JJMIE

Volume 1, Number 1, Sep. 2007
ISSN 1995-6665
Pages 43 - 55

Jordan Journal of Mechanical and Industrial Engineering

A Neural Network Based Real Time Controller for Turning Process

Bahaa Ibraheem Kazem [a,*], Nihad F. H. Zangana [b]

[a] Mechatronics Engineering Dept., [b] Mechanical Engineering Dept., University of Baghdad, Baghdad-Iraq

Abstract

In this paper, the design and implementation of an effective neural network model for turning process identification as well as a neural network controller to track a desired vibration level of the turning machine is as an example of using the neural network for manufacturing process control. Multi – Layer Perceptron (MLP) neural network architecture with Levenberg Marquardt (LM) algorithm has been utilized to train the turning process identifier. Two different strategies have been used for training turning process identifier, and for training the controller model, where there is no mathematical model till now could relate the vibration level to the input turning process parameters "feed, speed, and depth of cut". The vibration signal obtained by the experimental work has been used to train a neural network for identification and control of the turning process. The developed Neuro – controller has been checked by applying different reference vibration signals where it is found that the controller has good ability to track the reference within maximum settling time that does not exceed (4 sec for 95% of the signal); maximum overshot not exceed (30%) of the reference signal used for checking.

Keywords: Real Time Control; Neural Network; Turining.

1. Introduction

The present goal of manufacturing researches focuses on developing flexible, self –adjusting and unattended intelligent machine systems. The limited presence of operators at manned machine tools leaves the supervision, monitoring and control tasks to computer controllers. Although an unattended machining process needs almost no attendance of an operator, tasks such as sensing the effect of process variables and adjusting the conditions accordingly have to be done by appropriate sensors and associated monitors. One solution is to provide on-line adjustment of operating parameters based on sensor information. Systems which posses such capabilities are referred to as adaptive control AC systems. Actually most machining adaptive controllers are categorized in so-called adaptive control for constraints ACC systems, where the operating parameters are adjusted so as to maximize productivity while respecting process constraints like cutting force or power limits. In practice, the most important draw back of ACC is their lack of feed back on part quality where there is no measurement device that could measure part quality "surface finish" in an on line real time manner. In contrast adaptive control with optimization ACO systems adjust the operating parameters so that predefined parameters of performance index are optimized [1]. Most ACO systems assumed a detailed process model is available and complete with known analytical or empirical constant. A great draw back rise here which is the need to collect very specialized experimental and analytical data generally required for model simulation before its implementation in the feed back control scheme. To alleviate some of these problems and provide the model and the controller with more intelligence, better fit to nonlinear behaviour and capacity of adaptation over time, neural networks appear as one of the most interesting techniques.

In recent years, there was an increase interest shown in the utilization of neural networks for various research fields such as robotics, optimization, linear and non-linear programming, pattern recognition, and computer vision. This was due to the advances in neural network algorithms and also the availability of fast parallel architectures that are used to control dynamical systems such as machining systems. The aim is of using multilayered neural network composed of feed back and feed forward controllers, and several learning architectures to train the neural controller in order to provide appropriate inputs to the plant, so the desired response is obtained. In comparison with traditional adaptive controller, their results indicate that neural network approaches well in noise elimination, work for linear and non-linear systems and can be implemented

* Corresponding author. e-mail: drbahaa@uob.edu.iq

very efficiently for large-scale systems [2]. The neural network also used in wide variety of data processing applications, where real-time data analysis and information extraction is required. A great advantage of neural network approach is that most of intense computation takes place during the training process. Once the neural network is trained for particular task, operation is relatively fast and unknown samples can be rapidly identified in the field [3]. Other applications such as modeling, industrial inspection and quality control have been spread in manufacturing field [2].

Presentation and analysis for computerized numerical control CNC for manufacturing system have been introduced by Koren [4]. Two types of CNC systems referred to as Reference-Pulse and Sampled-Data are discussed. In the first system, reference pulses were generated by the computer and supplied to an external digital control loop. With the Sampled – Data technique, the computer served as a comparator of the control loop and transmitted the position error at fixed time intervals. Fig.1. represent block diagram of a Sampled-Data CNC system. Both types had been analyzed analytically and verified experimentally and the results were satisfied.

Athani and Vinod [5] proposed several changes on special type of lathe machine used for watch making. Two stepper motors used to drive the carriage and the cross slide, and low cost PC type (Sinclair ZX spectrum) used as a control platform.

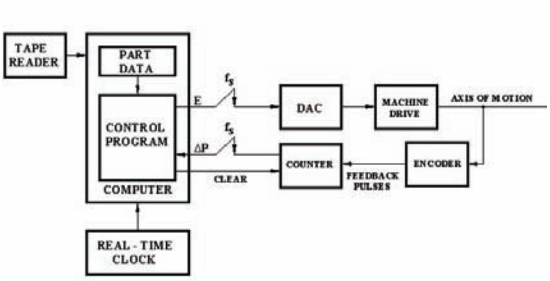


Figure 1: Block diagram of a sampled-data CNC system [4]

Achi [6] have described the utilization of microcomputer to control stepping motor –actuated hydraulic servos deriving a two axis milling machine; also a program for interpolation purpose developed and saved in an assembly language form in the memory.

Altintas and Peng [7] have made a suggestion and implementation of a program for electronically controlling of the speed and position associated to feed operation in a research milling machine. The system main consistence were DC-servomotor (actuator), encoder (for position feed back signal), and tachogenerator (for velocity feed back signal). An IBM PC and interface card controller type (DMC-230 motion controller) used. The response analysis for system analytically and experimentally were found close as follows, a (60) Hz was the frequency operation band for velocity loop, (20) ms was the settling time, and a steady state error of (0.0137fc) “fc feed velocity command”. Fig.2. represent the architecture of the suggested control system.

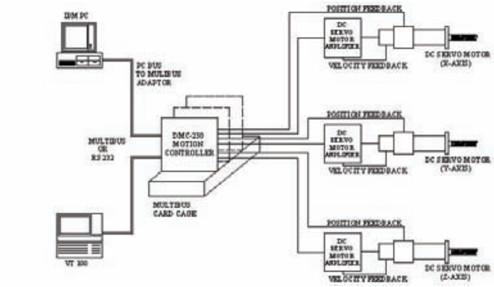


Figure 2: Architecture of speed and position control system for milling process [7]

George et al. [8] evolved a synchronizing control algorithm. This algorithm was developed to minimize the tracking error and the contouring error with stronger emphasis on contouring error. An Intel (486) based AT compatible computer had been used (applying this algorithm) to control a (Matsuura MC510V) high speed, (3-axis) vertical machining centre. A schematic diagram of the control system is shown in Fig.3.

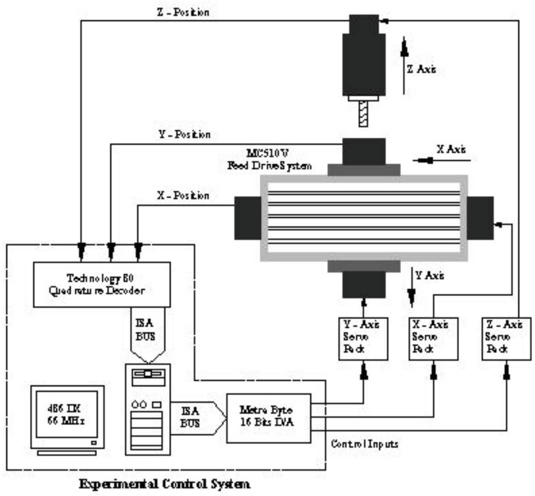


Figure 3: Schematic diagram of the control system for 3-axis vertical machining centre [8]

Jeffery et al. [9] proposed a cutting force-monitoring approach. The approach did not utilize force dynamometers but rather estimates the cutting forces based on the spindle motor current and speed as well as a model that relates these measurements to the cutting force. This method was demonstrated on a CNC lathe machine; the empirical tests showed that the static accuracy was less than (5%) for the proposed system. For large cutting forces the accuracy was. reasonable (20dB S/N ratio), while with lower cutting forces the accuracy decreased.

