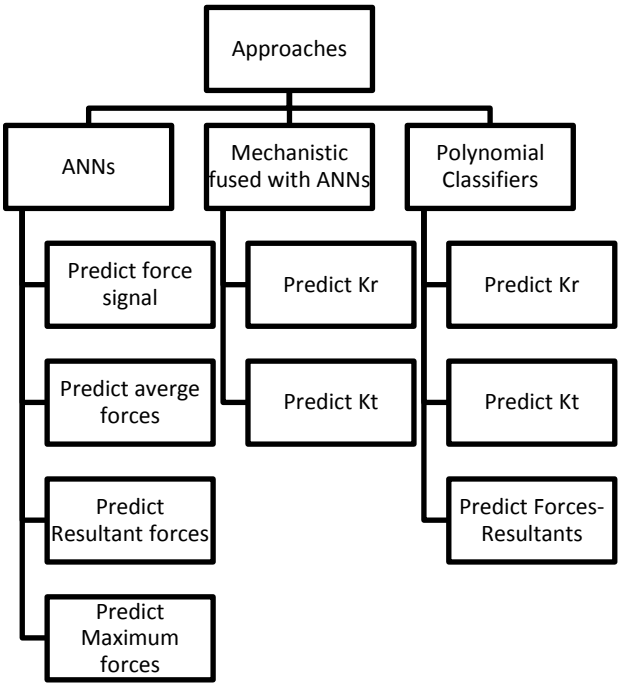

Figure 8 Road map for Chapter 4 – predicting full force signal

4.1 Full force signal over time Prediction

In this section, the three force components are predicted as function of time. To that effect, the ANNs used consist of an input layer with five neurons (corresponding to the five input parameters: revolution per minute, depth of cut, chipload, feed and time), an output layer with three neurons (corresponding to the output parameters: F_x, F_y & F_z as a function of time) and one or two hidden layers. Figure 9 shows a schematic of the network used to predict the variation of the cutting forces with time. Matlab's Neural Network tool box has been used.

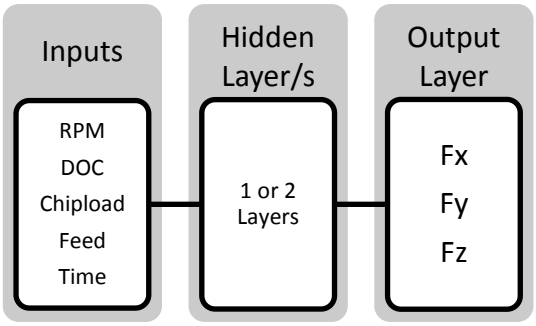

Figure 9 Network inputs and outputs to predict the cutting forces

4.1.1 Networks Tested

A wide range of network architectures such as the feed forward back propagation, cascade feed forward back propagation, radial basis, Elman back propagation and layer recurrent were investigated [25]. Runs with different number of hidden layers and/or number of neurons per hidden layer were performed to study the impact of these parameters on the predictions obtained. The networks were also trained using different training algorithms, such as Levenberg-Marquardt, resilient back propagation and scaled conjugate back propagation algorithms. Different transfer functions such as *tansig, logsig* and *purelin* were also considered [34]. In all cases, the predicted results presented are average values of five ANN runs to overcome the effect of any inconsistencies that may result due to the randomness of the ANN's initial guess.

4.1.2 Results

A Parametric study varying ANN architecture and variables was conducted for the purpose of finding an optimum ANN for this application. The details of the networks used are shown in appendix D. The details of the results obtained are presented next.

4.2 ANN Characteristics

4.2.1 ANN architecture

The network architectures that were tested are: feed forward back propagation (FFBP), radial basis (exact Fit), cascade feed forward back propagation, Elman back propagation and layer recurrent. The results obtained show that both Elman

back propagation and layer recurrent architectures require a longer time to converge compared to the other architectures. The results also show that the error obtained using the radial basis function is very high as shown in

Table 3. Therefore, the investigation was continued with FFBP. For the FFBP, the effect of varying the training function, the transfer function, the number of hidden layers, and the number of hidden neurons was investigated. The architectures of the various neural networks used in this study are shown in Appendix D. Figure 10 shows a comparison of the network architectures. The best prediction is that of network 1 (a 3 layer FFBP network, with *trainlm* as the training function and 10 neurons in the first layer and a transfer function of *logsig* and 5 neurons in the second layer and a *purelin* transfer function) for cutting condition C10. Networks 9, 10 and 11 predictions are also shown in Figure 10. Table 3 compares the NMSE output prediction using the 5 networks stated, the values presented are the average NMSE of predicting the 12 validation conditions (C2, C4, C8, C10, C14, C18, C19, C23, C26, C28, C33 and C35) as shown in Appendix C table 16. The lowest normalized mean square error (NMSE) obtained was for C10 case at a NMSE 4.8% predicted by network 1.

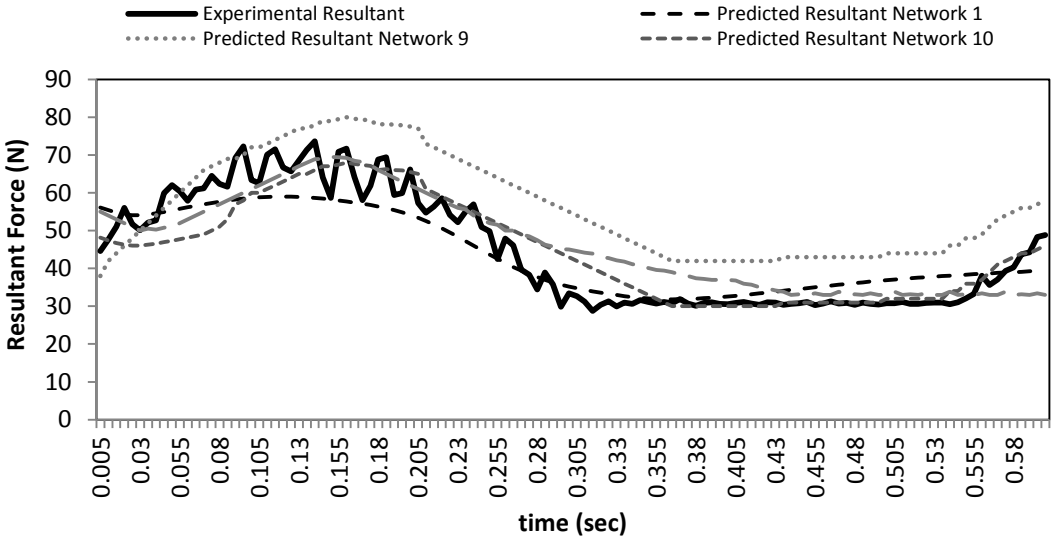

Figure 10 Predicted and measured resultant force for one insert cutting, C10 (RPM 1000, Depth of cut of 40, Chipload 8 and feed of 20) for networks 1, 9, 10 and 11

Table 3 Effect of Architecture

Net number	Type	Number of hidden layers	NMSE (%)
net 1	FFBP	3	4.83%
net 8	Radial basis	-	364.8%

net 9	Cascade forward back propagation	-	19.6%
net 10	Elman back propagation	-	11.5%
net 11	Layer Recurrent	-	21.3%

4.2.2 Training Algorithm

The effect of the type of training algorithm used was investigated next. *Levenberg-Marquardt (trainlm)*, resilient back propagation (*trainrp*) and scaled conjugate back propagation algorithms (*traincg*) were used. To compare training functions, FFBP networks with one and two hidden layers were used. Table 4 shows that the *Levenberg-Marquardt* algorithm with one and two hidden layers gave the lowest NMSE. The NMSE presented in table 4 is an average of the 12 validation conditions (C2, C4, C8, C10, C14, C18, C19, C23, C26, C28, C33 and C35) as shown in Appendix C table 16.

Table 4 Effect of type of training function on prediction for cutting condition C4

Net number	Type	Number of hidden layers	Training Function	NMSE (%)
net 1	FFBP	2	*Trainlm*	4.44
net 4	FFBP	1	*Trainlm*	4.55
net 12	FFBP	1	*Trainrp*	6.85
net 13	FFBP	1	*Trainscg*	6.03
net 14	FFBP	2	*Trainrp*	4.35
net 15	FFBP	2	*Trainscg*	6.03

4.2.3 Number of hidden layers

To study the effect of the number of hidden layers on the accuracy of the ANN prediction, FFBP networks with one and two hidden layers were investigated. Since the predictions obtained from varying the training function (Table 4) showed that *trainlm* gave the best predictions, it was used here for all the cases considered. The *tansig* transfer function was selected as the use of this function allows a nonlinear relationship between the input and the output, moreover, this function is self limiting and hence the output cannot grow infinitely large or small.

Table 5 compares the results obtained using one and two hidden layers. The NMSE presented in table 5 is an average of the 12 validation conditions (C2, C4, C8, C10, C14, C18, C19, C23, C26, C28, C33 and C35) as shown in Appendix C table 16.For the case shown in

Table 5, the predictions show that a FFBP with two hidden layers result in the lowest error.

Table 5 Effect of the number of layers on the FFBP predictions

Net. number	Number of hidden layers	Hidden Layer 1		Hidden Layer 2		Training function	NMSE (%)
		Transfer function	Number of neurons	Transfer function	Number of neurons		
net 2	1	*Tansig*	20			*trainlm*	6.08
net 4	1	*Tansig*	10			*trainlm*	5.00
net 18	2	*Tansig*	10	*tansig*	10	*trainlm*	4.12
net 19	2	*Tansig*	10	*tansig*	20	*trainlm*	9.85
net 20	2	*Tansig*	10	*tansig*	5	*trainlm*	4.08
net 22	2	*Tansig*	20	*tansig*	10	*trainlm*	8.87

4.2.4 Transfer functions

To investigate the effect of varying the transfer function, 3 single hidden layer networks with the same number of neurons in the hidden layers and same transfer function are compared and 2 2 hidden layer networks are compared as in Table 6. The NMSE presented in table 6 is an average of the 12 validation conditions (C2, C4, C8, C10, C14, C18, C19, C23, C26, C28, C33 and C35) as shown in Appendix C table 16.

Figure 11 compares network 1 and network 20 for condition C4. Both networks have the same 3 layer architecture, number of layers, number of neurons, training function; the only difference is in the transfer function. Network 1 had *logsig* as the first transfer function and *purelin* as the second. Network 20 has both transfer functions as *tansig*. The NMSE obtained for C4 using network 20 is slightly lower than that of network 1 (3.56% vs. 4.44%), whereas the *logsig* transfer function in a single hidden layer network gave the lowest errors.

Table 6 Effect of varying the transfer function

Net number	Number of	Hidden Layer 1		Hidden Layer 2		Training function	NMSE (%)
		Transfer	Number	Transfer	Number		

	hidden layers	function	of neurons	function	of neurons		
net 4	1	*Tansig*	10			*trainlm*	14.85
net 6	1	*Purelin*	10			*trainlm*	21.52
net 25	1	*Logsig*	10			*trainlm*	10.49
net 1	2	*Logsig*	10	*Purelin*	5	*trainlm*	4.44
net 20	2	*Tansig*	10	*Tansig*	5	*trainlm*	3.65

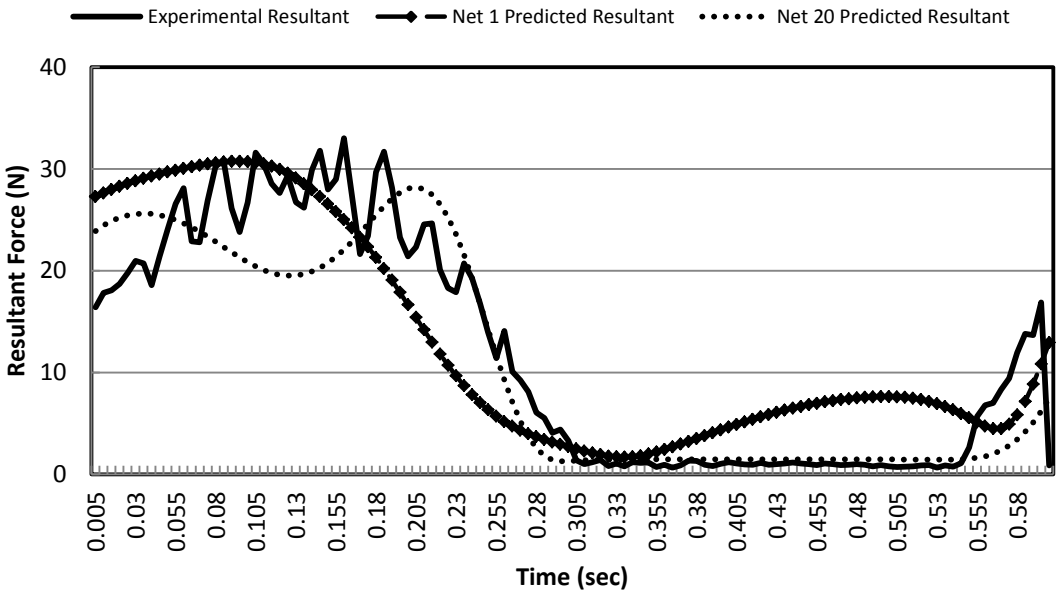

Figure 11 Predicted and measured resultant force for one insert cutting, C4 (RPM 1000, Depth of cut of 40, Chipload 4 and feed of 8)

4.2.5 Number of neurons in the hidden layers

To study the effect of the number of hidden neurons, several networks with one hidden layer using the same training function (*trainlm*) and the same transfer function (*tansig*) were compared. For cutting conditions C10, the network with 5 hidden neurons has the lowest NMSE at 9.70%. Table 7 shows the average NMSE prediction if the 4 networks of the 12 validation conditions (C2, C4, C8, C10, C14, C18, C19, C23, C26, C28, C33 and C35 as shown in Appendix C table 16).

