Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Ruturaj Jayant Kulkarni

Entitled
AI-BASED MODELING AND OPTIMIZATION OF TURNING PROCESS

For the degree of Master of Science in Mechanical Engineering

Is approved by the final examining committee:

Hazim A. El-Mounayri
Chair

Sohel Anwar

Tamer Wastfy

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Hazim A. El-Mounayri

Approved by: Sohel Anwar 06/29/2012
Head of the Graduate Program Date

Graduate School Form 20
(Revised 9/10)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:

AI-BASED MODELING AND OPTIMIZATION OF TURNING PROCESS

For the degree of Master of Science in Mechanical Engineering

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22,* September 6, 1991, *Policy on Integrity in Research.**

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Ruturaj Jayant Kulkarni

Printed Name and Signature of Candidate

06/29/2012

Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

A.I.- BASED MODELLING AND OPTIMIZATION OF TURNING PROCESS

A Thesis
Submitted to the Faculty
of
Purdue University
by
Ruturaj Jayant Kulkarni

In Partial Fulfillment of the
Requirements for the Degree
of
Master of Science in Mechanical Engineering

August 2012
Purdue University
Indianapolis, Indiana

ACKNOWLEDGMENTS

I would like to express my deepest gratitude and thanks to my adviser Dr. H. A. El-Mounayri for his guidance, motivation, patience and support, throughout the course of this research. I warmly thank Dr. Sohel Anwar and Dr Tamer Wastfy for being in my advisory committee. I would also like to express my sincere thanks to Dr.Snehashish Mukhopadhyay, for doing a great job in teaching me the concepts of Computational Intelligence and for assistance in the implementation. I also thank Dr.Ilhan Asilturk, Harpreet Banvait. I would like to extend sincere appreciation to the staff of Purdue School of Engineering, R. Earlson, Ginger Lauderback, Aimee Wood, for their help and constant support.

Most importantly, I owe gratitude to my parents, family and friends who stand by me always and make everything possible. Their trust, encouragement, support and prayers have been of immense help to me in this endeavor.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Kulkarni, Ruturaj Jayant. M.S.M.E., Purdue University, August 2012. A. I. – based Modeling and Optimization of Turning Process. Major Professor: Hazim El-Mounayri.

In this thesis, Artificial Neural Network (ANN) technique is used to model and simulate the Turning Process. Significant machining parameters (i.e. spindle speed, feed rate, and, depths of cut) and process parameters (surface roughness and cutting forces) are considered. It is shown that Multi-Layer Back Propagation Neural Network is capable to perform this particular task. Design of Experiments approach is used for efficient selection of values of parameters used during experiments to reduce cost and time for experiments. The Particle Swarm Optimization methodology is used for constrained optimization of machining parameters to minimize surface roughness as well as cutting forces. ANN and Particle Swarm Optimization, two computational intelligence techniques when combined together, provide efficient computational strategy for finding optimum solutions. The proposed method is capable of handling multiple parameter optimization problems for processes that have non-linear relationship between input and output parameters e.g. milling, drilling etc. In addition, this methodology provides reliable, fast and efficient tool that can provide suitable solution to many problems faced by manufacturing industry today.

1. INTRODUCTION

Metal forming can be described as a process of working with metals, to create parts, their assemblies, and their structures. It constitutes a wide range of manufacturing processes. About 70% of the metals forming processes are represented by metal cutting which is also a form of subtractive manufacturing. During these procedures, power driven machine tools are utilized to remove the material from the existing geometry or shape to the desired shapes. Cutting processes are divided into several operations which mainly include drilling, milling, grinding, and turning. Considering its applicability and versatility, there should be no doubt that turning is among the most important processes. The following processes can be carried out successfully on the turning machine: Chamfering, a cutting angle on the on the corner of the workpiece for easier mating of the parts; parting, cutting the end of the part by giving radial feed to the tool; threading, to produce either external or internal threads; boring, a single point tool given a linear feed along the axis of rotation. The primary applications would include machine components, shafts, engine components such as pistons, cylinders, pulleys, axles, etc.

1.1 The Turning Operation

Turning is a material removal process, a subtractive form of machining which is used to create parts of circular or rotational form of desired geometry/shape by removing unwanted material. The essential elements of the turning process are machine or lathe, workpiece material which is a piece of a pre-shaped part, the fixture to which the material is attached. The fixture is secured to the turning machine and is then allowed to rotate for a wide range of speeds. The other end of the workpiece is hooked up with the tail stock to allow perfect rotation and avoid eccentric rotations. The conceptual representation of a turning machine is depicted in Figure 1.1.