Khanchustambham and Zhang [10] developed an intelligent on-line monitoring system through a neural network approach. Where the monitoring system detects the cutting force produced during the machining, estimates the tool wear status and finish quality from the dynamic variation of detected cutting force signal and makes a decision for taking corrective action when it is needed .The monitor have been built on feed forward back propagation-algorithm. After the training of the network, it had been applied for cutting force and surface finish monitoring during the turning of advanced ceramic materials. Fig.4. represents a monitoring system.

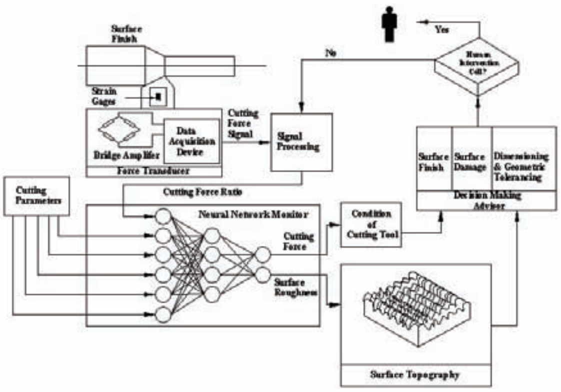
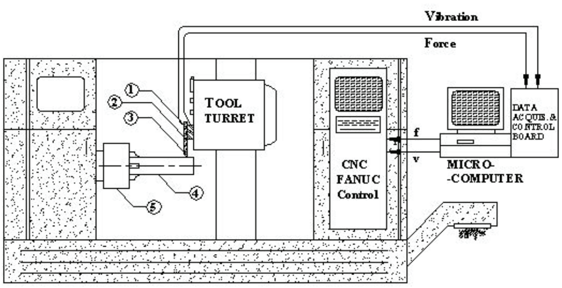


Figure 4: Nural net work Monitoring system for turning process[10]

Larsen [11] have showed that due to significant effects of friction and backlash with turning at low feed rate in nanometric positioning accuracy of machine tool axis especially with significant turning operation such as diamond turning of glass, ceramic, germanium and zinc sulfide for optical usage. A learning motion control algorithm based on the cereblller model articulation controller neural network developed for servo control; the learning controller was implemented using ‘C’ language on a digital signal processing based apon architecture controller or the single point diamond turning machine.

Azouzi and Guillot [12] evolved an inverse process neurocontroller implemented in multilayer feed forward neural network. On-line adjustments of feed rate and cutting speed ware carried out based on a cost/quality performance index “where the chosen performance index was reaching best quality of product within minimal cost” which were estimated from force and vibration sensor measurements. The simulation and experimental investigations demonstrated the effectiveness of neural network for controlling and optimizing of manufacturing operations. Applied to a single point turning of a typical finishing cutting process, the final dimensions and surface finishes were found to be better by (40) and (80) percent respectively, while productivity was increased by (40) percent over the conditions proposed in machining data handbooks. Fig.5. demonstrates the overall experimental installation.



No.	Describition	No.	Describition
1	Dynamometer	4	Workpiece
2	Accelerometer	5	Spindle
3	Cutting Tool		

Figure 5: Overall experimental installation for on-line adjustments of feed rate and cutting speed [12]

Özel and Nadgir [13] developed two neural network models, one was the back-propagation training neural network and the other was the back-propagation prediction neural network. A trained set of back-propagation neural network algorithms used to predict flank wear of cutting tool with chamfered and honed edge preparation during the orthogonal cutting of hardened steel work pieces. The experiment showed that the neural network could estimate the flank wear progress very fast and accurately once the forces are known. The percentage error was found to be (0.59% - 15.09%) between the measured and the predicted values of flank wear.

Jing et al. [14] proposed a novel XY positioning table synchronously driven by one- side dual linear motors. The redundant drive system commonly suffers from synchronous drive precision problem at high speed and acceleration rates. In this paper, the dynamic model of dual linear motors redundant drive system along X-axis direction is given, and synchronous drive precision can be assured by using a synchronous control scheme. This scheme has two model reference adaptive controllers and a synchronous error compensator based on neural networks. Simulation results are provided to manifest the control system has better static and dynamic performance and higher synchronous drive precision at a morn than 10g ($g = 0.81\ m/s^2$) high acceleration profile motion.

In this work, an approach for using the neural network for identification and control of the vibration signal “acceleration" was utilized.

2. Vibration in Manufacturing Process

Vibration and chatter of a cutting tool are complex phenomena, which, if uncontrolled can lead to premature tool failure, bad surface finish, etc. This is particularly important with brittle tool materials such as ceramics, some carbides and diamond. In addition, vibration affects the mechanical surface and its integrity. If excessive, vibrations may even damage machine tools. Furthermore, the noise generated may be objectionable, particularly if it is at a high frequency. Basically, there are two types of vibration in machining [1].

1. Forced vibration: this type of vibration generally caused by some periodic force present in the machine such as that comes from gear drive, imbalance of the machine tool components, etc. in machining process such as milling or turning a shaft with a key way or splined shaft, forced vibrations are also caused by the periodic entry and exit of the cutting tool. The essential efforts here is to minimize the vibration amplitude, since it cause bad surface finishing of the work piece, and changing its frequency far away from the natural frequency of the system to prevent the probability of resonance occurrence. Although changing the cutting process parameters generally doesn’t appear to have much influence of forced vibration, changing the cutting speed may sometimes help [15]. Changing the cutting forces especially the thrust force also can help [1].
2. Self – excited vibration: these vibrations, called “chatter”, happen due to the interaction of the dynamics of the chip removal process and the structural dynamics of the machine tool. The excited vibrations are usually very high in amplitude and may cause damage to the

machine tool. Chatter typically begins with a disturbance in the cutting zone, such as lack of homogeneity in the work piece material or its surface condition and geometry or a change in a frictional condition at the tool-chip interface. The most important type of the self – excited vibration called “regenerative chatter”, this results from the tool cutting a surface that has roughness or disturbances left from a previous cut. Because of the resulting fluctuations in the cutting forces the tool is subjected to vibration and the process repeated continuously, hence the term regenerative. Changing the operating parameters, which generally include feed rate, cutting speed and depth of cut, could control the chatter .

The need of making measurement of vibration has arisen mainly because of the growth of environmental testing. Specification, many a time requires that the equipment should withstand stated levels of vibrations. This could be done quantitatively only through vibration measurements. Vibration measurements are frequently carried out on rotating and reciprocating machinery for analysis, design and trouble shooting purposes. Much knowledge has been gained in the recent years and computer solutions of various vibration problems have been developed [16]. However, many a time it becomes essential to make actual measurements of vibration characteristics by test during development, either on the machine itself or on its prototype because of the fact that it is difficult to build a perfect mathematical model with all its interrelationship and complexity. The most familiar instrument used for vibration measurements is the accelerometer. This instrument is commercially available in a wide verity of types and ranges to meet corresponding diverse application requirements.

The basis for this popularity lies in the following features [17]:

1. Frequency response is from zero to some high limiting value. Steady accelerations can be measured (except in piezoelectric type).
2. Displacement and velocity can be easily obtained by electrical integration, which is much preferred to differentiation.
3. Measurement of transient (shock) motion is more readily achieved than with displacement or velocity pickups.
4. Destructive forces in machinery are related more closely to acceleration than velocity or displacement.