Table 7 Effect of the number of hidden neurons in a FFBP type ANN with one hidden layer

Net number	Number of hidden layers	Hidden Layer 1		Training function	NMSE (%)
		Transfer function	Number of neurons		

net 2	1	*Tansig*	20	*trainlm*	16.45
net 3	1	*Tansig*	5	*trainlm*	9.7
net 4	1	*Tansig*	10	*trainlm*	12.43
net 5	1	*Tansig*	40	*trainlm*	39.3

Table 8 shows the effect of the number of hidden neurons for networks of the same architecture with the same training function and transfer function using two hidden layers. In this case, the network with 10-5 hidden neurons resulted in the lowest NMSE. The NMSE presented in table 8 is an average of the 12 validation conditions (C2, C4, C8, C10, C14, C18, C19, C23, C26, C28, C33 and C35) as shown in Appendix C table 16.

Table 8 Effect of the number of hidden neurons in a FFBP type ANN with two hidden layers.

Net number	Number of hidden layers	Hidden Layer 1	Hidden Layer 2	NMSE (%)
net 18	2	10	10	14.74
net 19	2	10	20	24.59
net 20	2	10	5	9.20
net 21	2	10	40	57.64
net 22	2	20	10	33.11

In conclusion, it was found that 2 layer networks with the *trainlm* training function gives the lowest errors.

4.3 Average Force Prediction

Knowledge of the average forces produced in milling operations can provide a fast estimate of the forces needed for different machining scenarios and can be used in different analyses of the machining process. Tests have been carried out using 12 data sets (C2, C4, C8, C10, C14, C18, C19, C23, C26, C28, C33 and C35 as shown in Appendix C table 16) and 4 network architectures. Figure 12 shows the predictions obtained compared to the experimental results. Network 29 gave the lowest NMSE at 1.6% on average. **Table 9** shows a comparison between the NMSE obtained using each of the networks considered.

Results

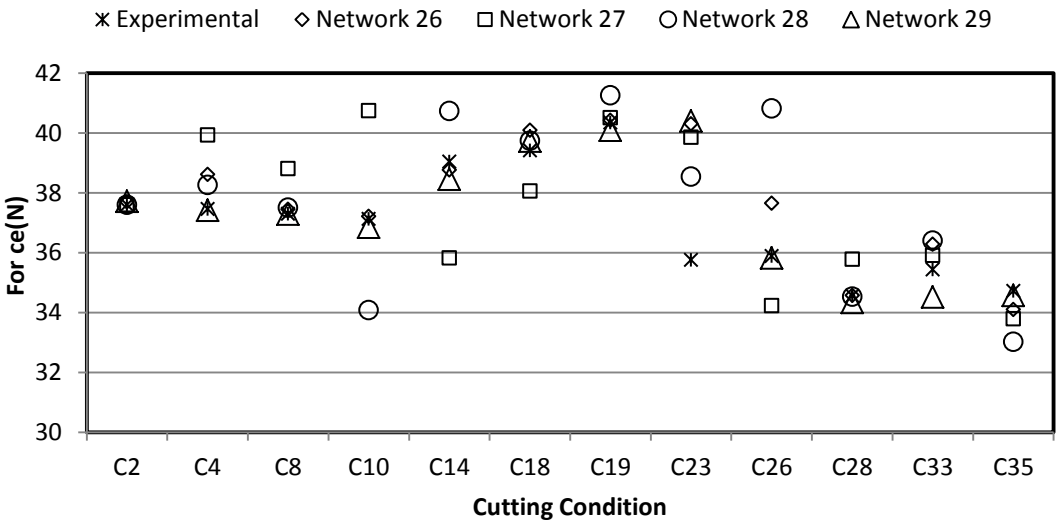

Figure 12 Sample predictions of average F_x (over time)

Table 9 Prediction of average forces (F_x, F_y and F_z) using tansig transfer function and trainlm training function on average over the 12 validation conditions

Net number	Type	Number of layers	Hidden Layer 1	Hidden Layer 2	NMSE (%)
			Number of neurons	Number of neurons	
net 26	FFBP	2	4		2.9
net 27	FFBP	3	8	16	4.2
net 28	Radial Basis				3.9
net 29	FFBP	2	8		1.6

4.4 Resultant Force Prediction

Thirty-six tests have been carried out using 12 data sets for validation (C2, C4, C8, C10, C14, C18, C19, C23, C26, C28, C33 and C35 as shown in Appendix C table 16) and three Networks- detailed in Table 10 : 30, 31 and 32 are compared with experimental data as shown in Figure 13. NMSEs stated in table 10 are the average NMSE of the 12 validation data sets.

Results

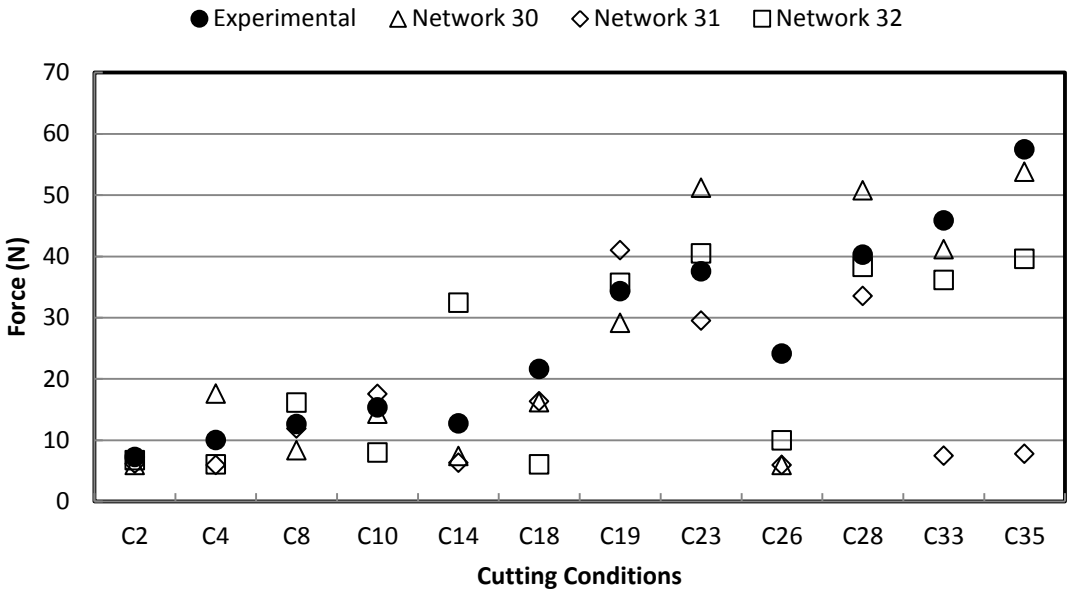

Figure 13 Predicted resultant forces as function of condition

Table 10 Resultant forces prediction- *Tansig/logsig* function, *Trainlm* Training function

Net No.	Type	No. of layers	No. of Neurons	No. of Neurons	Transfer function	NMSE (%)
Network 30	FFBP	2	8		tansig	16.1
Network 31	FFBP	3	8	4	tansig	37.7
Network 32	FFBP	2	8		logsig	20.5

Network 30- a 2 layer FFBP networks with 8 neurons in the hidden layer and *trainlm* as the training function and *tansig* as the transfer function has the lowest NMSE of ~16%.

4.5 Maximum Force Prediction

Predicting the maximum forces is very useful in machine tool selection and simulation of the worst case scenario: conservative analysis. The network has 4 inputs and 3 outputs. The outputs are the maximum forces in the X, Y and Z directions respectively. Thirty-six tests have been carried out using 12 data sets and three Networks: 33, 34 and 35 are compared as shown in Figure 14. Network 33 gave the lowest NMSE at 4.10%.

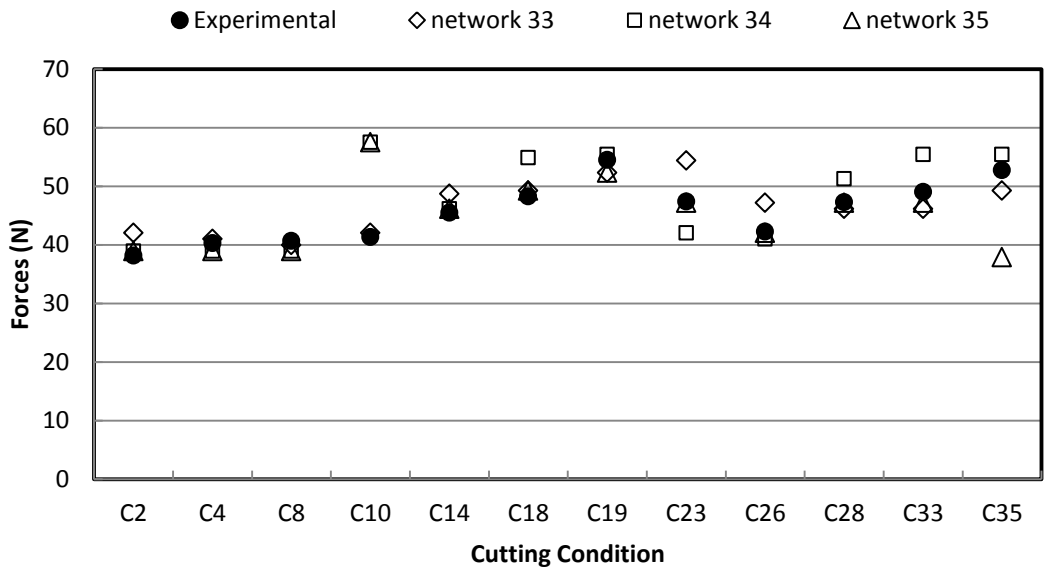

Figure 14 F_x Predicted maximum forces as function of condition

4.6 Force prediction using hybrid model: Mechanistic & ANNs prediction

As mentioned earlier, mechanistic methods have a high accuracy of predicting cutting forces; however, their main drawback is their lack of generality. To address this problem, ANNs were used to predict the cutting coefficients. An investigation was performed to investigate the effect of varying ANN parameters and architectures on the prediction accuracy and to determine the optimum ANN parameters and architecture. Experimental and ANN predicted cutting coefficients are compared.

The cutting pressure coefficients have been calculated using the code in Appendix B and the resulting K_r and K_t tabulated in Appendix E

4.6.1 Prediction of cutting coefficients and forces from calculated cutting coefficients

Figure 15 and Figure 16 show a comparison between the calculated and the predicted values of the tangential cutting coefficient (K_t) and the radial cutting coefficient (K_r) respectively, using a number of different networks. All networks

used in this case were FFBP with *Levenberg-Marquardt* training function. Table 11 shows the average NMSE of the predictions of the 12 validation conditions (C2, C4, C8, C10, C14, C18, C19, C23, C26, C28, C33 and C35 as shown in Appendix C table 18), which shows that network 37 gives the lowest errors.

Table 11 NMSE of Networks 36, 37 and 38

	Network 36	Network 37	Network 38
K_t	47.54%	16.64%	57.83%
K_r	21.98%	8.96%	35.23%

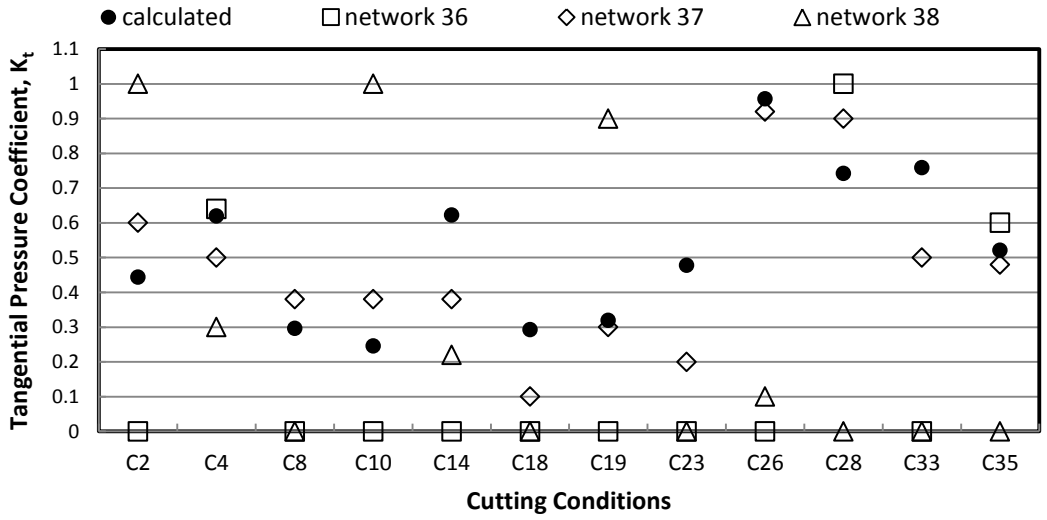

Figure 15 Comparison between the calculated values of the tangential pressure coefficient and the predictions obtained using various ANN architectures for a variety of cutting conditions.