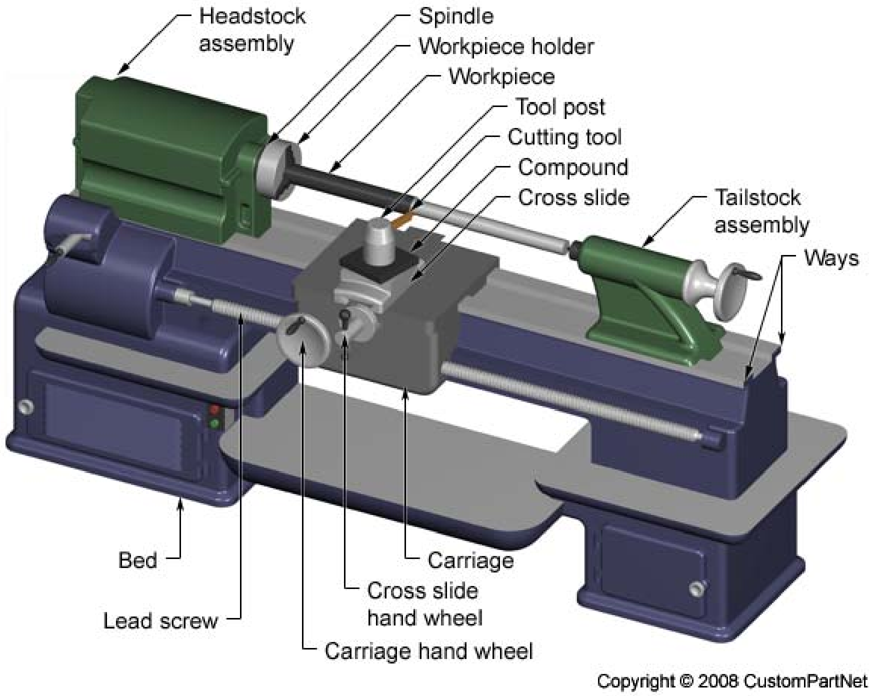


Figure 1.1 Representation of lathe machine

The cutter in the turning operation is usually a single-point cutting tool, except for a few exceptions where multi-point tools are used. The tool is secured to the tool post which is then attached to the machine. During the operation, the cutting tool is fed into the rotating work piece with lateral motion along with the axis of rotation. The unwanted material is removed in the form of chips until the desired shape is attained. Figure 2 shows the cutting process in the turning operation.

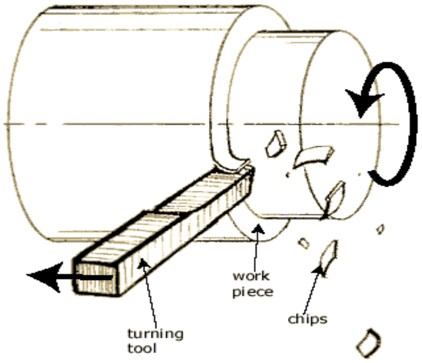


Figure 1.2 Cutting process in turning operation.

Turning is usually opted as a secondary process; it is chosen in order to improve and refine the characteristics and features on parts made by using other processes. Turning is used to produce rotational, typically axi-symmetric, parts that have many features, such as holes, grooves, threads, tapers, various diameter steps, and even contoured surfaces. Parts completely made only on a turning machine are used as prototypes or parts with limited quantity, e.g. designed shafts and fasteners. The turning process offers very high tolerance and good surface roughness; hence, using it for improvements in the already existing part is recommended.

The accuracy of any process depends on involvement of operational variables. The operating parameters that contribute to the turning process are cutting feed (linear distance covered by the tool during one revolution of the workpiece), cutting speed (Speed of the workpiece surface relative to the edge of the cutting tool during a cut), spindle speed (the workpiece's revolution speed per minute), feed rate (linear velocity of the cutting tool with respect to the workpiece), depth of cut (depth of the tool tip with respect to the surface of the workpiece). Vibrations, tool wear, tool life, surface finish, and cutting forces, etc. are also in direct relation with values selected for process parameters. Hence, to improve the efficiency of the process and the quality of the product, it is necessary to control the process parameters.

So far in the industry the trend was such that the selection of the cutting parameters was carried out using previous records or experience. This may not be the most convenient way for the process to work efficiently as it may result in system failure, i.e. loss of tool life or insufficient surface roughness, which will add to the cost of production. Most researchers have shown a limited level of accuracy using either analytical or semi-empirical approaches. Even after considering the complicated nature of the process, researchers have been using a new technique, Artificial Neural Network (ANN) for simulation. In a short time this technique became the favorite since it offers simplicity, accuracy, and robustness.