Piezoelectric accelerometer is widely used for shock and vibration measurements. In general, it doesn’t give output for constant acceleration because of the basic characteristics of piezoelectric motion transducer, but it do have large output voltage signal, small size, and can have very high natural frequency. No damping is provided, with material hysteresis being the only source of energy loss. This result in a very low (about 0.01) damping ratio, but this is acceptable because of the very high natural frequency. The design details of piezoelectric accelerometers can emphasize selected features of performance desired for particular application; no single configuration is ideal for all situations since tradeoffs exit here just as in all engineering design. Several designs have been developed for piezoelectric accelerometer and one of the most interested design scheme is the delta shear, shear design use bolted stacks of flat plate element has been introduced recently to gain further improvement in performance [17].

3. Analysis Of Vibration Signal

One of several quantities that could be used to describe the vibration effects is peak value either it is displacement, velocity, or acceleration form of vibration, on the other hand, more complex vibrations are being studied other descriptive quantities may be preferred. One of the reasons for this is that the peak value describes the vibration in terms of a quantity, which depends only upon an instantaneous vibration magnitude regardless of the time history producing it. A further descriptive quantity that does take the time history into account is the Root Mean Square (RMS) value and could be formulated as

$$X_{RMS} = \sqrt{\frac{1}{T}\int_0^T X^2(t)\,dt} \qquad (1)$$

X_{RMS} : The RMS value of the vibration signal.

T : Period of vibration signal. ; t : Time (sec).

The importance of RMS value comes from its simple relationship to the power content of the vibration, even with more complex form of vibration signal such as random one; it will be suitably meaningful take the RMS value of the signal [18]. Fig.6. represents a random vibration signal.

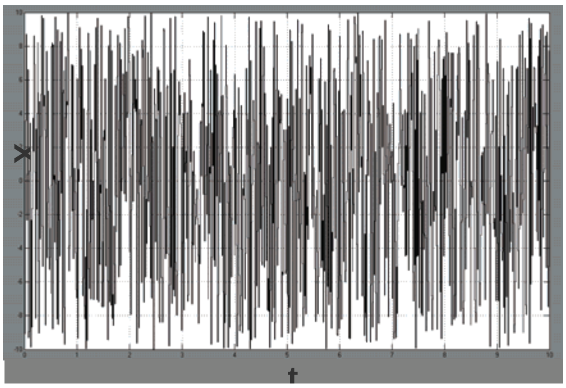


Figure 6: Random vibration signal

Some tests have been done to ensure the randomness of the vibration signal by comparing some of its statistical information with those of a periodic sin wave signal. The tests results are shown in table(1) & Fig.(7).

Table 1: sine wave and random vibration signal statistics data

Statistical data	Sine wave signal	Random signal
Mean	0.034526	0.3287
STD	7.0698	5.8071
RMS	7.0348	5.7876

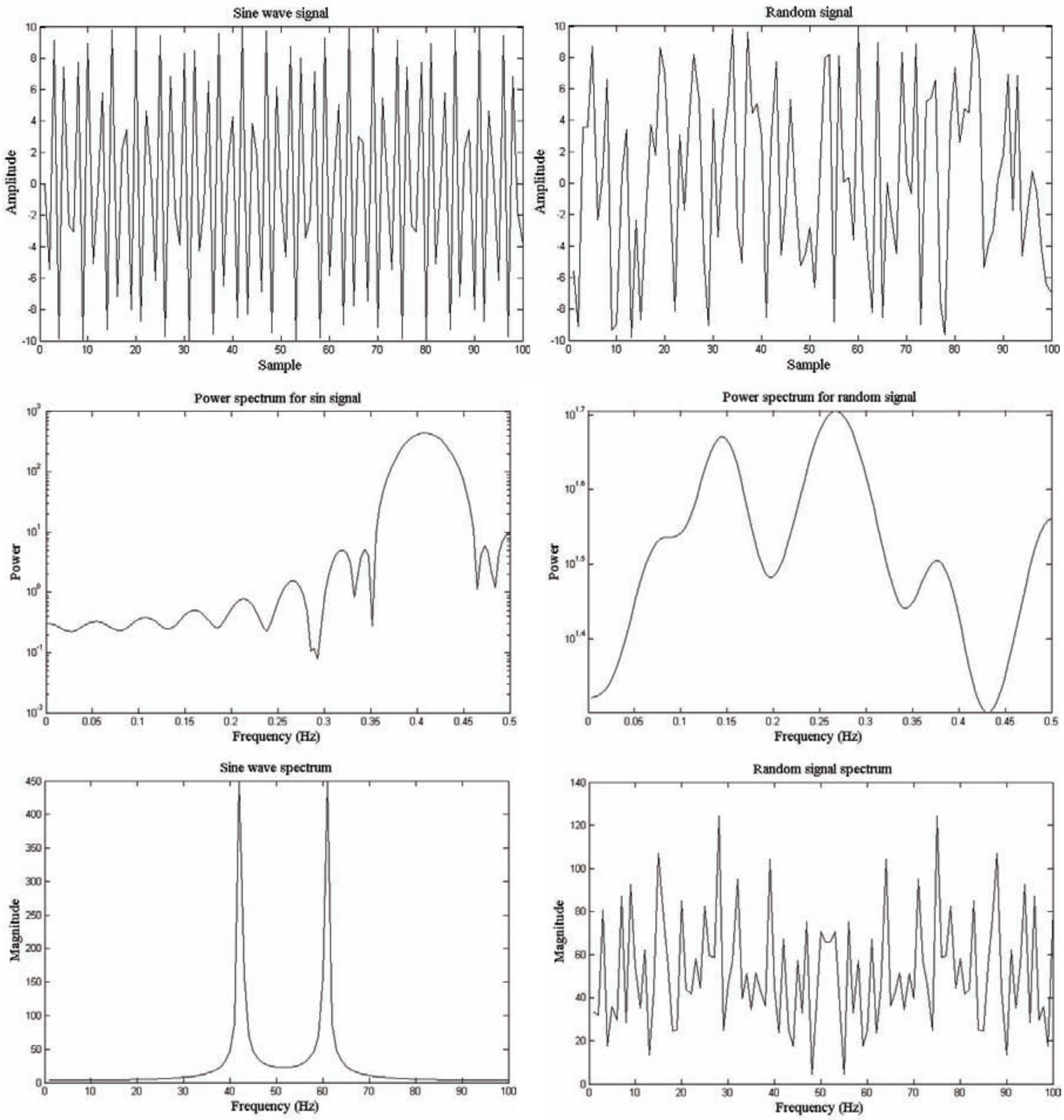


Figure 7: Sine wave and random vibration signal tests results

3.1. Vibration Signal Acquisition

It was found that the selected accelerometer type (4370) was suitable for our application since it has acceleration measurement range of (0.0002 –20000) m/s^2 and a voltage sensitivity of (8.5) mV/ms^{-2}.

3.2. Signal Amplification

Since the vibration signal produced by the accelerometer is too small to be read directly by the A/D converter and by the PC parallel port, the signal should be enlarged using a special type of amplifiers called “instrumentation amplifier”. Those amplifiers have special properties such as its high input impedance, low noise, and moderate bandwidth [17]. Those properties have been satisfied using a conditioning amplifier type (2626) [19].

3.3. Calibration Process

Since the measurement system is often made up of a chain of components, each of which is subject to individual inaccuracy, it will be important to know how these inaccuracies may affect the over all system measurement precision. The most common method to do this is to find the least square criterion, which minimizes the sum of the squares of the vertical deviations of the data points from the fitted line. The algorithm explained briefly in the appendix C for single component. But for chain of components, the collected information of each individual component calibration should be taken into consideration according to a specified procedure [17].