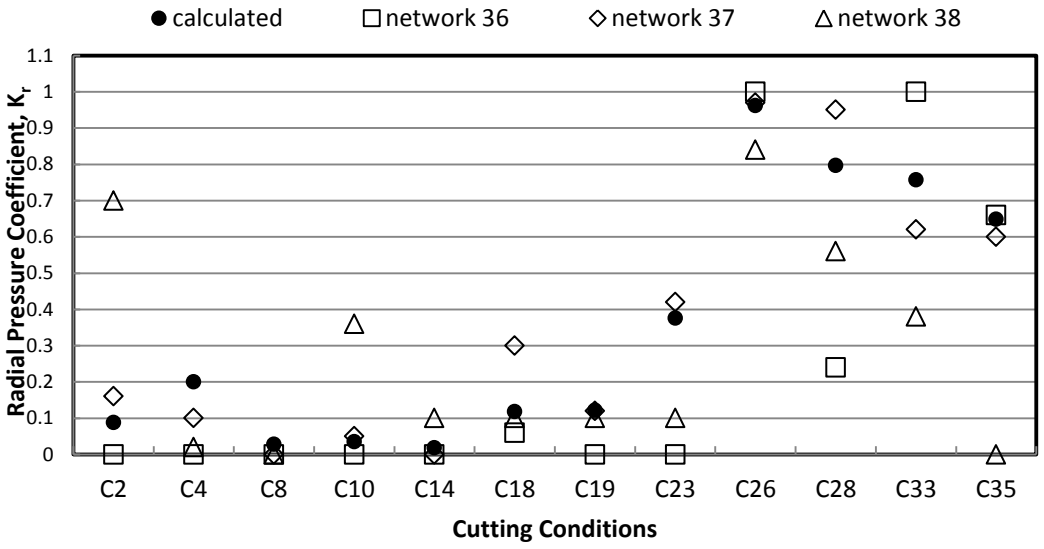

Figure 16 Comparison between the calculated values of the radial pressure coefficient and the predictions obtained using various ANN architectures for a variety of cutting conditions.

Network 37 is a 2 layer feed forward back propagation network with a trainlm transfer function, a logsig transfer function and 8 neurons in the hidden layer. Figure 17 shows the resultant force calculated from eq. 3-6, 8 and 9 from the predicted K_t and K_r values.

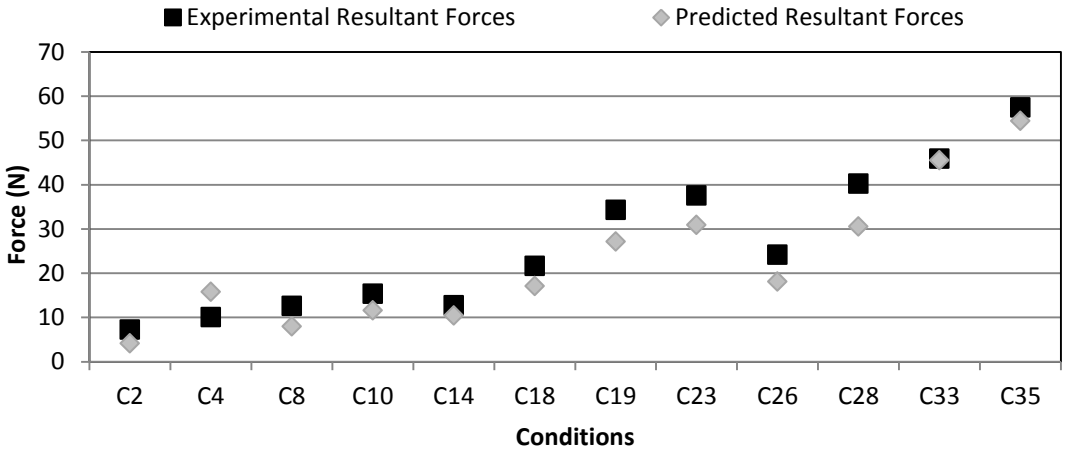

Figure 17 Resultant force calculated from Kt and Kr for network 37

4.6.2 Prediction of cutting coefficients from the cutting forces

Another possible approach is to use the three force components to predict the cutting coefficients K_r and K_t. This approach is helpful in simulating the cutting process when the force variation is needed, e.g. finite element analysis. Alternatively, if the cutting force components are collected experimentally, they could be used as inputs to the network to predict the cutting coefficients, rather than using the cutting parameters. Figure 18 and Figure 19 compare the predictions obtained using a variety of ANNs with the values calculated using average forces for several cutting conditions.

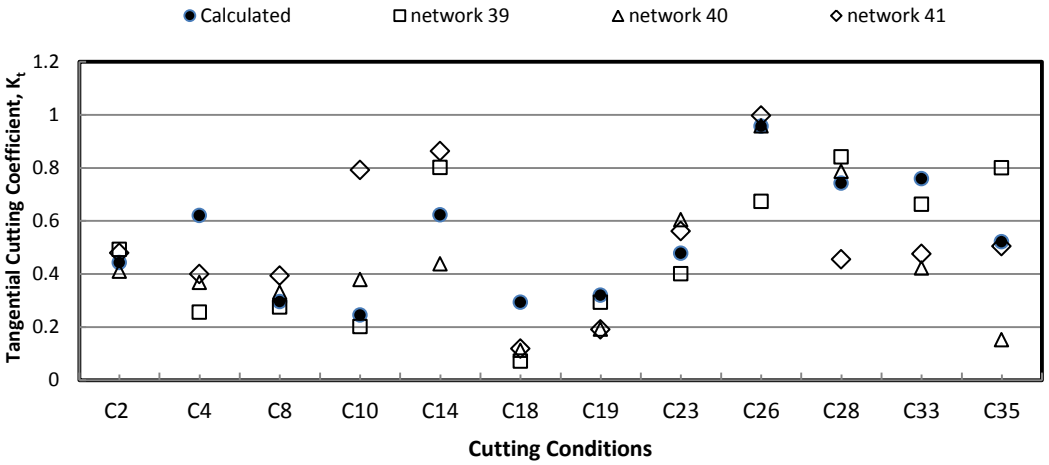

Figure 18 Comparison between the values calculated for the tangential pressure coefficient using the average forces and the predictions obtained using various ANN architectures for a variety of cutting conditions.

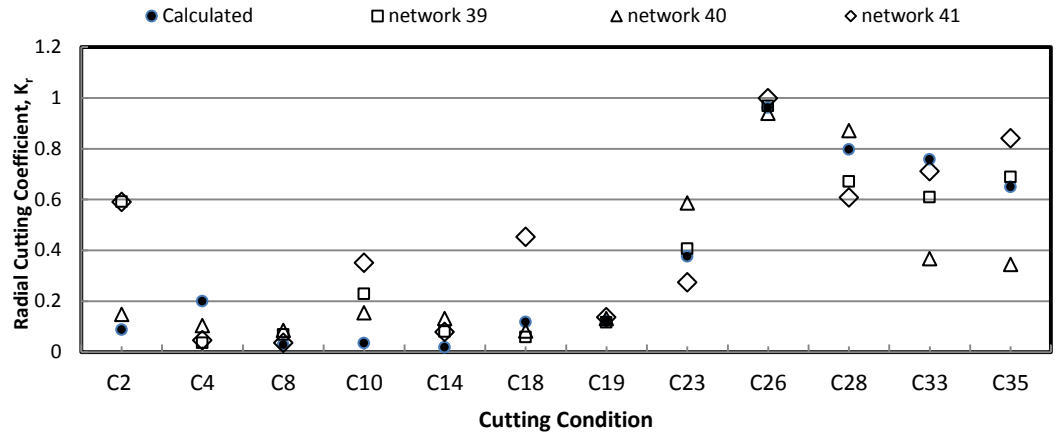

Figure 19 Comparison between the values calculated for the radial pressure coefficient using the average forces and the predictions obtained using various ANN architectures for a variety of cutting conditions.

For K_t and K_r predictions, networks 39, 40 and 41 gave comparable NMSE. Table 12 shows the NMSEs of networks 39, 40 and 41. The 3 networks are feed forward back propagation networks with trainlm training function, networks 39 and 40 are 2 layer networks, the first with 10 neurons in the hidden layer and a tansig transfer function, the latter has 8 neurons in the hidden layer and a logsig transfer function. Network 41 is a 3 layer network, with 8 neurons in the first hidden layer and a logsig transfer function and 4 neurons in the second hidden layer and a purelin transfer function. The forces can be extracted by using the predicted K_t and K_r values and applying eq. 3-6, 8 and 9.

Table 12 NMSE of Networks 39, 40 and 41

	Network 39	Network 40	Network 41
K_t	18.31%	19.03%	23.00%
K_r	17.48%	16.80%	21.95%

4.7 Summary of Results for force prediction using ANNs

Artificial neural networks were used to estimate the cutting forces from the cutting parameters at first. Then ANNs where used with mechanistic methods to predict the cutting pressure coefficients which were then used to calculate the cutting forces. It can be claimed that the comparison of the predictions obtained from the neural models with the experimental results confirms the potential of the model to predict the cutting forces. The approach ensures estimation of the cutting forces in real time which is needed for simulation of different aspects of the machining process such as fixture configuration selection and optimization of cutting parameters which would enable leaner and more efficient machining process.

For the cutting conditions considered, the best cutting forces predictions using the cutting parameters as inputs were obtained using FFBP networks with the *Levenberg-Marquardt* training function algorithm and the *tansig* transfer function. This ANN produced an average NMSE of ~4% over the 12 validation conditions. When predicting the resultant and the maximum forces, the radial basis network produced the lowest error compared to the experimental values. When using the fused ANNs with mechanistic methods, FFBP networks where investigated and where found to give NMSE as low as 8%.

CHAPTER 5

FORCE PREDICTION USING POLYNOMIAL CLASSIFIERS

5.1 Predicting Cutting forces using Polynomial Classifiers

Despite the many advantages of neural networks and their ability to obtain adequate results, the repeatability of their predictions is always a concern for both designers and users. Cutting force predictions can be obtained with neural networks depending on the type of network, the number of hidden layers and the training algorithm used. Furthermore, one should remember that the initial weights chosen by any neural network are random in nature and therefore one should expect slightly different predictions if the same neural network is applied numerous times (although this can be remedied by taking the average results obtained from several runs). Finally it should be noted that the methods used by the neural networks are iterative ones rather that direct solutions. To address the above-mentioned shortcomings of neural networks, the polynomial classifiers (PC) method is considered next.

The same parameters used with ANN will be used in the investigation involving polynomial classifiers. As before, data from the same 36 cutting conditions were used for training and testing purposes. 75% of the data was used for training and the remaining 25% used for validation and testing. MATLAB Software was once again used to construct, train and test the classifiers [34].

To predict the cutting forces, the use of first, second and higher order classifiers will be investigated. For each case, the predictions obtained will be compared to experimental data and the NMSE will be used to gauge the effectiveness of the polynomials used.

5.1.1 Predicting Resultant Forces using first, second and third Order PC

PCs were also used to predict the resultant forces. Figure 20 compares ANNs' prediction with first, second and third order PCs. Table 13 shows the NMSE of ANNs and PCs.

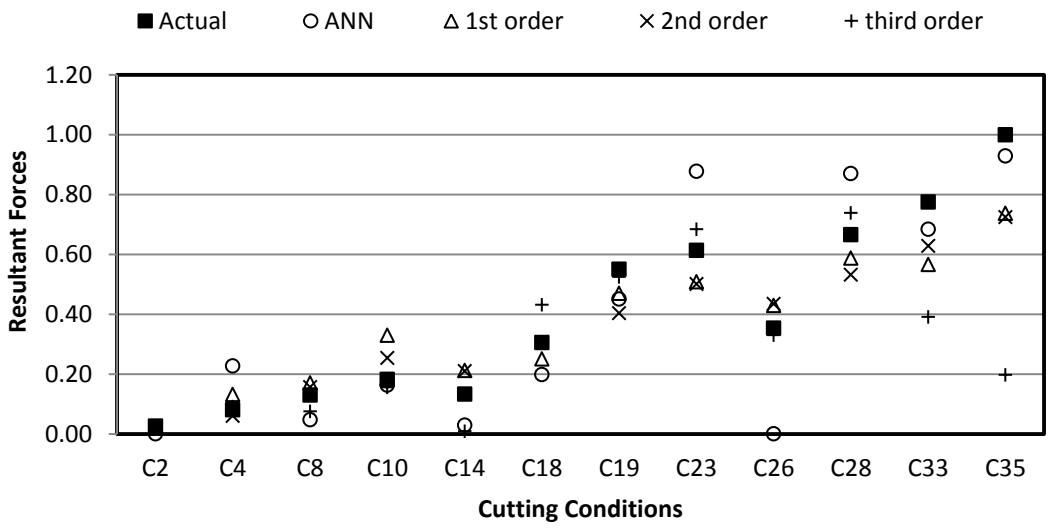

Figure 20 Resultant Forces Prediction with ANNs and PCs

Table 13 NMSE of ANNs and PCS

	ANNs	1st Order	2nd Order	3rd Order
Resultant Forces	16.12%	12.23%	11.98%	26.64%

5.1.2 Predicting Average Forces using first, second and third Order PC

PCs were also used to predict the average forces. Figure 21 compares ANNs' prediction with first, second and third order PCs. Table 14 shows the NMSE of ANNs and PCs.