1.2 Problem Statement

The turning process is one of the oldest and most used processes in the industry. It has a wide range of applications. It is a material removal process. Once the process is complete there is no way that things can be reverted. The surface roughness of manufactured product is outcome of the turning process parameters, and an important characteristics that define product quality, aesthetics etc. Manufactured parts are often rejected, because of failure to comply with the surface quality requirements. Sometimes, if possible, rework or finishing cut can manage to reduce surface roughness and make part acceptable. There is not just one, but many factors responsible for resulting surface roughness e.g. feed, speed, depth of cut, cutting forces, tool geometry etc.

As we know, the choice of cutting parameter values for turning process is mostly made on shop floor by the machinist from his previous experience or from material handbook. However, there is still possibility that the estimated outcome would not occur, as not all the factors can be manually controlled. Therefore, we simulate a model using Artificial Neural Network technique which will predict the outcomes for selection of corresponding cutting parameters. We are interested in finding out optimum values of these cutting parameters, so as to ensure the resultant surface roughness will be minimum. Another objective we want to achieve during this study is to keep the cutting force values low while attaining the desired value of surface roughness for selected process parameters.

Surface roughness requirement is one of the important and necessary requirements that have to be satisfied for standards in the industry. Requirement for quality of surface roughness changes with the application for which the part is being used e.g. aeronautical engineering, food and medicine requires very high level of surface finish i.e about 0.8 to 1 micron, automotive industry needs good surface roughness even though not as high as aerospace industry i.e. anywhere below 1.5 micron, general application products that are not considered in any of the above industry does not requires surface roughness to be very high due to limitations of cost or functionality constraints.

2. LITERATURE REVIEW

2.1 Process Simulation Technique

2.1.1 Mathematical Modeling Technique

Forces acting on the machine tool frame during the material removal process cause static and dynamic distortion. The effort is made in order to understand process physics so as to choose a suitable design for tools, fixtures, and machining parameters [1].To investigate the dynamic characteristics such as cutting forces, displacements, acceleration, etc., simulations of machining processes are effectively used [5,6]. Simulation of the manufacturing process is of integrative type. If it has to include machine, tool and work piece, clamping and machining process as controlling factors. The list of things to predict includes component quality, which is surface roughness, the stability of the system, hence the cutting forces, and then the simulation is effective [10].Using simulations instead of actual experiments allows the researcher to test factors influencing the process, i.e. chatter, tool wear, and tool breakage, and to produce fairly accurate results at a fraction of the cost of actual experimentation. In addition, simulation allows the researcher to test extreme situations without any damage or failure [6]. Unless the models of the cutting process and the machine tool are accurate, simulations of machining will not be effective. It is time consuming and costly to use empirical cutting force models unless the cutting coefficients are already determined experimentally [2]. On the other hand, analytical models use plastic theory to describe the mechanics of the process with available information of cutting conditions and properties of materials [3].

In the case of numerical simulations, nonlinear material behavior and contact conditions between the tool and the material is used for detailed study of chip formation [4].

2.1.2 Computational Programming Technique

Computer programming is also used to simulate the turning operation. The program takes into consideration the tool geometry simulation module and discrete transfer function representing the machining process as well as tool structure [5].Programs are made to run for different steps values of controlling factors to study if simulated results agree with real ones. Along with the computer programs, additional features can also be applied to improve the simulation and also to make it more useful. In article [7], research was carried out to improve the efficiency of the Computer Numerically Control (CNC) machining by allowing additional features of automatic feed rate selection. A combination of low cost noninvasive spindle motor power sensors with geometric and mechanistic cutting process models was made. Selection algorithm chooses the fastest possible feed rate from the feed rate selection planner, when the program is run for one tool move at one instant. Constraints such as surface roughness, tool health, etc., are set. The accuracy of force model coefficients dictates the accuracy of force prediction. To make the force model robust different types of cutting geometries are selected.

If all the necessary boundary conditions are taken into consideration the mathematical equations of the model become so complex that finding the solution for that is not possible[10]. In an analytical approach, if work hardening or non-trivial geometry of the workpiece are to be incorporated, the theory of plasticity leads to a non-solvable set of equations [11].

2.1.3 FEM Technique

In order to overcome the limitations of the analytical method and make the model more feasible, the Finite Element Method (FEM) model of orthogonal cutting is used [1].