Assume a case where a computing quantity K .

$$K = f(u_1, u_2, u_3, ..., u_n) \tag{2}$$

K :Known function of the n independent variables $u_1, u_2, u_3, ..., u_n$. The $u's$ are the measured quantities (instrument or component outputs) and are in error by $\pm \Delta u_1, \pm \Delta u_2, \pm \Delta u_3, ..., \pm \Delta u_n$, respectively. These errors will cause an error ΔK in the computed result K. The $\Delta u's$ may be considered as absolute error.

$$K \pm \Delta K = f(u_1 \pm \Delta u_1, u_2 \pm \Delta u_2, u_3 \pm \Delta u_3, ..., u_n \pm \Delta u_n) \quad (3)$$

By subtracting K in equation (2) from $K \pm \Delta K$ in equation (3), we finally obtain $\pm \Delta K$ which is needlessly time consuming procedure; however an approximate solution valid for engineering purposes may be obtained by application of the Taylor series. Expanding the function f in a Taylor series, we get

$$f(u_1 \pm \Delta u_1, u_2 \pm \Delta u_2, u_3 \pm \Delta u_3, ..., u_n \pm \Delta u_n) = f(u_1, u_2, u_3, ..., u_n) + \Delta u_1 \frac{\partial f}{\partial u_1} + \Delta u_2 \frac{\partial f}{\partial u_2} + \Delta u_3 \frac{\partial f}{\partial u_3} + ... + \Delta u_n \frac{\partial f}{\partial u_n} + \frac{1}{2}\left[(\Delta u_1)^2 \frac{\partial^2 f}{\partial u_1^2} + ...\right] + ... \quad (4)$$

In actual practice, the $\Delta u's$ will all be small quantities and thus terms such as $(\Delta u)^2$ will be negligible. Then equation (4) may be given approximately as

$$f(u_1 + \Delta u_1, u_2 + \Delta u_2, u_3 + \Delta u_3, ..., u_n + \Delta u_n) = f(u_1, u_2, u_3, ..., u_n) + \Delta u_1 \frac{\partial f}{\partial u_1} + \Delta u_2 \frac{\partial f}{\partial u_2} + ... + \Delta u_n \frac{\partial f}{\partial u_n} \quad (5)$$

So absolute error E_a is given by

$$E_a = \Delta K = \left|\Delta u_1 \frac{\partial f}{\partial u_1}\right|_{u_1} + \left|\Delta u_2 \frac{\partial f}{\partial u_2}\right|_{u_2} + \left|\Delta u_3 \frac{\partial f}{\partial u_3}\right|_{u_3} + ... + \left|\Delta u_n \frac{\partial f}{\partial u_n}\right|_{u_n} \quad (6)$$

When the $\Delta u's$ are considered not as absolute limits of error, but rather as statistical bounds such as $\pm 3S$ limits. The equation (6) modified to the root – sum square (rss) formula.

$$E_{arss} = \sqrt{\left(\Delta u_1 \frac{\partial f}{\partial u_1}\right)^2 + \left(\Delta u_2 \frac{\partial f}{\partial u_2}\right)^2 + \left(\Delta u_3 \frac{\partial f}{\partial u_3}\right)^2 + ... + \left(\Delta u_n \frac{\partial f}{\partial u_n}\right)^2} \quad (7)$$

the measurement system consists of two major parts. The first part (accelerometer and charge amplifier), and the second part (interface card and host PC). Our individual calibration process for parts one and two of the measurement system shows the following results.

First part curve-fitting equation

$y_1 = 0.97x$, $S = 0.00494$

Second part curve-fitting equation

$y_2 = 0.95x - 0.026$, $S = 0.146$

Where S is the standard deviation value STD

Assume $\pm 3S$ limits, E_{arss} could be evaluated as follows.

$$E_{arss} = \sqrt{(3*0.00494*0.97)^2 + (3*0.146*0.95)^2} = 0.46$$

$\cong 0.5$

Percentage inaccuracy =

$$\frac{E_{arss}}{full\ scale} * 100\% = \frac{0.5}{10} * 100\% = 5\%$$

Fig.8, and Fig.9. Show part one and part two calibration curve fitting figures respectively.

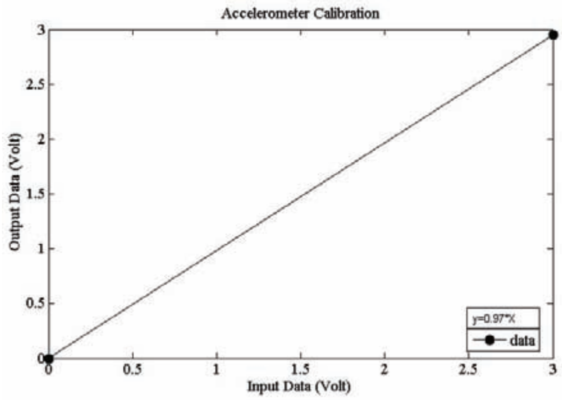


Figure 8: Accelerometer calibration

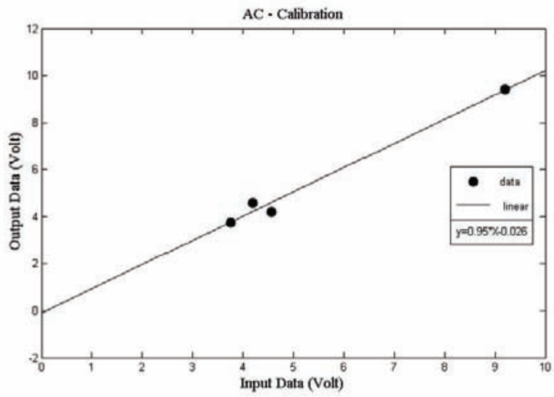


Figure 9: AC – signal calibration

4. Experimental Cutting Test

The experimental tests accomplished using turning machine, and the cutting tool used for the process was HSS with medium carbon steel as a work piece. The tests have been done with a constant depth of cut equal to (0.6mm), with no cooling fluid. Fig.10. demonstrates the experimental work layout. Table .2 . shows the statistical information obtained at each cutting process. Figs 11-14 show some experimental reading for the acceleration at cutting tests.

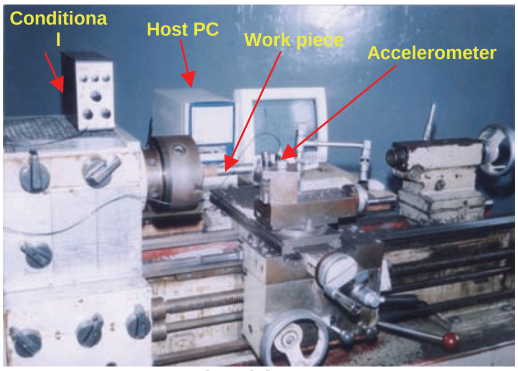


Figure 10: Experimental work layout

Table 2: Statistical information obtained by experimental work

Test	Speed (rpm)	Feed (mm/rev)	RMS	Mean	STD
1	540	0.03	4.9767	-0.0240	4.9794
2	540	0.04	4.8862	0.2680	4.8816
3	540	0.05	4.9206	-0.0444	4.9231
4	540	0.1	5.4808	0.3021	5.4756
5	540	0.06	4.7578	-0.0258	4.7604
6	260	0.06	3.7975	0.0036	3.7997
7	260	0.03	5.1199	0.2867	5.1148
8	260	0.04	5.3227	0.1029	5.3347
9	260	0.05	5.0697	0.3848	5.0579
10	260	0.08	5.3073	0.3794	5.2967
11	370	0.1	5.0680	0.1484	5.0687
12	370	0.08	4.8541	0.3651	4.8431
13	370	0.06	4.1390	0.2286	4.1350
14	370	0.04	4.4053	0.0160	4.4077
15	370	0.03	4.0282	0.3876	4.0117
16	125	0.06	3.7041	0.2813	3.6955
17	125	0.08	3.4407	0.1027	3.4411
18	125	0.1	2.9997	-0.0269	3.0013
19	125	0.04	3.3285	0.1664	3.3262
20	125	0.05	3.1207	-0.0336	3.1223
21	180	0.05	3.8133	0.0602	3.8150
22	180	0.08	3.9633	0.0463	3.9653
23	180	0.06	2.7079	-0.0342	2.7092
24	180	0.1	3.0113	0.0366	3.0128
25	180	0.04	2.6154	-0.1283	2.6137
26	180	0.03	2.9496	-0.1303	2.9484
27	85	0.1	2.3361	-0.0314	2.3372
28	85	0.08	2.1937	-0.0454	2.1945
29	85	0.06	2.2187	0.0987	2.2177
30	85	0.04	2.3136	0.1203	2.3118