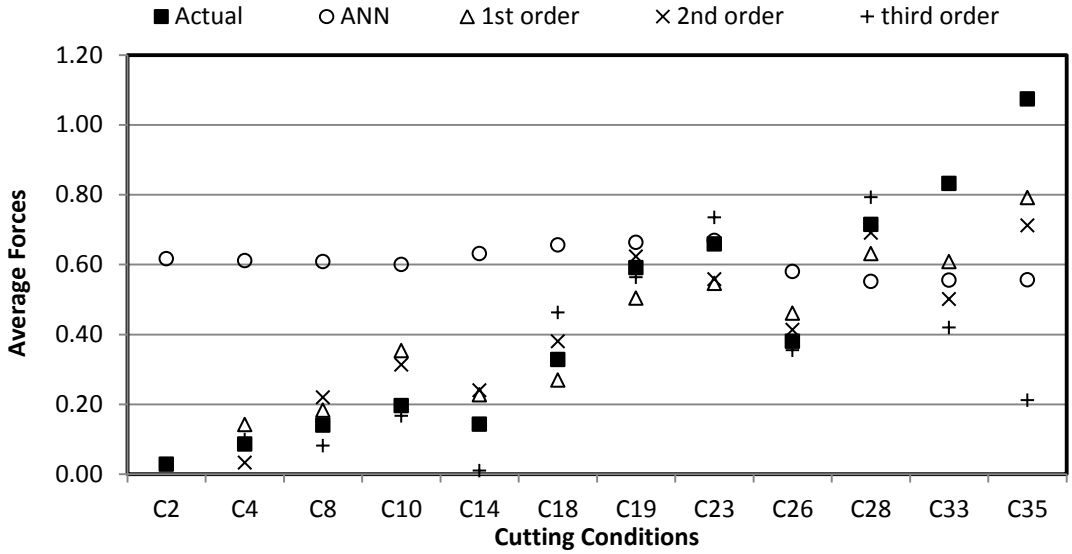

Figure 21 Average Forces Prediction with ANNs and PCs

Table 14 NMSE of ANNs and PCS

	ANNs	1st Order	2nd Order	3rd Order
Average Forces	1.6 %	13.13%	15.86%	28.60%

5.1.3 Predicting Max Forces using first, second and third Order PC

PCs were also used to predict them forces – in X direction. Figure 21 compares ANNs' prediction with first, second and third order PCs. Table 15 shows the NMSE of ANNs and PCs.

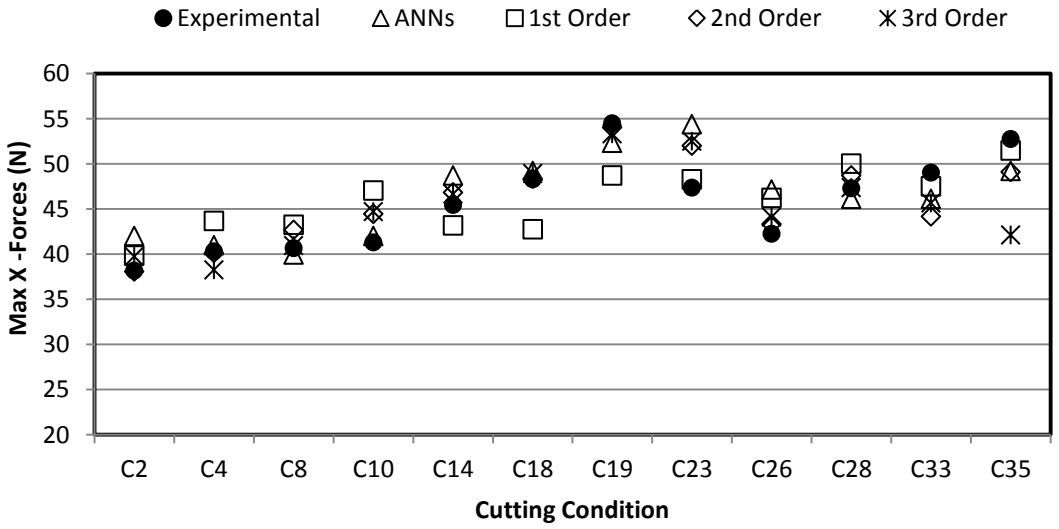

Figure 22 Max. Forces Prediction with ANNs and PCs

Table 15 NMSE of ANNs and PCS

	ANNs	1st Order	2nd Order	3rd Order
Average Forces	4.1 %	6.88%	4.93%	7.43%

5.2 Predicting Coefficients K_t & K_r using Polynomial Classifiers

5.2.1 Predicting Cutting Coefficients using First Order PC

As there are 4 parameters to consider as input to the classifier, which are: RPM, Depth of Cut DOC, Chipload A_c and feed F, resulting in a full first order PC.

$$P_1(X) = [1, RPM, DOC, Ac, F] \quad \textbf{(21)}$$

The output of the logarithm is the K_r (the radial pressure coefficient) and the K_t (the tangential pressure coefficient) values. The predictions obtained using the first order PC were compared to the experimental data and were found to score an NMSE of the order of 18%. Figures 23 and 24 show the output of the first order PC for predicting K_r and K_t respectively. 5 plots are shown; a first order polynomial classifier was used with only one parameter, in order to determine the most influential parameter. As can be seen from Figure 23 and Figure 24 and table 16, DOC has the most influence; this has been verified by simulating the PC with one term only, as in Figure 23, DOC resulted in the lowest error and will therefore be emphasized in higher order PC in an attempt to get a more accurate prediction. Table 16 lists the NMSE of the first order parameters.

Table 16 NMSE of First and Second Order PC

	RPM	DOC	A_c	Feed	Full First Order	Second Order
K_r	33.37%	17.74%	33.48%	33.53%	19.07%	11.75%
K_t	23.48%	19.07%	21.46%	21.45%	17.19%	17.58%

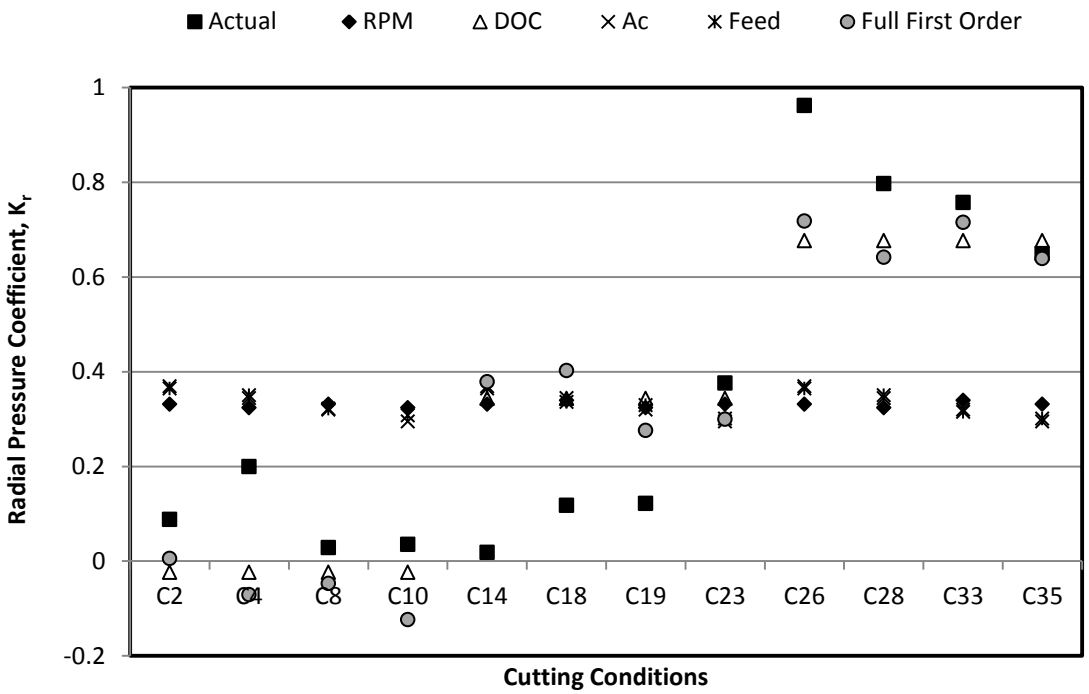

Figure 23 Predicting K_r using a separate and full first order PC

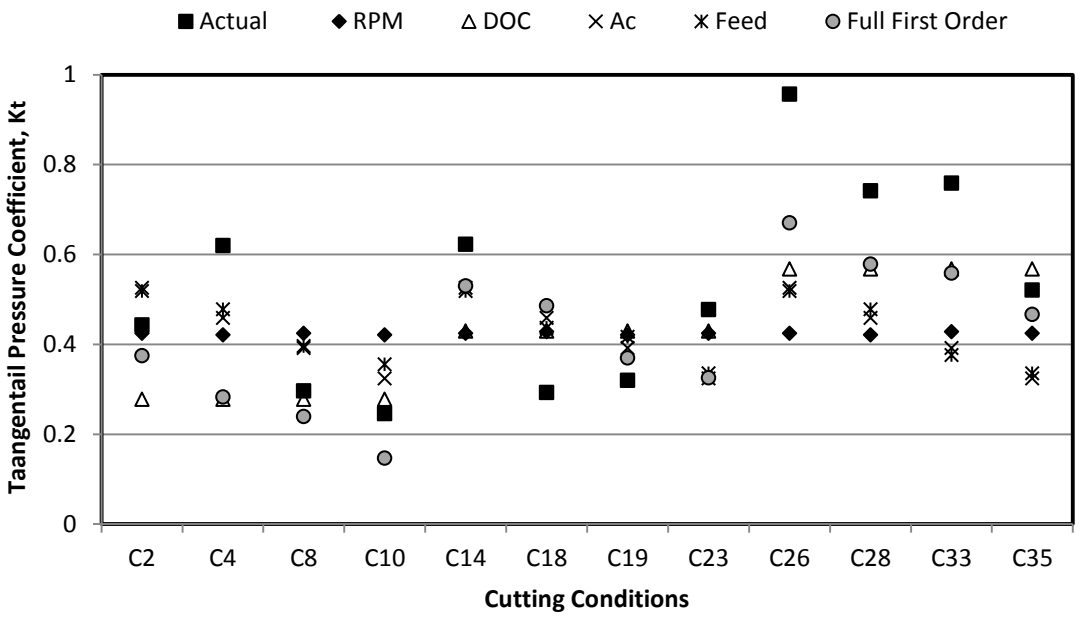

Figure 24 Predicting K_t using separate and full first order PC

In an attempt to reduce the errors, higher orders will be investigated.

5.2.2 Predicting Cutting Coefficients using Second Order PC

In order to reduce the error obtained using the first order PC, second order PC was attempted. In this case, the input parameters include the first order terms shown in addition to the square of each of these terms and the cross multiplication of each two of these terms to give the second order as shown below:

$$\boldsymbol{P_2(X)} = [\mathbf{1}, RPM, DOC, Ac, F, (RPM)^{\mathbf{2}}, (DOC)^{\mathbf{2}}, (Ac)^{\mathbf{2}}, (F)^{\mathbf{2}}, RPM * DOC, RPM * Ac, RPM * F, DOC * Ac, DOC * F, Ac * F] \quad \textbf{(22)}$$

A best NMSE of 11.7% was reached with the employment of the second order. An improvement compared to the first order. Results are sown in Figure 25 and Figure 26 and Table 16.