Results are best when frictional stress is calculated from normal stress acting on the tool. Use of FEM has been extended up to 2D- 3D FEM code and from the process to the chip formation point of view. Results of FEM simulation of process assist in tool design for manufacturers. It also takes considerable computational time [8]. In FEM 3D code for simulation of the cutting process, orthogonal as well as oblique cutting has been modeled [1, 8]. The thermo-elastic-plastic properties are incorporated in the FEA which results in a better friction model. Along with the friction model, friction coefficient determined by force calibration and sheer flow stresses characterizes the tool chip interaction. Results or predictions from FEA model are consistent as per sensitivity analysis. The friction model proposed in this approach is valid and tool chip interaction is the most dominant factor [9]. However, in FE based simulation models general deviation is noticed for predicted feed and thrust forces from actual experimental measurements [12, 13, 14, 15 and 16].

2.1.4 ANN Technique

Considering the needs of a fast growing manufacturing industry, researchers came up with a new alternative to avoid deviation in prediction, a technique which operates far differently from traditional simulation techniques. This technique is Artificial Neural Network (ANN). It is a relatively new technology. It operates on a philosophy similar to that of biological nervous systems. This technique became famous quickly and has been applied throughout the industry for real world problem solving. Artificial Neural Network (ANN) is computational modeling inspired by the neural architecture of the human brain. It is one of the most popular nonlinear mapping systems in A.I. This technique is mainly used for two applications, classification and prediction. The neural network determines the pattern between input and output data by training. It has multiple layers and has a parallel structure. It consists of an Input layer, hidden layers, and output layers. Links between hidden layers and output layers consist of output weight matrix. The ability of neural networks to solve complex and nonlinear problems makes them more suitable for simulation of the manufacturing process.

Superior performance for simulation of the process using the neural network model has been reported. Several researchers (e.g. Risbood and Dixit [17]; Davim and Gaitonde [19]; Bandit Sukswat [21]; Natarajan and Muthu [24]; Sharma and Dhiman [25]; Ezugwu and Fadare [26]; El-Mounayri and Deng[30]) demonstrated the capability of the neural network algorithm to successfully evolve the implicit relationship between the process parameters and the set of output variables. In the last few years, there has been a lot of emphasis on using the Artificial Neural Network technique for simulations of manufacturing processes, describing the relationship between the process and the output parameters. Manufacturing processes are complex and hence, pose a challenge for true mathematical representation. Cutting forces acting on the tool and surface roughness are among the most commonly used output parameters by researchers, considering its importance for product quality, life, while Feed rate, spindle speed and depth of cut are most commonly used input parameters (e.g. Risbood and Dixit [17];Lin and Lee [18]; Davim and Gaitonde [19]; Bandit Sukswat [21]; Natarajan and Muthu [24]; Sharma and Dhiman [25]; Ezugwu and Fadare [26]). The ANN technique is successful because it uses previous data and because of its accuracy in prediction for most complex processes.

While the ANN technique gained pace among researchers, there was one more technique which also caught the eye of the researchers, the Regression based technique. Lin and Lee [18]; Nuez and Simao [20]; Chavoshi and Tajdari [22] have compared the feed forward multilayer neural network with regression based methods and concluded that the multilayer neural network is more useful for process simulation.Efficient chip form monitoring and control through Multi-layer Neural Network was achieved by Suksawat [21] in order to improve machining process reliability, surface quality, and productivity. Similarly Zang and Chen [23] used artificial neural network for In-process Surface Roughness Prediction (INNSRP) as well as In-process adaptive parameter control (INNAPC). This was used to predict the surface roughness of ongoing work and make the changes in feed as per required surface roughness. Taking the research to the next step, Ezugwu and Fadare [26]; Ozel and Karpat [29] also predicted the tool flank wear which extended the use of the neural network for prediction of tool life. El-

Mounayri and Deng [30] used a novel approach by reverse mapping of the ANN model for cutting force estimation. Back propagation networks provide acceptable degree of accuracy, when it comes to simulating different manufacturing process. One of the reasons why BPNN is widely used is its ability to produce reasonable results when presented with inputs that were never seen during training [32].

Multilayer BPNN is slow to converge because of the use of sigmoid nonlinear transformation function. Some researchers chose to use the /a Radial Basis Network because of its versatility, as it uses a/the Gaussian curve to map the values. It is fairly good with function approximation. It is very fast in convergence and easier to define terms of number of characteristic parameters. It is mainly used for pattern recognition. El-Mounayri and Briceno [27] used RBN as an alternative to simulate the manufacturing process. Also, El-Mounayri and Briceno[28] compared the performance of BPNN and RBN for similar processes and concluded that RBN is superior for that case.

To summarize, the advantages of ANN compared with other traditional/nontraditional techniques:

- Only experimental data is required in order to simulate the process, e.g. Input parameters, output parameters, etc.
- Ability to capture dynamic characteristics of the process since data used for training is generated from the actual process.
- It is easier to simulate implicit relationship of complex processes considering its methodology is flexible.
- It is easier to use and faster to apply for any manufacturing process.