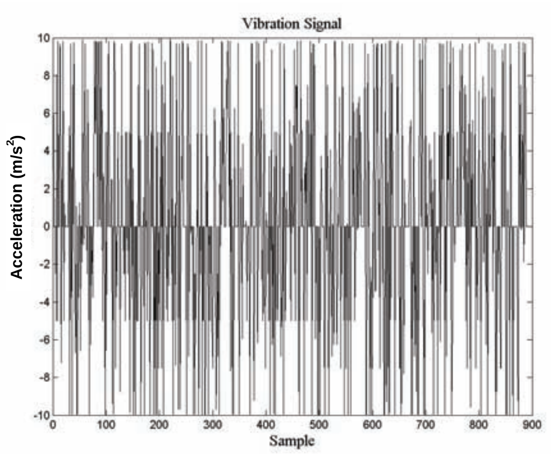


Figure 11: Test (No.4)

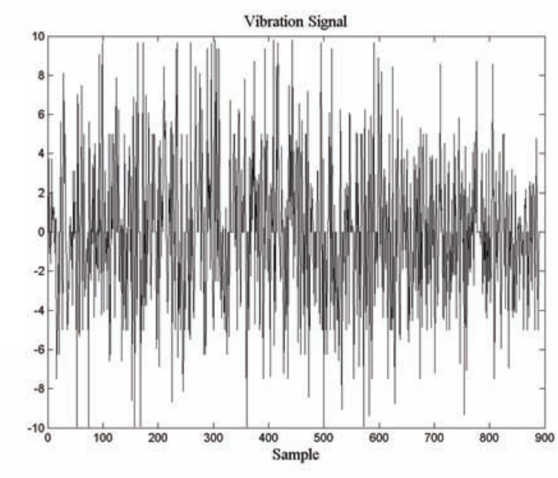


Figure 12: Test (No. 6)

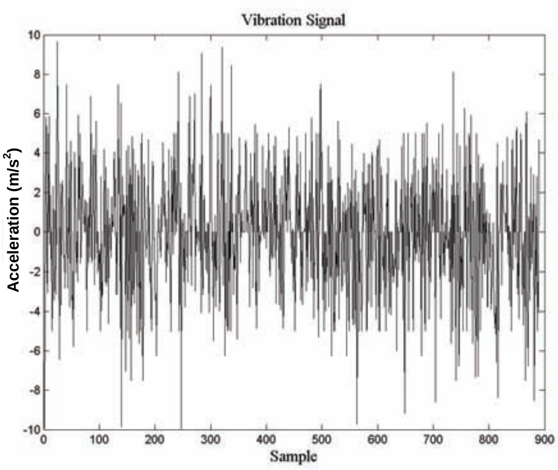


Figure 13: Test (No.20)

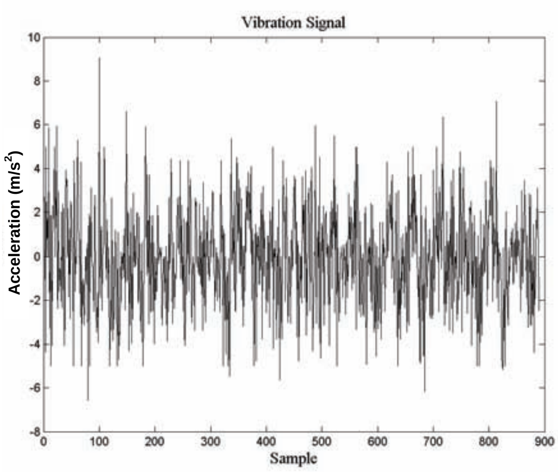


Figure 14: Test (No.27)

5. Network Architecture

Commonly one neuron, even with much input, may not be sufficient. A need for more complicated network architecture arises that serve as multi neurons operating in parallel in what is called “layer”. Sometimes multi layers are more powerful than the single layer network. For instance, a two-layer network having a sigmoid transfer function in the first layer and a linear transfer function in the second layer can be trained to approximate most functions arbitrarily well [20]. For more demonstration Fig.15. shows two-layer network.

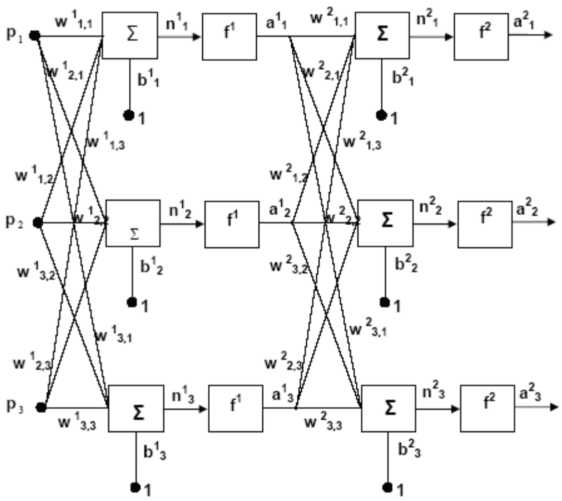


Figure 15: Two layer network arcitecture

It will be more convenient to describe the input/output mathematical relationship in matrix form, where:

$$a^2 = f^2\left(\left[w^2\right] * f^1\left(\left[w^1\right] * \{p\} + \{b^1\}\right) + \{b^2\}\right) \tag{8}$$

or

$$\begin{Bmatrix} a_1^2 \\ a_2^2 \\ a_3^2 \end{Bmatrix} = f^2\left(\begin{bmatrix} w_{1,1}^2 & w_{1,2}^2 & w_{1,3}^2 \\ w_{2,1}^2 & w_{2,2}^2 & w_{2,3}^2 \\ w_{3,1}^2 & w_{3,2}^2 & w_{3,3}^2 \end{bmatrix} * f^1\left(\begin{bmatrix} w_{1,1}^1 & w_{1,2}^1 & w_{1,3}^1 \\ w_{2,1}^1 & w_{2,2}^1 & w_{2,3}^1 \\ w_{3,1}^1 & w_{3,2}^1 & w_{3,3}^1 \end{bmatrix} * \begin{Bmatrix} p_1 \\ p_2 \\ p_3 \end{Bmatrix} + \begin{Bmatrix} b_1^1 \\ b_2^1 \\ b_3^1 \end{Bmatrix} \right) + \begin{Bmatrix} b_1^2 \\ b_2^2 \\ b_3^2 \end{Bmatrix} \right) \tag{9}$$

In spite of existence of several other network architecture, that could be useful for a lot of applications, and it has been used in this research.

6. Neural Network Training

It is important now to know how the weights and biases of a network could be determined. With complex network, having many inputs and complicated architecture, the training algorithms solve this problem. The training algorithms (learning rules) could be defined as “a procedure for modifying the weights and biases of a network in order to train the network to perform some task” [20].

6.1. General Architecture Selection

One of the problems that occur during neural network training is called overfitting. The error on the training set is driven to very small value, but when new data is presented to the network the error becomes large. The network has memorized the training examples, but it has not learned to generalize to new situation.

One method for improving network generalization is to use a network that is just large enough to provide an adequate fit. The larger the network, the more complex the functions the network can create. If we use a small network, it will not have enough power to over fit the data. Unfortunately, it is difficult to know before hand how large a network should be for a specific application .

6.2. Data Pre – Processing

Neural network can be made more efficient if certain preprocessing steps are performed on the network inputs and targets. This subsection describes some common processing techniques that could be used to make training process more effective.