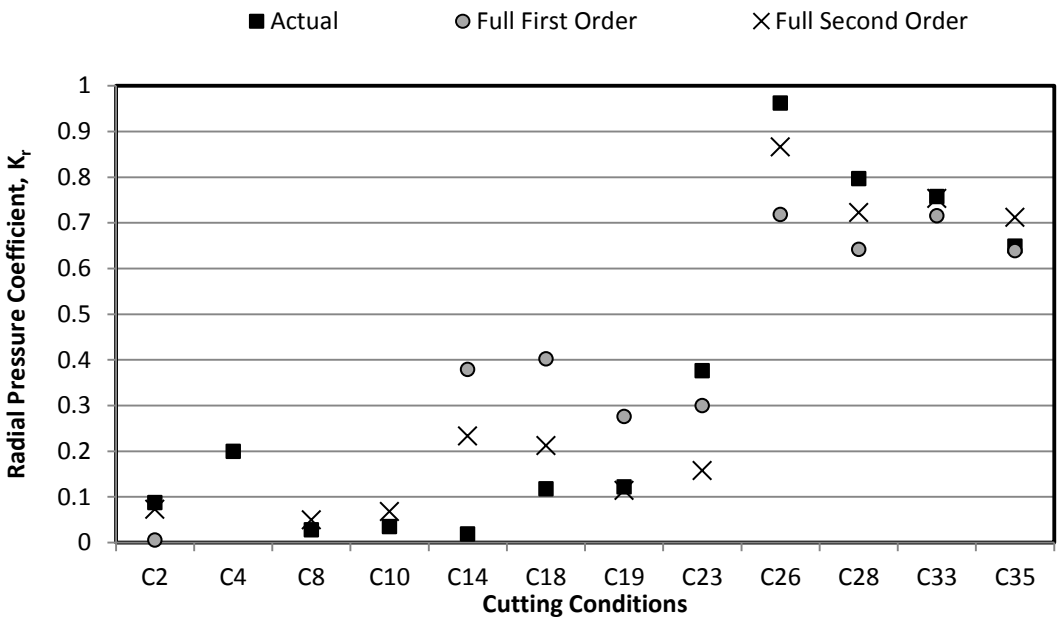

Figure 25 Second order PC K_r

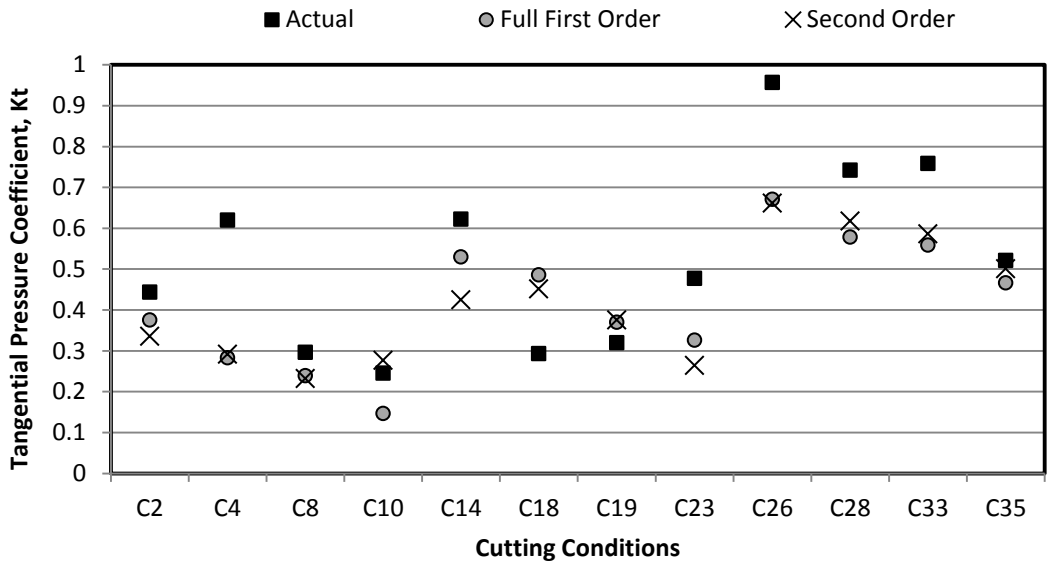

Figure 26 Second order PC K_t

Emphasized second order where only the DOC terms are kept as below:

$$\boldsymbol{P_2(X) = [1, DOC, (DOC)^2, RPM * DOC, DOC * Ac, DOC * F]} \quad \textbf{(23)}$$

yields an NMSE of 8.85%

5.2.3 Predicting Cutting Coefficients using Third Order PC

In attempt to lower the NMSE, higher order terms of the third order will be added to the polynomial classifier to yield more accurate predictions. The equation below shows the added higher order terms:

$$P_3(X) = [1, RPM, DOC, Ac, F, (RPM)^2, (DOC)^2, (Ac)^2, (F)^2, RPM * DOC, RPM * Ac, RPM * F, DOC * Ac, DOC * F, Ac * F,\ RPM * DOC * Ac, RPM * DOC * F, DOC * Ac * F, RPM * DOC^2, RPM * Ac^2, RPM * F^2, DOC * Ac^2, DOC * F^2, Ac * F^2, DOC * RPM^2 *, Ac * RPM^2, F * RPM^2, Ac * DOC^2, F * DOC^2, Ac * F^2\ , (RPM)^3, (DOC)^3, (Ac)^3, (F)^3 \quad \textbf{(23)}$$

The corresponding NMSE obtained is 6.45%. Plots of predictions are shown in Figure 27 and Figure 28.

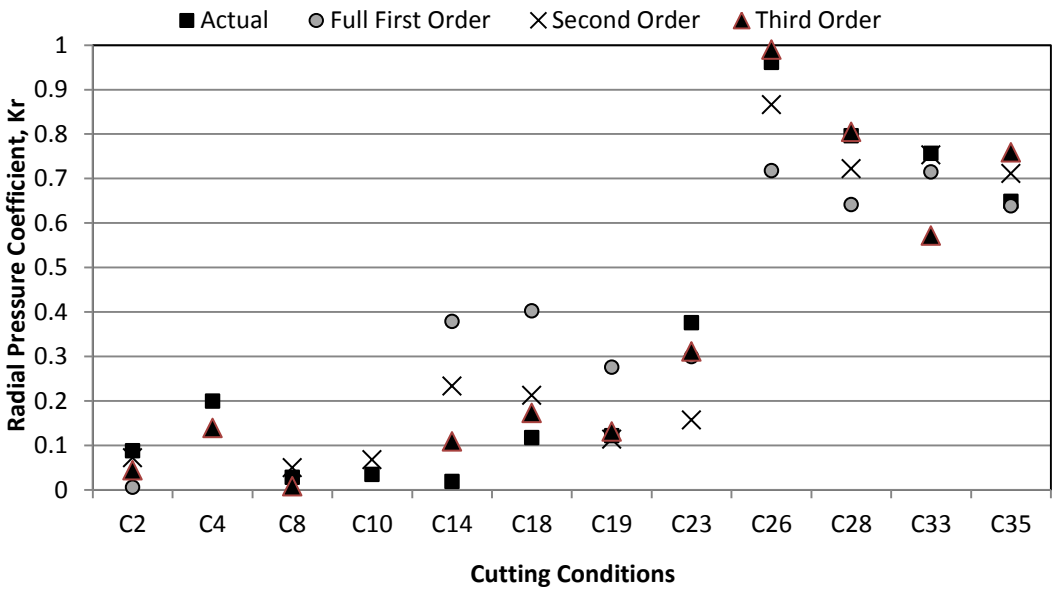

Figure 27 Third order PC K_r

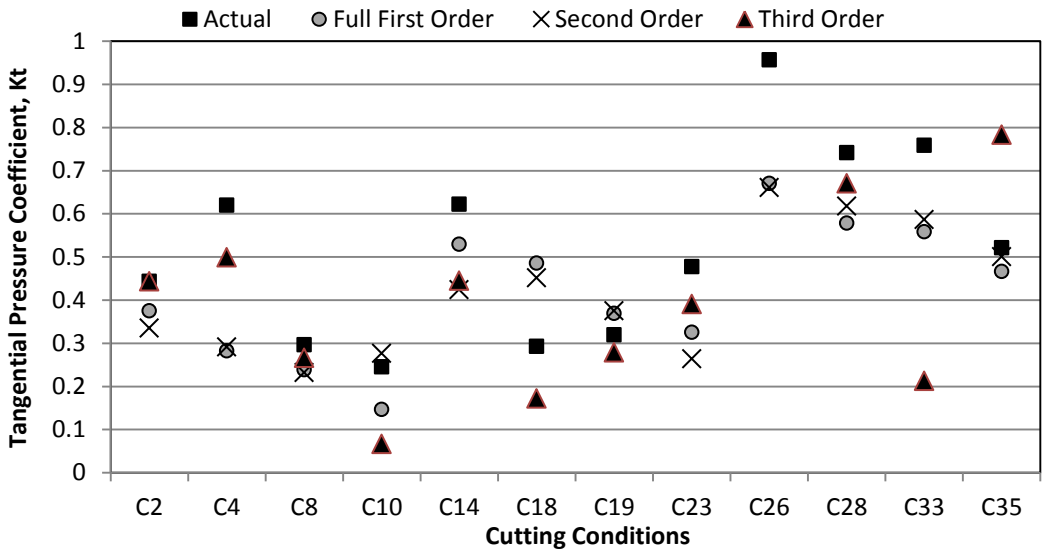

Figure 28 Third order PC K_t

5.3 Comparative study of force prediction methods

Chapters 4 and 5 have covered 3 different force prediction approaches: first is using ANNs to predict forces, using Hybrid techniques to predict the cutting pressure coefficients and using PCs to predict the cutting pressure coefficients. **Figure 29** below shows the comparison between experimental results and the predicted obtained using ANNs and PCs.

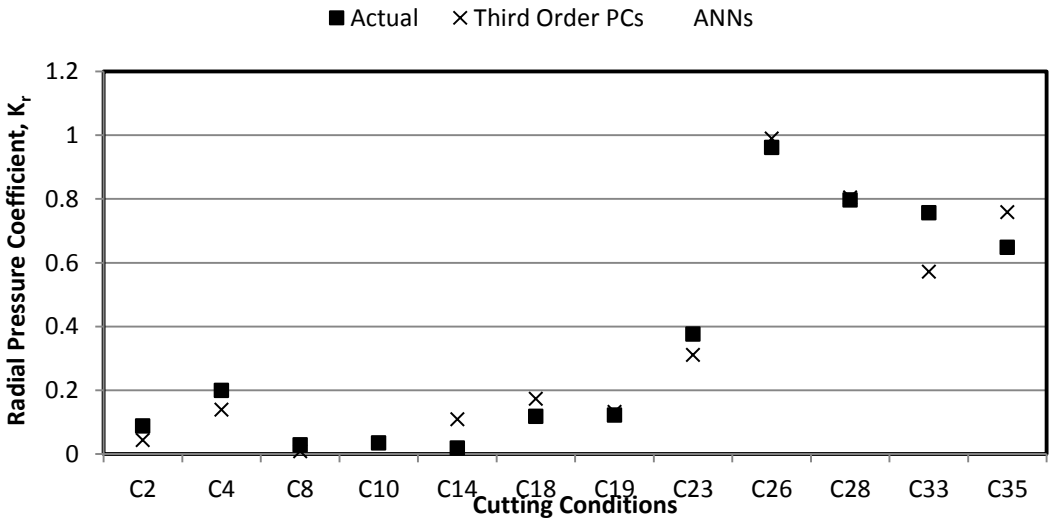

Figure 29 Third order PCs and ANNs compared with experimental

The use of ANNs scored a NMSE of 6.7%, whereas the third order PC scored 7%. Another run was carried out for the Tangential cutting pressure coefficient, shown in Figure 30 below.

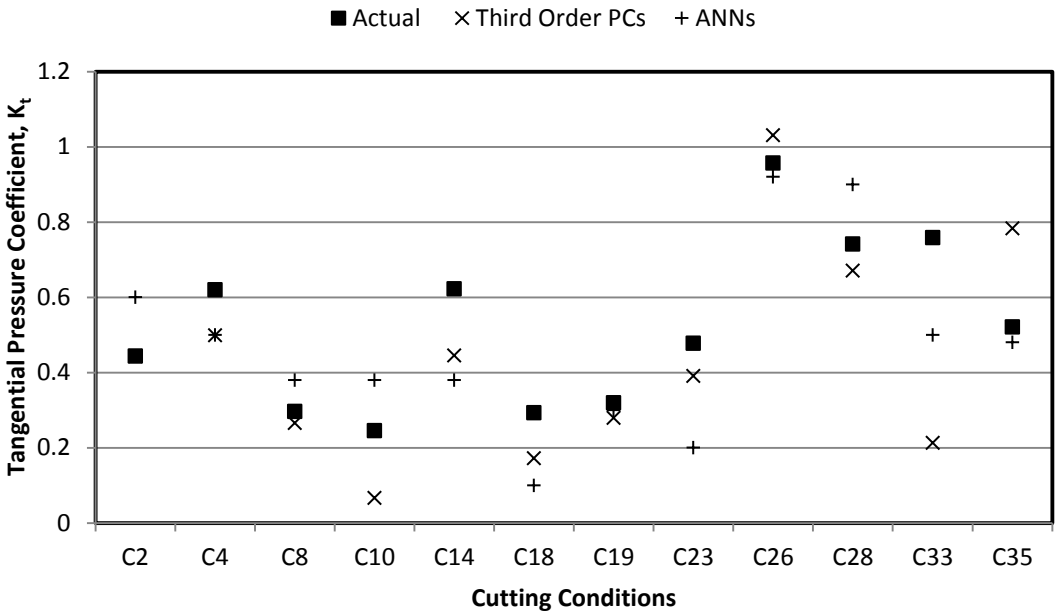

Figure 30 Third order PCs and ANNs compared

The ANNs scored a NMSE of 8.85%, whereas the third order PC scored 8%. Hence, both prediction techniques gave very similar accuracies.

CHAPTER 6

CONCLUSION & FUTURE WORK

Conclusion

This work presented the successful implementation and development of artificial neural networks and polynomial classifiers to predict the forces and the cutting pressure coefficients in end milling. Different neural network architectures using a variety of training functions were used. Training was performed on certain cutting conditions while the prediction was done for different parameters. The following summarizes the results:

1- Artificial neural networks can be used as efficient tool in predicting the full force signal, average forces, maximum forces and the resultant forces for cutting cases other than those used in the training of the network.
2- The predictions obtained are affected by the input parameters, network architecture, number of hidden neurons and training function used.
3- The PC method can lead to repeatable predictions for the cutting pressure coefficients. First and Second order classifiers seem to give the best predictions compared to experimental data.
4- The predictions obtained using PCs are compared to those of ANNs.

Future Work

1- Using ANN to predict the force signal by expanding the input parameters to different materials and other inserts
2- Using GMDH to better select the higher order terms be added to second degree PC leading to the best force prediction.

The same methodology can be applied to different applications in a manufacturing facility, such as prediction of metallurgical properties in a hardening facility, where certain parameters such as, pressure, volume of gases, temperature, current and voltage can be inputs to the ANN or PCs and the output can be the case depth and hardness. Hence, time, effort and money are saved, as no numerous experiments are required.

REFERENCES

[1] K. D. Ehmann, "Machining Process modeling: A Review", *Journal of Manufacturing Science & Technology,* vol. 119, pp. 655-663, November 1997.

[2] C. A. Luttervelt, T. H. C. Childs, I. S. Jawahir, F. Klocke, and P. K. Venuvinod, "Present Situation and future trends in modeling of machining operations", in *Annals of the CIRP,* vol. 47, 1998, pp. 2.

[3] F. Koenigsberger and J. P. Sabberwal, "An Investigation into the cutting force pulsation during milling operations", *International Journal of Machine Tool Design and Research,* vol. 1, pp. 15-33, 1961.

[4] B. V. Coelho, "Experimental Evaluation of Cutting force parapeters applying mechanistic model in orthogonal milling", *Journal of the Brazilian Society of Mechanical Science and Engineering* , vol. XXV, pp. 247-253, 2003.

[5] K. D. Jayaram, "Estimation of the Specific Cutting pressures for Mechanistic cutting force models", *International Journal of Machine Tool & Manufacture,* vol. 41, pp. 265-281, 2001.

[6] R. J. Saffar, , "Simulation of three dimension cutting force and tool deflection in the end milling operation based on finite element method", *Simulation model and practise theory,* vol. 16, pp. 1677-1688, 2008.

[7] M. Wan, "A novel cutting force modelling method for cylindrical end mill", *Applied Mathematical Modeling*, vol. 34, pp. 823-836, 2010.

[8] I. M. Deiab, *Effect of fixture dynamics on the face milling process*, Ontario, MacMaster University, 2003.

[9] D. A. Fu, "A Mechanistic Model for Prediction of the Force System in Face Milling Operations", *Journal of Engineering for Industry* , pp. 81-88, 1984.

[10] E. Kim and K. F. Ehmann, "A cutting force model for face milling operations", *International Journal of Machine Tools and Manufactures,* vol. 33, pp. 651-673, 1993.

[11] F. Gu, *Prediction of cutting forces and surface errors in face milling with generalized cutter and workpiece geometry*, University of Illinois, 1993.

[12] I. S. G. Kang, J. S. Kim, J. H. Kim, M. C. Kang and Y. W. Seo, "A mechanistic model of cutting force in the micro end milling process". *Journal of Materials Processing Technology*, pp. 250-255, 2007.

[13] D. W. Yun, "Accurate 3-D prediction using cutting condition independent coefficients in end milling", *International Journal of Machine Tool & manufacture* , vol. 14, pp. 463-478, 2001.

[14] N. Z. Yussefian, B. Moetakef-Imani and H. El-Mounayri, "The prediction of cutting force for boring process", *International Journal of Machine Tools & Manufacture* , vol. 48 , pp. 1387–1394, 2008.

[15] Min Wan and Wei-HongZhang, "Systematic study on cutting force modeling methods for peripheral milling", *International Journal of Machine Tools & Manufacture*, vol. 49, pp. 424–432, 2009.

[16] S. Haykin, *Neural Networks for statistical modeling*, Van Nostrand Reinhold Editor, 1993.

[17] D. Skapura, *Building neural networks*, Addison-Wesley ACM Press, 1996.

[18] S. V. Kartalopoulos, "Understanding neural networks and fuzzy logic: basic concepts and applications", in *IEEE Press*, 1996.

[19] H. El Kadi, "Modeling the Mechanical Behavior of Fiber-Reinforced Polymeric Composite Materials Using Artificial Neural Networks – A Review", *Composite Structures*, vol. 73, pp. 1-23, 2006.

[20] Z. Zhang and K. Friedrich, "Artificial neural networks applied to polymer composites", *Composite Science and Technology*, vol. 63, pp. 29-44, 2003.

[21] V.Tandon and H. El-Mounayri, "A Novel Artificial Neural Networks Force Model for End Milling", *International Journal of Advanced Manufacturing Technology*, vol. 18, pp. 693-700, 2001.

[22] S. Aykut , M. Gölcüa, S. Semiz and H. S. Ergur, "Modeling of cutting forces as function of cutting parameters for face milling of Satellite 6 using an artificial neural network", *Journal of Materials Processing Technology*, vol. 190, pp. 199–203, 2007.

[23] F. Cus, U. Zuperl and M.Milfelner, "Dynamic neural network approach for tool cutting force modeling of end milling operations", *International Journal of General Systems*, vol. 35, pp. 603-618, 2006.

[24] U. Zuperl, F. Cus, B. Mursec and T. Ploj, "A hybrid analytical-neural network approach to the determination of optimal cutting conditions", *Journal of Materials Processing Technology*, vol. 175, pp. 82–90, 2004.

[25] A. M. Zain, "Prediction of surface roughness in the end milling machining using Artificial Neural Network", *Expert Systems with Applications*, vol. 37, pp. 1755-1768, 2010.

[26] Z. Uros, "Adaptive network based inference system for estimation of flank wear in end milling", *Journal of Materials Processing Technology*, vol. 209, pp. 1504-1511, 2009.

[27] T. Ozel and Y. Karpat, “Predictive modelling of surface roughness and tool wear in hard turning using regression and neural networks”, *International Journal of Machine Tool and Manufacture,* vol. 45, pp. 467–479, 2005.

[28] A. M. A. Al-Ahmari, “Predictive Machinability models for a selected hard material in turning operations”, *Journal of Materials Processing Technology,* vol. 190, pp. 305-311, 2007.

[29] J. P. Davim, V. N. Gaitonde and S. R. Karnik, “Investigations into the effect of cutting conditions on surface roughness in turning of free machining steel by ANN models”, *Journal of Materials Processing Technology,* vol. 205, pp. 16-23, 2008.

[30] J. F. Briceno, H. Mounayri and Mukhopadhyay, “Selecting an artificial neural network for efficient modeling and accurate simulation of the milling process”, *International Journal of Machine Tools & Manufacture*, vol. 42, pp. 663-674, 2002.

[31] Y. Liu and C. Wang, “Neural network based adaptive control and optimisation in the milling process”, *International Journal of Advanced Manufacturing Technology* , vol. 15, pp. 791–795, 1999.

[32] W. Campbell, K. Assaleh and C. Broun, “Speaker recognition with polynomial classifiers”, in *IEEE Transactions on Speech and Audio Processing*, 2002, pp. 205-212.

[33] I. M. Deiab, K. Assaleh and F. Hammad, “On modeling of tool wear using sensor fusion and polynomial classifiers”, *Mechanical Systems and Signal Processing*, vol. 23, pp. 1719–1729, 2009.

[34] MATLAB version 6.5.1. Natick, Massachusetts: The MathWorks Inc., 2003.

APPENDICES

APPENDIX A

Table 17 Cutting Parameters

Set	RPM	DOC	Chipload	Feed
C1	1000	40	2	2
C2	2000	40	2	4
C3	3000	40	2	6
C4	1000	40	4	8
C5	2000	40	4	10
C6	3000	40	4	12
C7	1000	40	6	14
C8	2000	40	6	16
C9	3000	40	6	18
C10	1000	40	8	20
C11	2000	40	8	22
C12	3000	40	8	24
C13	1000	150	2	2
C14	2000	150	2	4
C15	3000	150	2	6
C16	1000	150	4	8
C17	2000	150	4	10
C18	3000	150	4	12
C19	1000	150	6	14
C20	2000	150	6	16
C21	3000	150	6	18
C22	1000	150	8	20
C23	2000	150	8	22
C24	3000	150	8	24
C25	1000	250	2	2
C26	2000	250	2	4
C27	3000	250	2	6
C28	1000	250	4	8
C29	2000	250	4	10
C30	3000	250	4	12
C31	1000	250	6	14
C32	2000	250	6	16
C33	3000	250	6	18
C34	1000	250	8	20
C35	2000	250	8	22
C36	3000	250	8	24

APPENDIX B

Shown below in details is for the first condition C1.

```
%
load c1.txt
GainKisX=200/4.448222;
GainKisY=200/4.448222;
GainKisZ=200/4.448222;
FX=GainKisX*c1(:,1);
FY=GainKisY*c1(:,2);
FZ=GainKisZ*c1(:,3);
SizeFX=size(FX);
SizeFX=SizeFX(1)
%
Range=[1:SizeFX];
hold off;
plot(FX(Range));hold on;
plot(FY(Range),'r');
plot(FZ(Range),'c');
legend('FX','FY','FZ');grid
zoom on,pause
```

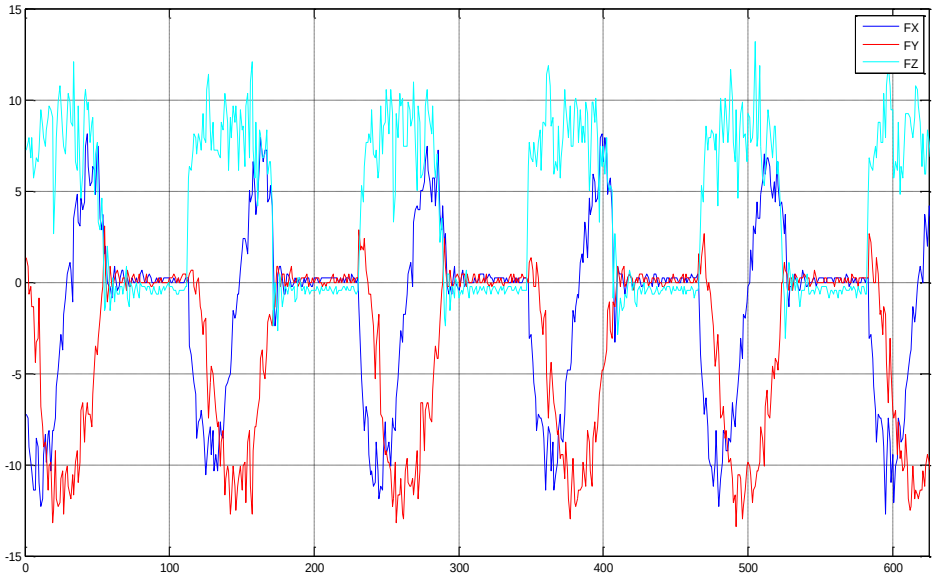

Figure 31 Transformation of signal into forces

The resulting signal shown in **Figure 31** is now measured in forces; this signal however, needs to be filtered. This is done by subtracting the drift. It is known that there is a known range where there is no contact between the insert and the work piece, during that time, the signal should measure zero. For the case of C1, there are 17 engagements of the insert with the work piece and the tool has 4 inserts,

therefore the signal presented is that of 4 revolutions and the signal is periodic. As can be seen from the figure above, the range where the insert and the work piece are not in contact are in the ranges of 70-100, 180-210 and so on. The range chosen should be the first and the last insert engagement with the work piece, for C1 condition is 70-100, and 1950-1980. The average of the signal for the 2 ranges is calculated and the difference between the 2 ranges is obtained and multiplied by the signal and divided by the sample size.

```
%
RangeOne=[70:100];
RangeTwo=[1950:1980];
%
SizeFX=size(FX);
SizeFX=SizeFX(1)
MeanFXOne=mean(FX(RangeOne));
MeanFXTwo=mean(FX(RangeTwo));
FXOffset=(MeanFXTwo-MeanFXOne)*[1:SizeFX]/SizeFX+MeanFXOne;
hold off
plot(FX);hold on;plot(FXOffset,'r')
FXNew=FX-FXOffset';
hold off
plot(FX);hold on;plot(FXNew,'r')
zoom on,pause
%
SizeFY=size(FY);SizeFY=SizeFY(1)
MeanFYOne=mean(FY(RangeOne)); MeanFYTwo=mean(FY(RangeTwo));
FYOffset=(MeanFYTwo-MeanFYOne)*[1:SizeFY]/SizeFY+MeanFYOne;
hold off
plot(FY);hold on;plot(FYOffset,'r');
FYNew=FY-FYOffset';
hold off
plot(FY);hold on;plot(FYNew,'r')
%
SizeFZ=size(FZ);SizeFZ=SizeFZ(1)
MeanFZOne=mean(FZ(RangeOne)); MeanFZTwo=mean(FZ(RangeTwo));
FZOffset=(MeanFZTwo-MeanFZOne)*[1:SizeFZ]/SizeFZ+MeanFZOne;
hold off
plot(FZ);hold on;plot(FZOffset,'r')
FZNew=FZ-FZOffset';
hold off
plot(FZ);hold on;plot(FZNew,'r')
%
FX=FXNew;FY=FYNew;FZ=FZNew;
clear FXNew FYNew FZNew
%
bb=[FX FY FZ]
save mod_c1.txt bb -ascii
%
```

The signal after filtering is as shown in **Figure** 32

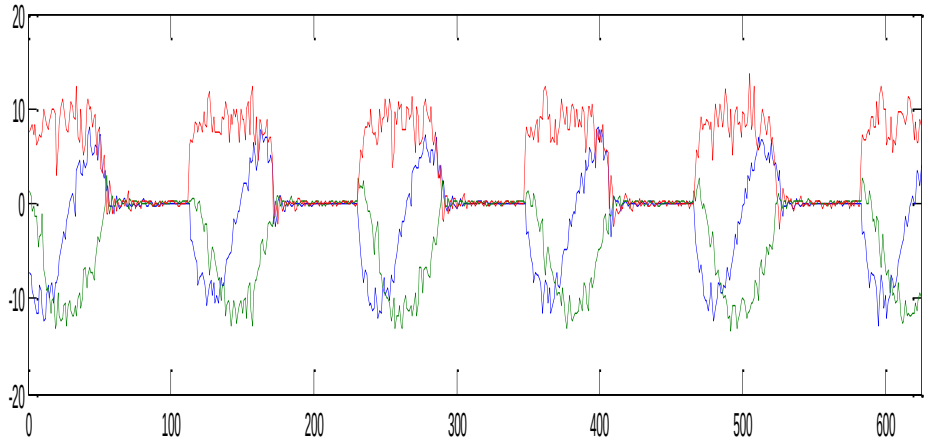

Figure 32 Force Signal after filtering

The RPM at condition 1 is 1000 rev/min and the sample frequency is set at 2005 for all conditions. The tool holder houses 4 inserts. Therefore, the number of data points in one revolution is calculated as follows:

```
NSpin=1000
ForceSampleFrequency=2005;
hold off;plot(FX)
NPointPeriod=4*60*ForceSampleFrequency/NSpin
%
```

For C1, the number of data points per insert is 120, where the first 60 points are during engagement of the insert with the work piece and the other 60 points is during no cutting, 480 data points per revolution, the start data point of the insert engagement with the work piece should be defined, for C1 the first engagement of the tool with the work piece is shown in Figure 33, where it is zoomed into and it shows that it is at data point 112.