6.3. Process Dependency on Min and Max of Data Values

Before training, it is useful to scale inputs and targets so that they always fall within a specified range. This process makes the data fall in the range [-1, 1]. After the completion of training process the network out put should be converted back into its original units that were used for the original targets.

6.4. Process Dependency on Mean and STD of Data Values

The original network inputs and outputs are given in the matrices p and t respectively. The normalized inputs and targets that are returned will have zero mean and unity standard deviation. Also the network outputs should be converted back to the original units of the targets.

The data pre – processing used for the data depends on normalizing it according to data’s maximum and minimum values. This process carried out for both input “feed, speed” vector and output “vibration RMS” vector. Equation 10 shows the normalization equations.

$$p_n = \left(2 * (p - \min p) \big/ (\max p - \min p)\right) - 1 \tag{10}$$

Where p : Input matrix

p_n : Normalized input matrix.

$\min p, \max p$: Minimum and maximum input value in the matrix.

6.5. Data Post – Processing Analysis

The performance of a trained network can be measured on some extent by the errors on the training. But it is often useful to investigate the network response in more detail. One option is to perform a regression analysis between the network response and the corresponding targets. The process will return three parameters. The first two parameters, (m and b), correspond to the slope and the y – intercept of the best linear regression relating targets to the network outputs. The third variable that returned by the post processing is the correlation coefficient (R – value) between the outputs and the targets. It is a measure of how well the variation in the output is explained by the targets. If this number is equal to (1), then there is perfect correlation between the targets and outputs [22].

7. Neuro – Controller

Most of the neuro –control schemes developed until now are based on the following design approaches [20]. Series control scheme: the neural network directly learns the mapping from the desired reference signal to the control inputs, which yields these signals.

Parallel control scheme: a neural network is used to compensate the control signal which is provided by conventional controller such that the plant output can track the desired output as close as possible.

Self – tuning control scheme: a neural network tunes the control parameters including the conventional controller such that the plant output follows the desired output signal as much as possible.

Emulator and controller scheme: it maximizes some measure of utility or performance over time, but can't efficiently account for noise and can't provide real – time learning for slow convergence, also known as backpropagation - through - time.

The self – tuning control scheme has been used in this research and it is explained as follows.

7.1. Self - Tuning Neuro – Control Scheme

The self – tuning neuro – control scheme is illustrated in Fig.16., where a neural network is used to tune the parameters of a conventional controller similar to adjustment made by a human operator. The process need that the human operator has a moderate amount of experience and a great knowledge on the control system, however, unlike the computer, it is rather impossible for the operator to store past data history of the system for any kind of operating condition. If one can include the experience and the knowledge of the operator into a neural network and train it based on the past data history, then the trained neural network could be used as means to tune the controller parameters in an on – line way[23].

8. Design of the Self - Tuning Neuro - Controller

Neural network has been applied very successfully in the identification and control of dynamic systems. The universal approximation capabilities of the multilayer perceptron MLP make it a popular choice for modeling nonlinear systems and for implementing general-purpose nonlinear controller The following describes the process of design of the neural network controller. There are typically two steps that involved when using neural network for control .

1. System identification.
2. Control design.

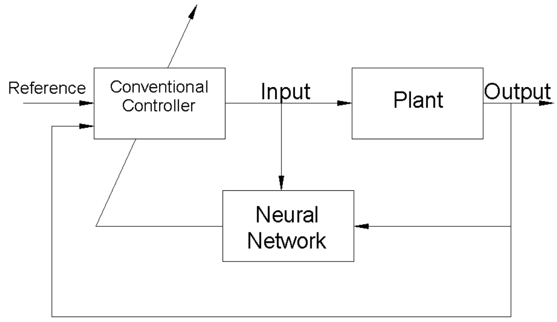


Figure 16: tuning neuro – control scheme

9. Design of the Self - Tuning Neuro - Controller

Neural network has been applied very successfully in the identification and control of dynamic systems. The universal approximation capabilities of the multilayer perceptron MLP make it a popular choice for modeling nonlinear systems and for implementing general-purpose nonlinear controller The following describes the process of design of the neural network controller. There are typically two steps that involved when using neural network for control .

1. System identification.
2. Control design.

As described previously in system identification stage, a model for the system want to be controlled should be developed. In control stage the developed model should be used in training the controller.

This controller uses a neural network model to predict future plant responses to potential control signals. An optimization process then computes the control signal that optimizes the future plant performance. The predictive control process is based on receding horizon technique. The neural network model predicts the plant response over a specified time horizon. The predictions are used by a numerical optimization process to determine the control signal that minimizes the performance criterion "equation (11)" over the specified horizon. Fig..17. demonstrates the controller optimization procedure. This controller has been designed and used for this work.

$$Z = \sum_{j=N_1}^{N_2} \left(y_r(t+j) - y_m(t+j) \right)^2 + \rho \sum_{j=1}^{N_u} \left(u'(t+j-1) - u'(t+j-2) \right)^2 \qquad (11)$$

Z : Optimization performance index.

N_1, N_2, N_u : The horizons over which the tracking error and the control increments are evaluated.

u' : The tentative control signal "Tentative Feed and Speed Values".

y_r : The desired response "Desired Vibration RMS Value".

y_m : The network model response “Vibration RMS Value Developed by the Model”.

ρ : Factor determines the contribution that the sum of the square of the control increments has on the performance index.

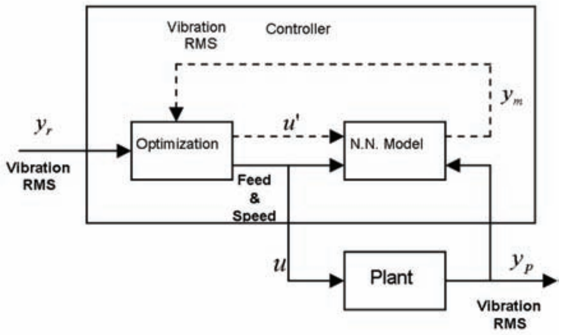


Figure 17: optimization procedure

10. Result and Discussion

The designed controller has been used to control turning process identifier by applying different reference signals to it as test inputs, to find out its ability to follow the desired response; the controller is fed by the reference signals, which were the vibration RMS values, and the responses that have been shown on the scope. The controller here is designed to generate the values of the feed and the speed and then the neural network turning process identifier maps it into vibration RMS values, the controller reads back the output signal and compare it to the signal results from its neural network model where the optimization algorithm update the control action (feed, speed) signals so that the turning process identifier follow the required reference signal. Fig18. Demonstrate the controller model training process. The learning rate (α) set to (0.75), and control process maximum error set to ($1*10^{-3}$), which make the training process reached to the specified goal at (199) epoch. The designed controller model has been trained by the backpropagation algorithm described in chapter four, it consist of two layer perceptron neural network with eight neuron in the hidden layer and one neuron at the output layer.

The controller model designed with “*tansig*” activation function in the hidden layer and “*purelin*” activation function in the output layer. Fig.19. shows the turning process controller. The optimization for process control is simulated by the flowchart shown in Fig.20. Different reference signals have been used for testing the response of the controller .The first was for input reference equal to 5, the second was for input reference 4.5 for the first two second and then the reference decreased to 3.5. The final case was the response of the controller for input reference signal started with 3.5 and then increased to 4. The sampled time used was (0.01 sec “100 sample = 1 sec”).