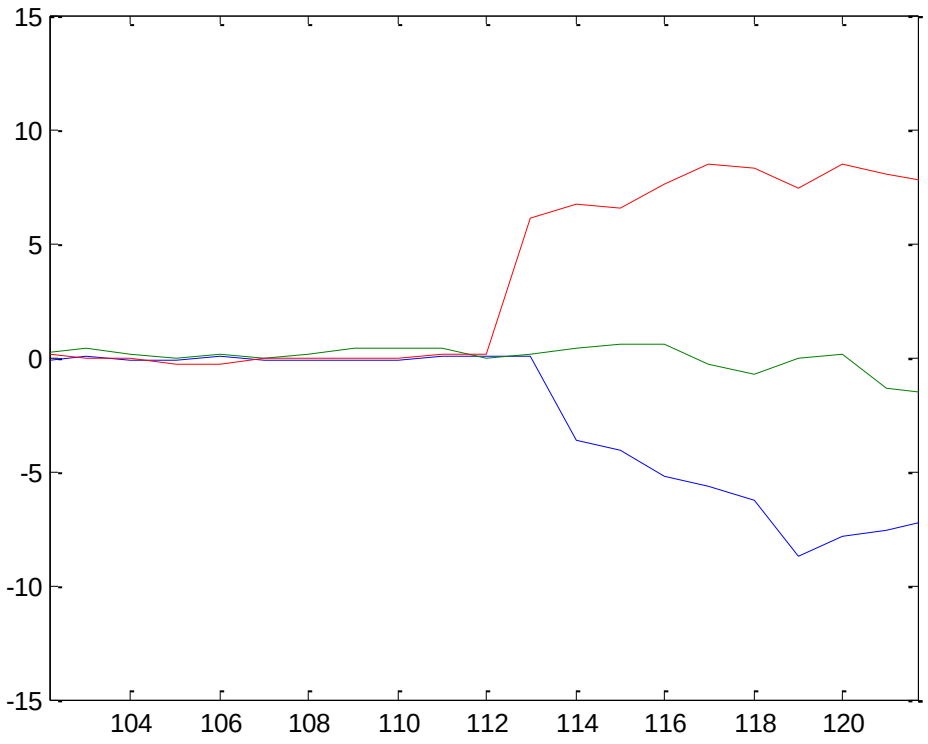

Figure 33 Zoom in on first engagement of tool with workpiece

5 ranges are chosen so that an average force is obtained, that is done by taking 5 ranges starting from 112 and adding 120 for every range.

```
% Mean force

StartingPoint=112;
Range1=[StartingPoint:StartingPoint+NPointPeriod-1];
Range2=Range1+NPointPeriod;
Range3=Range1+2*NPointPeriod;
Range4=Range1+3*NPointPeriod;
Range5=Range1+4*NPointPeriod;
Range6=Range1+5*NPointPeriod;
hold off
plot(FX(Range1));hold
on;plot(FX(Range2),'r');plot(FX(Range3),'c');plot(FX(Range4),'m');
plot(FX(Range5),'g');
grid on
legend('1','2','3','4','5')

% FX Mean
MeanFXOnRev=(FX(Range1)+FX(Range2)+FX(Range3)+FX(Range4)+FX(Range5
))/5;
hold off
subplot(3,1,1),plot(FX(Range1));hold
on;plot(FX(Range2),'r');plot(FX(Range3),'c');plot(FX(Range4),'m');
plot(FX(Range5),'g');
plot(MeanFXOnRev,'LineWidth',2.);hold off;title('FX')
legend('1','2','3','4','5','Average')
```

```
grid on
% FY Mean
MeanFYOnRev=(FY(Range1)+FY(Range2)+FY(Range3)+FY(Range4)+FY(Range5
))/5;
hold off
subplot(3,1,2),plot(FY(Range1));hold
on;plot(FY(Range2),'r');plot(FY(Range3),'c');plot(FY(Range4),'m');
plot(FY(Range5),'g');
plot(MeanFYOnRev,'LineWidth',2.);;title('FY');
legend('1','2','3','4','5','Average')
grid on
hold off
% FZ Mean
MeanFZOnRev=(FZ(Range1)+FZ(Range2)+FZ(Range3)+FZ(Range4)+FZ(Range5
))/5;
hold off
subplot(3,1,3),plot(FZ(Range1));hold
on;plot(FZ(Range2),'r');plot(FZ(Range3),'c');plot(FZ(Range4),'m');
plot(FZ(Range5),'g');
plot(MeanFZOnRev,'LineWidth',2.);title('FZ');
legend('1','2','3','4','5','Average')
grid on
hold off
```

The output of this file is shown in **Figure** 34 below:

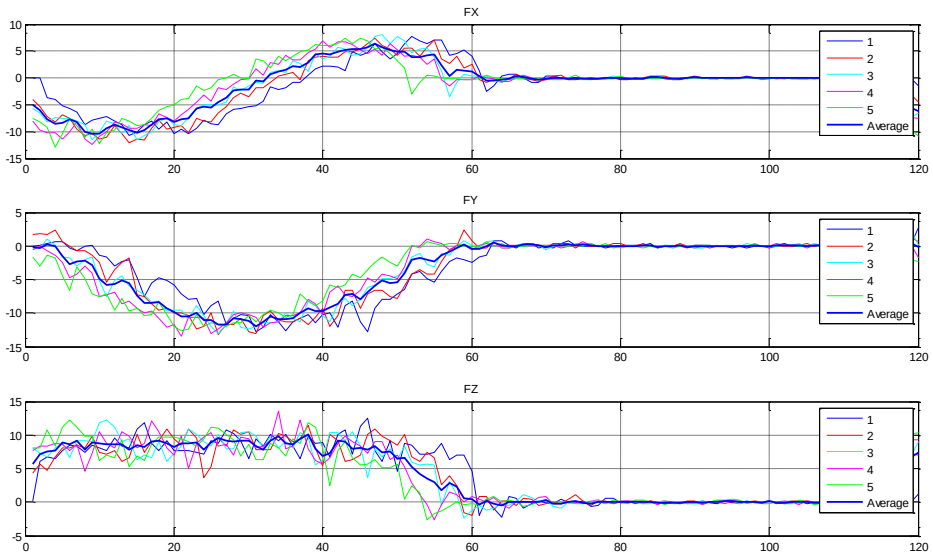

Figure 34 Average of 5 signals of F_x, F_y and F_z

```
cc1=[MeanFXOnRev MeanFYOnRev MeanFZOnRev]
```

As mentioned earlier, the first 60 data points are those of the tool engagement, therefore the mean on the first 60 points will be taken into consideration

```
Range=[1:60];
MeanFXOnRevTronc=MeanFXOnRev(Range);
MeanFYOnRevTronc=MeanFYOnRev(Range);
MeanFZOnRevTronc=MeanFZOnRev(Range);
a=[MeanFXOnRevTronc MeanFYOnRevTronc MeanFZOnRevTronc]
```

```
b=mean(a)
```

The mean force on the tool is now obtained in matrix b, as the following, in newtons:

b =

-1.2802 -3.1901 4.0912

```
AC=4
T=[1 tan(30)*cos(40)/cos(50);tan(30)
cos(40)/cos(50);tan(50)/cos(30) sin(40)/cos(50)*cos(30)]
R=[cos(30) -1*sin(30) 0;sin(30) cos(30) 0;0 0 1]
T1=pinv(T)
b1=b'
cuttingforce=T1*inv(R)*b1
ft=cuttingforce(1,:)
fr=cuttingforce(2,:)
Kt=ft/AC
Kr=fr/AC
```

APPENDIX C

Table 18 Data Matrix

Set	RPM	DOC	Chipload	Feed	Training	Validation
C1	1000	40	2	2	x	
C2	2000	40	2	4		x
C3	3000	40	2	6	x	
C4	1000	40	4	8		x
C5	2000	40	4	10	x	
C6	3000	40	4	12	x	
C7	1000	40	6	14	x	
C8	2000	40	6	16		x
C9	3000	40	6	18	x	
C10	1000	40	8	20		x
C11	2000	40	8	22	x	
C12	3000	40	8	24	x	
C13	1000	150	2	2	x	
C14	2000	150	2	4		x
C15	3000	150	2	6	x	
C16	1000	150	4	8	x	
C17	2000	150	4	10	x	
C18	3000	150	4	12		x
C19	1000	150	6	14		x
C20	2000	150	6	16	x	
C21	3000	150	6	18	x	
C22	1000	150	8	20	x	
C23	2000	150	8	22		x
C24	3000	150	8	24	x	
C25	1000	250	2	2	x	
C26	2000	250	2	4		x
C27	3000	250	2	6	x	
C28	1000	250	4	8		x
C29	2000	250	4	10	x	
C30	3000	250	4	12	x	
C31	1000	250	6	14	x	
C32	2000	250	6	16	x	
C33	3000	250	6	18		x
C34	1000	250	8	20	x	
C35	2000	250	8	22		x
C36	3000	250	8	24	x	

APPENDIX D

Table 19 Networks Tested

Network number	Type	No. of layers	Layer 1		Layer 2		Training function
			Transfer Function	No. of neurons	Transfer Function	No. of neurons	
network 1	Feed Forward back propagation	3	logsig	10	purelin	5	Trainlm
network 2	Feed Forward back propagation	2	tansig	20			Trainlm
network3	Feed Forward back propagation	2	tansig	5			Trainlm
network 4	Feed Forward back propagation	2	tansig	10			Trainlm
network 5	Feed Forward back propagation	2	tansig	40			Trainlm
network 6	Feed Forward back propagation	2	purelin	10			Trainlm
network 7	Feed Forward back propagation	2	tansig	20			Trainlm
network 8	Radial basis						
network 9	Cascade forward back propagation	2		20			Trainlm
network 10	Elman back propagation						
network 11	Layer Recurrent						
network 12	Feed Forward back propagation	2	tansig	10			Trainrp
network 13	Feed Forward back propagation	2	tansig	10			Trainscg
network14	Feed Forward back propagation	3	logsig	10	purelin	5	Trainrp
network15	Feed Forward back propagation	3	logsig	10	purelin	5	Trainscg
network 16	Feed Forward back propagation	2	tansig	50			Trainlm
network 17	Feed Forward back propagation	2	logsig	50			Trainlm
network 18	Feed Forward back propagation	3	tansig	10	tansig	10	Trainlm
network 19	Feed Forward back propagation	3	tansig	10	tansig	20	Trainlm
network 20	Feed Forward back propagation	3	tansig	10	tansig	5	Trainlm
network 21	Feed Forward back propagation	3	tansig	10	tansig	40	Trainlm
network 22	Feed Forward back propagation	3	tansig	20	tansig	10	Trainlm
network 23	Feed Forward back propagation	3	tansig	20	tansig	40	Trainlm
network 24	Feed Forward back propagation	2	tansig	30			Trainlm
network 25	Feed Forward back propagation	2	logsig	10			Trainlm
network 26	Feed Forward back propagation	2	Tansig	4			Trainlm
network 27	Feed Forward back propagation	3	Tansig	8		16	Trainlm
network 28	Radial Basis			25			Trainlm
network 29	Feed Forward back propagation	2	Tansig	8			Trainlm

	Feed Forward back propagation	2	Tansig	8			Trainlm
	Feed Forward back propagation	3	Tansig	8		4	Trainlm
	Feed Forward back propagation	2	Logsig	8			Trainlm
	Feed Forward back propagation	3	Tansig	8		16	Trainlm
	Feed Forward back propagation	2	Tansig	8			Trainlm
	Radial Basis			24			Trainlm
	Feed Forward back propagation	2	Tansig	10			Trainlm
	Feed Forward back propagation	2	Logsig	8			Trainlm
	Feed Forward back propagation	3	Logsig	8	Purlin	4	Trainlm
	Feed Forward back propagation	2	Tansig	10			Trainlm
	Feed Forward back propagation	2	Logsig	8			Trainlm
	Feed Forward back propagation	3	Logsig	8	Purlin	4	Trainlm

APPENDIX E

Run of code in appendix gave the following results.