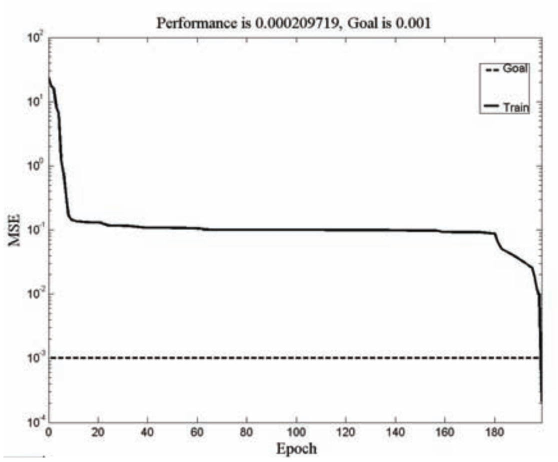


Figure 18: Controller model training process

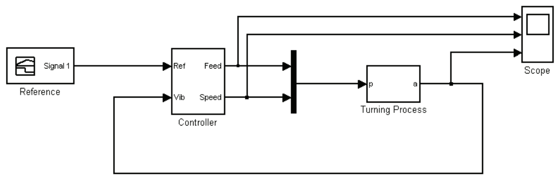


Figure 19:Turning process controller

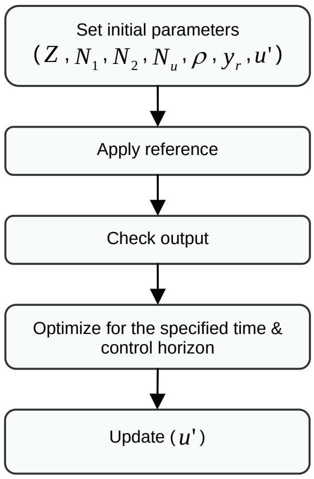


Figure 20: Optimization flowchart

The design of system identifier shows precise results with (10neuron), where the training performance index set to ($1*10^{-8}$) as MSE of the output of the neural network and the trained network met that goal since the error between the neural network output and the targets did not exceeds ($1.55*10^{-4}$). For the designed neural network controller, it is found that the controller track the reference signals set, where setting the reference signal to (5) as acceleration RMS value as shown in fig.22. made the controller after a starting with an initialization values of the feed and the speed be changed. For reference signal equal to (5) the controller adapts new feed and speed values 0.095mm/rev, and 440RPM respectively, this adaptation completes after (100samples = 1sec) “setting time” and the response progresses with a steady state error that doesn’t exceeds (8%) of the reference signal due to the fluctuating speed value. Two different reference signals have been set to check the ability of the controller to track these specified reference signals as shown in Fig.23, where setting of the reference signal to (4.5) as acceleration RMS value made the controller adapt new feed and speed values equal to 0.085 mm/rev and 440 RPM respectively, the controller reaches to the specified reference after (300 samples = 3 sec) “settling time”. The new reference signal (3.5) has been tracked by the controller after (400samples = 4sec) as settling time and the adapted feed and speed values were 0.073mm/rev and 270RPM. Finally new two reference signals (3.5,4) have been set as acceleration values as shown in Fig.23., the controller changes it initial values (feed = 0.035 mm/rev, speed = 270 RPM) to 0.068 mm/rev and 330 RPM to reach the reference signal within (200samples = 2sec).

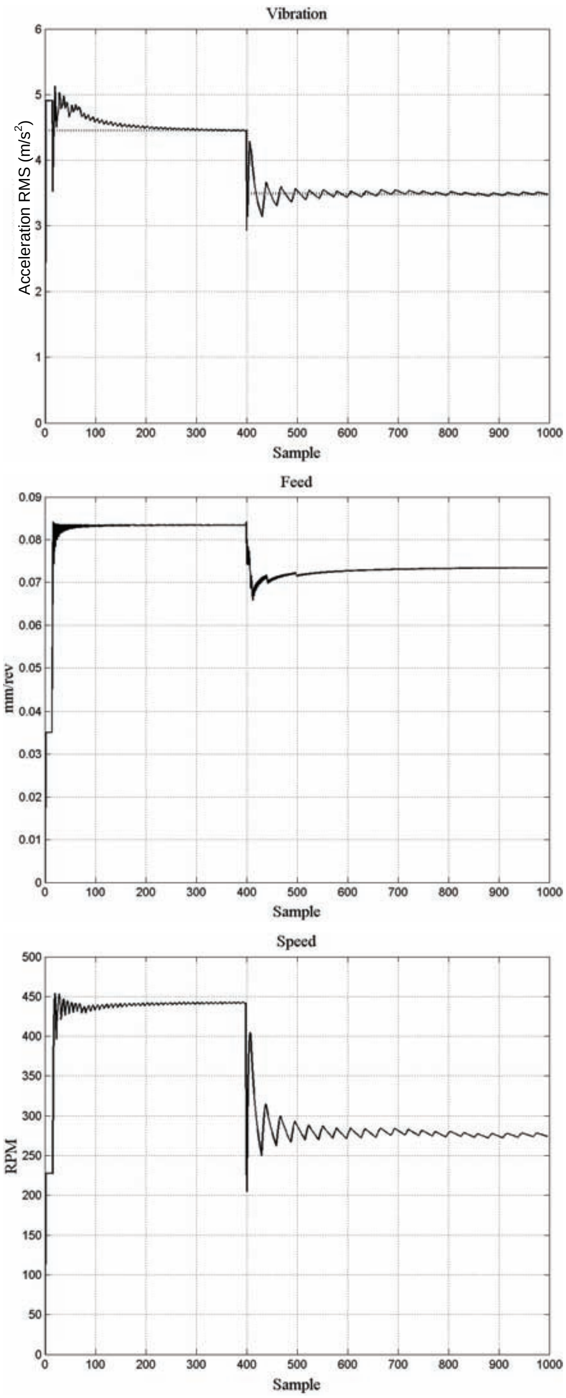


Figure 21: Controller response (input reference = 5)

Those reference signals tracking process that have been achieved by the Neuro - controller show the ability of the controller to track the reference signal with minimum settling time equals to (100samples = 1sec) and maximum settling time equals to (300samples = 3sec) and a maximum overshot for the test signals that doesn’t exceeds (30%) of the reference signals. Fig.21, Fig.22, and Fig.23 show that increases the value of reference (acceleration RMS) not necessarily leads to increase the control signal (Feed, Speed) neither decrease it should decrease the control signal.

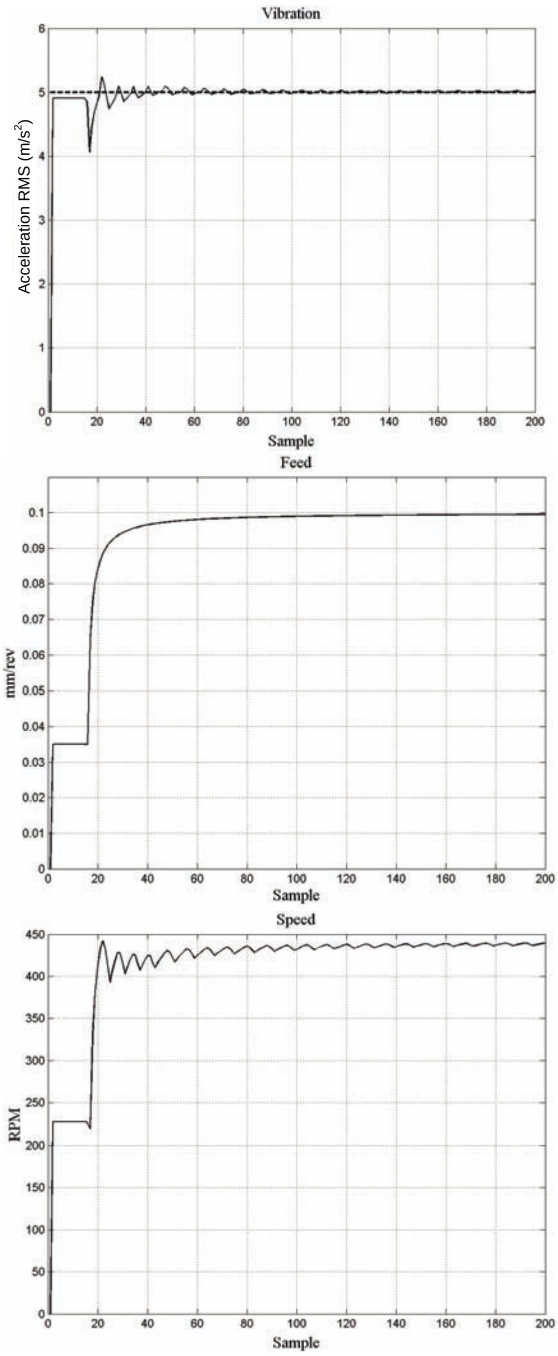


Figure 22: Controller response (input reference = 5)

11. Conclusion

In this work, the effectivness of using neural networks as identification and as an alternative to adaptive controller of metal cutting process are inverstigated. Also, using the neural network for system identification releases the controller designer from the problem of modeling complex real systems and the confusing related to selection of the least significant system variables which can be ignored.