Table 20 K_t and K_r Calculated Values

Set	RPM	DOC	Chipload	Feed	Training	Validation	K_t	K_r
C1	1000	40	2	2	X		1.2778	1.4923
C2	2000	40	2	4		X	1.7217	1.8324
C3	3000	40	2	6	X		2.2986	2.3972
C4	1000	40	4	8		X	2.2829	2.5073
C5	2000	40	4	10	X		1.5446	1.7089
C6	3000	40	4	12	X		1.4554	1.6798
C7	1000	40	6	14	X		1.5091	1.8648
C8	2000	40	6	16		X	1.2546	1.4741
C9	3000	40	6	18	X		1.2778	1.4923
C10	1000	40	8	20		X	1.0932	1.5153
C11	2000	40	8	22	X		1.0807	1.3594
C12	3000	40	8	24	X		0.8835	1.3051
C13	1000	150	2	2	X		1.3823	1.6817
C14	2000	150	2	4		X	2.2898	1.4166
C15	3000	150	2	6	X		1.7082	2.5506
C16	1000	150	4	8	X		1.1727	1.7917
C17	2000	150	4	10	X		1.6816	1.9980
C18	3000	150	4	12		X	1.2436	2.014
C19	1000	150	6	14		X	1.3288	2.0374
C20	2000	150	6	16	X		0.6411	2.2897
C21	3000	150	6	18	X		0.3136	2.3393
C22	1000	150	8	20	X		0.9305	2.0738
C23	2000	150	8	22		X	1.8299	3.5690
C24	3000	150	8	24	X		1.9173	3.8680
C25	1000	250	2	2	X		2.3321	6.2315
C26	2000	250	2	4		X	3.3520	7.0991
C27	3000	250	2	6	X		3.4895	7.3303
C28	1000	250	4	8		X	2.6693	6.1053
C29	2000	250	4	10	X		2.6629	6.6887
C30	3000	250	4	12	X		1.6269	5.6812
C31	1000	250	6	14	X		2.3671	5.6055
C32	2000	250	6	16	X		2.4323	5.7096
C33	3000	250	6	18		X	2.7234	5.8658
C34	1000	250	8	20	X		2.2052	5.3148
C35	2000	250	8	22		X	1.9680	5.2140
C36	3000	250	8	24	X		1.7372	4.8678

APPENDIX F – DETAILS ON ANNS

FEED FORWARD BACK PROPAGATION NEURAL NETWORK

Feedforward ANN in general consist of a layer of input neurons, a layer of output neurons and one or more layers of hidden neurons [17]. Neurons in each layer are interconnected fully to previous and next layer neurons with each interconnection have an associated connection strength or weight. The activation function used in the hidden and output layers' neurons is non-linear, where as for the input layer no activation function is used since no computation is involved in that layer. Information flows from one layer to the other layer in a feedforward manner. Various functions are used to model the neuron activity such as sigmoid, tanh or radial (Gaussian) functions. Figure 35 illustrates a feed forward neural network.

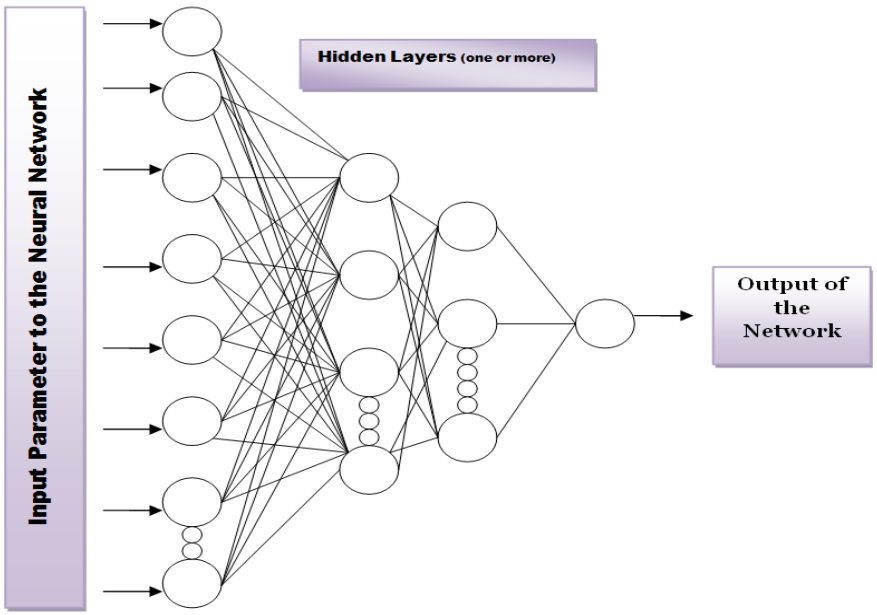

Figure 35 Proposed ANN Structure

The input to a node i in the k^{th} layer is given by [35[1]]:

$$net_{i,k} = \left[\sum_{j} w_{i,j,k} out_{j,k-1} \right] + \theta_{i,k} \qquad (24)$$

where,

$w_{i,j,k}$ represents the weight connection strengths for node *j* in the *(k-1)*th layer to node *i* in the k^{th} layer, $out_{i,k}$ is the output of node *i* in the k^{th} layer and $\theta_{i,k}$ is the threshold associated with node *i* in the k^{th} layer.

Collectively the hidden layers perform the application desired objective whether it is classification, modeling, pattern recognition …etc.

CASCADE FORWARD BACK PROPAGATION NEURAL NETWORK

In Matlab, the function *newcf* creates cascade-forward networks (CFFN). These are similar to feed-forward networks, but include a weight connection from the input to each layer, and from each layer to the successive layers. For example, a three-layer network has connections from layer 1 to layers 2, layer 2 to layer 3, and layer 1 to layer 3. The three-layer network also has connections from the input to all three layers. The additional connections might improve the speed at which the network learns the desired relationship.

ELMAN NEURAL NETWORK

The Elman network (ELM) is commonly a two-layer network with feedback from the first-layer output to the first-layer input. This recurrent connection allows the Elman network to both detect and generate time-varying patterns. A two-layer Elman network is shown in Figure 36.

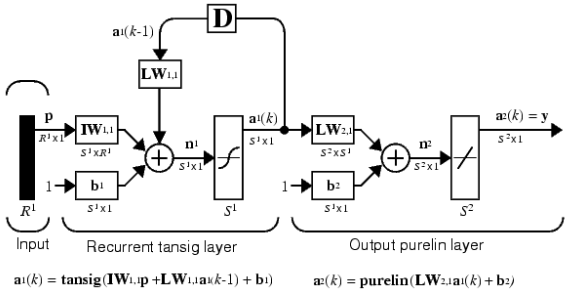

Figure 36 Elman Recurrent Network [34]

The Elman network has tansig neurons in its hidden (recurrent) layer, and *purelin* neurons in its output layer. This combination is special in that two-layer networks with these transfer functions can approximate any function (with a finite number of discontinuities) with arbitrary accuracy. The only requirement is that the hidden layer must have enough neurons. More hidden neurons are needed as the function being fitted increases in complexity. Note that the Elman network differs from

conventional two-layer networks in that the first layer has a recurrent connection. The delay in this connection stores values from the previous time step, which can be used in the current time step. Thus, even if two Elman networks, with the same weights and biases, are given identical inputs at a given time step, their outputs can be different because of different feedback states. Because the network can store information for future reference, it is able to learn temporal patterns as well as spatial patterns. The Elman network can be trained to respond to, and to generate, both kinds of patterns [34].

LAYER RECURRENT NEURAL NETWORK

The next dynamic network to be introduced is the Layer-Recurrent Network (LRN). An earlier simplified version of this network was introduced by Elman. In the LRN, there is a feedback loop, with a single delay, around each layer of the network except for the last layer. The original Elman network had only two layers, and used a tansig transfer function for the hidden layer and a *purelin* transfer function for the output layer. The original Elman network was trained using an approximation to the backpropagation algorithm. The *newlrn* command generalizes the Elman network to have an arbitrary number of layers and to have arbitrary transfer functions in each layer. The toolbox trains the LRN using exact versions of the gradient-based algorithms discussed in Backpropagation. Figure 37 illustrates a two-layer LRN [34].

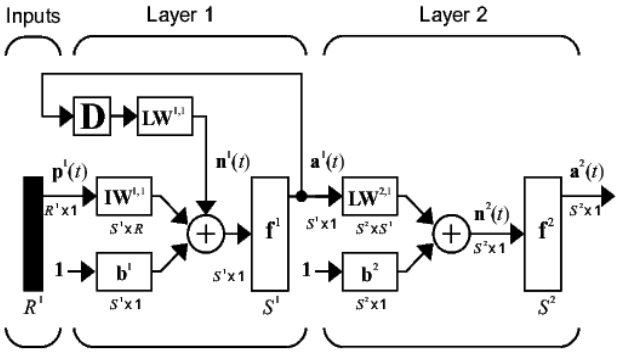

Figure 37 Layer Recurrent Neural Network [34]

The back-propagation training algorithm is commonly used to iteratively minimize the following cost function with respect to the interconnection weights and neurons thresholds:

$$E = \frac{1}{2}\sum_{1}^{P}\sum_{i=1}^{N}(d_i - O_i)^2 \tag{25}$$

where P is the number of experimental data pairs used in training the network and N is the number of output parameters expected from the ANN. d_i and O_i could be the experimental number of cycles to failure and the current life prediction of the ANN for each loading condition i respectively. Iteratively, the interconnection weights between the j^{th} node and the i^{th} node are updated as:

$$w_{ji}(t+1) = \alpha\, w_{ji}(t) + \eta\, x_i\, f'\left(net_j^k\right)\sum_{l=1}^{N}\left(d_l - O_l\right) f'\left(net_l^0\right) w_{lj} \tag{26}$$

where α is a momentum constant, η the learning rate, x_i the input pattern at the iterative sample t, net_N^0 the input to node N at the output layer and net_j^k is the input to a node j in the k^{th} layer and the function f' is the derivative of the neuron activation function. The learning rate determines what amount of the calculated error sensitivity to weight change will be used for the weight correction. It affects the convergence speed and the stability of weights during learning. The "best" value of the learning rate depends on the characteristics of the error surface. For rapidly changing surfaces, a smaller rate is desirable while for smooth surfaces, a larger value of the learning rate will speed up convergence. The momentum constant (usually between 0.1 and 1) smoothes weight updating and prevents oscillations in the system and helps the system escape local minima in the training process by making the system less sensitive to local changes. Much as the learning rate, the momentum constant "best" value is also peculiar to specific error surface contours. The training process is terminated either when the Mean-Square-Error (MSE), Root-Mean-Square-Error (RMSE), or Normalized-Mean-Square-Error (NMSE), between the actual experimental results and the ANN predictions obtained for all elements in the training set has reached a pre-specified threshold or after the completion of a pre-specified number of learning epochs.

In addition to the typical back-propagation algorithm, the following training functions are also considered in this study:

Resilient Back-propagation (RP) –

Multilayer networks typically use sigmoid transfer functions in the hidden layers. These functions are often called "squashing" functions, because they compress an infinite input range into a finite output range. Sigmoid functions are characterized by the fact that their slopes must approach zero as the input gets large. This causes a problem when steepest descent is used to train a multilayer network with sigmoid functions because the gradient can have a very small magnitude and, therefore, cause small changes in the weights and biases, even though the weights and biases are far from their optimal values. The purpose of the resilient back-propagation training algorithm is to eliminate these harmful effects of the magnitudes of the partial derivatives [34].

Gradient Descent (GD) –

In the steepest descent training function, the weights and biases are updated in the direction of the negative gradient of the performance function. The learning rate is multiplied by the negative of the gradient to determine the changes to the weights and biases. The larger the learning rate, the bigger the step. If the learning rate is made too large, the algorithm becomes unstable. If the learning rate is set too small, the algorithm takes a long time to converge. The training stops if the number of iterations exceeds the predetermined number of epochs, the performance function drops below a specific goal, the magnitude of the gradient is less than a stipulated value, or the training time surpasses a preset time [34].

Gradient Descent with Momentum (GDM) –

Gradient descent with momentum allows a network to respond not only to the local gradient, but also to recent trends in the error surface. Acting like a low pass filter, momentum allows the network to ignore small features in the error surface. Without momentum a network can get stuck in a shallow local minimum. With momentum a network can slide through such a minimum [34].

Variable Learning Rate (GDA) –

With standard steepest descent, the learning rate is held constant throughout training. The performance of the algorithm is very sensitive to the proper setting of the learning rate. If the learning rate is set too high, the algorithm can oscillate

and become unstable. If the learning rate is too small, the algorithm takes too long to converge. It is not practical to determine the optimal setting for the learning rate before training, and, in fact, the optimal learning rate changes during the training process, as the algorithm moves across the performance surface. The performance of the steepest descent algorithm can be improved by allowing the learning rate to change during the training process. An adaptive learning rate attempts to keep the learning step size as large as possible while keeping learning stable. The learning rate is made responsive to the complexity of the local error surface [34].

Variable Learning Rate with Momentum (GDX) –

This function combines adaptive learning rate with momentum training. It is invoked in the same way as GDA except that it has the momentum coefficient as an additional training parameter [34].

VITA

Amal Khattab was born on May 1984, in Abu Dhabi, UAE. She was educated in Al Worood School, a local school, and graduated with a Bachelor degree of science in Mechanical Engineering from the American University of Sharjah in 2006 with Cum laude.

She enrolled in the American University of Sharjah's Graduate Program of Mechatronics and worked as Graduate Teaching Assistant, for a year. She has been working in Tawazun Precision Industries from 2007 till current.