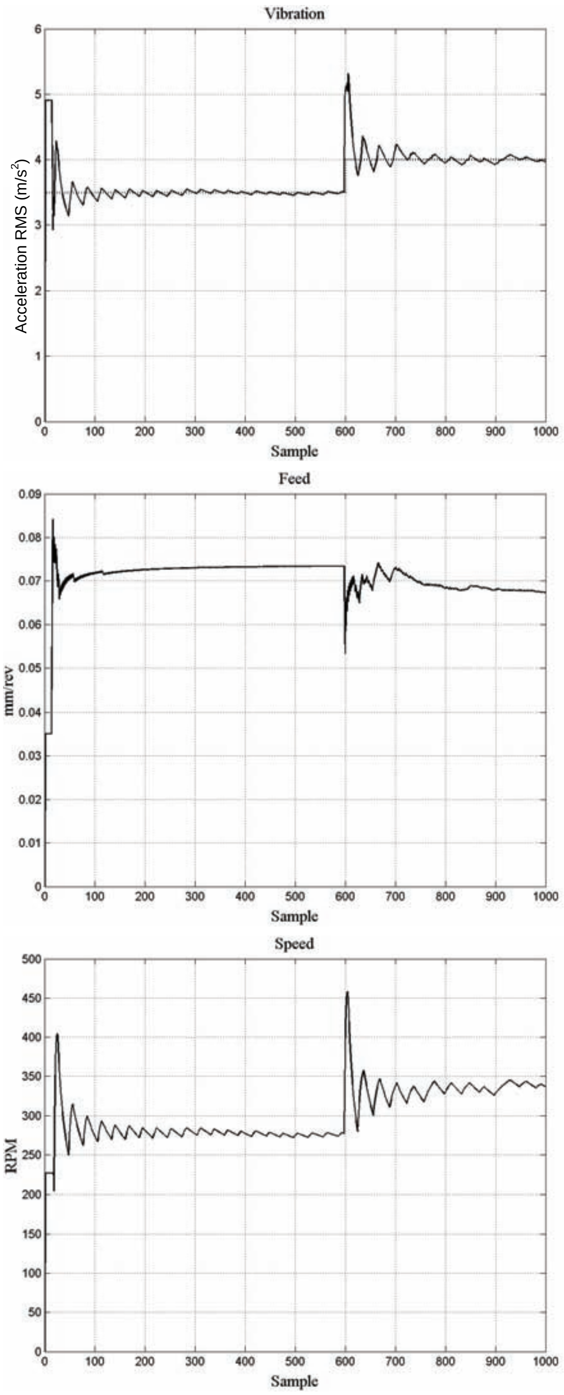


Figure 23: Controller response (input reference = 3.5,4)

The using of LM (Levenberg-Marquardt) training algorithm on the MLP network even with its property of needing to large memory is successful algorithm in minimizing the training error, which makes it good training algorithm for system identification with high degree of accuracy, while the traditional Back propagation algorithm can be used to train neural networks with greater error allowance. The controller proposed in the current work follow the desired response with control actions (Feed, Speed) not mentioned in the training data and this make the neural network has an advantage in being work as intelligent map. Few practical experiments used for training the neural network (identifier, controller) may cover the process with less error and this will minimize the efforts of achieving a lot of practical experiments if compared with the other traditional controllers.

References

[1] M. P. Groover "Fundamental of modern manufacturing process". by Prentice Hall International Inc.1996.

[2] P. Tai, H. A. Ryaciotaki, and D. Hollaway "Neural network implementation to control system". A survey of algorithms and techniques. Transaction of IEEE, Vol.25, No.3,212-217, 1991.

[3] P. E. Keller, R. T. Kouzes, L. J. Kangas, and S. Hashem "Neural network based system for manufacturing application". A paper represented at the advanced information system and technology conference in Williamsburg, VA, USA (28-30) March (1994).

[4] Y. Koren "Design of computer control for manufacturing system". Transactions of the American society of mechanical engineering (ASME), Vol.101, 1979. 326-331.

[5] V.V. Athani and H. N. Vinod"A CNC system for a lathe using a low cost PC". Computers in industry, Vol. 7, PP. 427-434, 1986.

[6] P. U. Achi "A low – cost computer numerical control machine system using LSI interfaces". IEEE transactions on industrial electronics, Vol. 34, No. 1, 1987.

[7] Y. Altintas and J. Peng "Design and analysis of a modular CNC system". Computers in industry, Vol. 13, 1990. 305-316.

[8] T. George, C. Chiu, and M. Tomizuka "Coordinated position control of multi-axis mechanical system". Journal of dynamic systems, measurement, and control, 1998. (389-393).

[9] L. Jeffery, H. Stein, and K. Huh "Monitoring cutting forces in turning: A model based – approach". Vol. 124, 2002. 26-31; ASME.

[10] R. G. Khanchustambham, and G. M. Zhang "A neural network approach to on - line monitoring of a turning process", Transaction of IEEE, Vol. 12, No.5, 1991.107-113.

[11] L.G. Alan "Neural network servo control for ultra-precision machining at extremely low feed rate". PhD Thesis, University of Illinois at Chicago, 1996.

[12] R. Azouzi and M. Guillot "On-line optimization of the turning process using an inverse process neurocontroller". Journal of Manufacturing Science and Engineering, Vol. 120, 1998,101-108.

[13] T. Özel and A. Nadgir "Prediction of a flank wear by using back- propagation neural network modeling when cutting hardened H-13 steel with chamfered and honed CBN tools" International journal of machine tools & manufacturing design research and application, 2001.

[14] Jeng Chen, Qiang Liu and Chang Qi" A Synchronous Drive Control Scheme Based on Neural Networks for a Novel XY-table" Proceedings of the IEEE International Conference on Automation and Logistics August 18 - 21, 2007, Jinan, China

[15] S. Kalpakjian "Manufacturing Engineering and technology" 4'th edition , by Prentice Hall International Inc, 2001.

[16] A. K. Sawhney "A course in mechanical measurements and instrumentation". Second edition 1995, by Dhanpat rai and sons.
[17] E. O. Doebelin "Measurement systems application and design" Fourth edition 1990, by McGraw –Hill publishing company.
[18] J. T. Broch "Mechanical vibration and shock measurements". First edition 1980, K. Larsen & Son A/S for publishing.
[19] Brüel & Kjaer corporation "Piezoelectric accelerometer and vibration preamplifier hand book". K. Larsen & Son A/S for publishing, 1978.
[20] M. T. Hagan, H. B. Demuth, and Mark Beale "Neural network design". First edition 1996, McGraw - Hill publishing company.
[21] S. Omatu, M. Khalid, and R. Yusof "Neuro – control and its applications". Springer publishing company, 1994.
[22] O. Omidvar and D.L. Elliot "Neural system for control"., Springer publishing company, 1997.
[23] D. T. Pham and L. Xing "Neural networks for identification, prediction, and control", Springer publishing company, 1997.