Hence in the present thesis, the ANN tool is utilized for simulation of the turning process.

2.2 Optimization of Machining Parameters

Considering the continuous growing need of the industry for manufacturing the products with good quality, long life, and agreeable aesthetic appearance, in order to keep up with the challenges there is no other option but to adapt to new trends in optimization. In process planning it is necessary to determine machining parameters that need to be optimized in order to satisfy the requirement which is most likely to involve either one or more of the machining economics, machining quality, and machining safety. Many approaches from a wide array of available options have been used to optimize turning operations. Su and Chen [33] used a stochastic optimization method based on simulated annealing algorithm and pattern search for machining optimization of the turning process. Cutting time was the parameter of interest in this case. Surface roughness, or quality of product, is one of the favorite parameters for optimization. Tool life, manufacturing cost, being another parameter. For many researchers (e.g. Tzeng and Yang [34]; Palanikumar and Karunamoorthy [35]; Benardos and Vosniakos [36]; Zhang and Chen [38]; Dogwa [39]; Kirby and Zhang [41]) minimizing surface roughness is the task for the optimization. Tool life is also considered as a parameter to be optimized by studying the tool wear by researchers (e.g. Hascalik and Caydas [37]; Yang and Tarng [40]). During every optimization exercise there are some constraints which are mostly limitations of the process. So cutting speed, feed, depth of cut, which are cutting parameters, and also cutting temperature, material specifications, etc., will be constraints . Only the lower and higher boundary for the values of their respective parameters are allowable in order to maintain the productivity of the process. Tandon and Mounayri [32] used an evolutionary computation technique called Particle Swarm Optimization (PSO). PSO is a relatively new technique being used for optimization of non-linear functions [45, 46]. It resulted in machining time reductions up to 35%. Karpat and Ozel also used the PSO technique to find optimum cutting parameters in order to maximize MRR and minimize surface roughness.

2.2.1 Cutting cost

Cutting cost is defined as the cost incurred for cutting of the metal. Cutting cost is calculated for multi-pass roughening as well as single-pass finishing as those are the two categories which divide the turning process. Cutting cost C_M is expressed as.

$$C_M = k_o t_m \tag{2.1}$$

t_m is the cutting time which is expressed as

$$t_m = t_{mr} + t_{ms} \tag{2.2}$$

Cutting time for rough turning is

$$t_{mr} = \frac{\pi\, D\, L}{1000\, V_r f_r} * \mathrm{n} \tag{2.3}$$

Cutting time for finished turning is

$$t_{ms} = \frac{\pi\, D\, L}{1000\, V_s f_s} \tag{2.4}$$

Now cutting cost is calculated by equation (2.1)

Whereas,

k_o = Direct labor cost including overheads($/Min).

$V_r f_r$ = Cutting speed and feed during rough turning.

$V_s f_s$ = Cutting speed and feed during finished turning.

n = Number of rough cuts, an integer.

2.2.2 Tool Cost

From Taylor's tool life equation, life of the tool is predicted as:

$$\mathrm{T} = \frac{C_0}{V^p f^q d^r} \tag{2.5}$$

The same tool is generally used for roughing and finishing but its wear is different in both cases, just by inserting value of feed and speed these parameters can be calculated.. Tool replacement cost is calculated as:

$$C_T = k_t \frac{t_m}{T} \tag{2.6}$$

t_m = Cutting time.

k_t = Cutting edge cost ($/Edge).

2.2.3 Material Removal Rate

In a cutting process, material removal rate is a key characteristic as it is an important parameter to define the productivity of the process. It is dependent on different cutting parameters such as feed, speed, and depth of cut. It is limited by the maximum force bearing capacity of the tool as well as the surface finish of the machined part. It is advisable to have the material removal rate as high as possible. It is expressed as the mathematical equation:

$$MRR = f * V * DOC \tag{2.7}$$

f = Feed rate (mm/rev)

V = Cutting Speed (m/min)

DOC = Depth of Cut (mm)

The optimization problem would be to maximize MRR [51]. This was used to find out optimum values of cutting parameters in order to find out maximum MRR.

2.2.4 Surface Roughness

However, not every parameter or outcome can be described analytically, in which cases the evolutionary Neural Network technique is used for predicting the output. It has been observed that actual surface roughness measured from experimental data does not match theoretical values calculated from existing analytical formulation. This is because of characteristics such as adhesion, ploughing and geometrical effects which cannot be easily modeled using analytical methods [51].

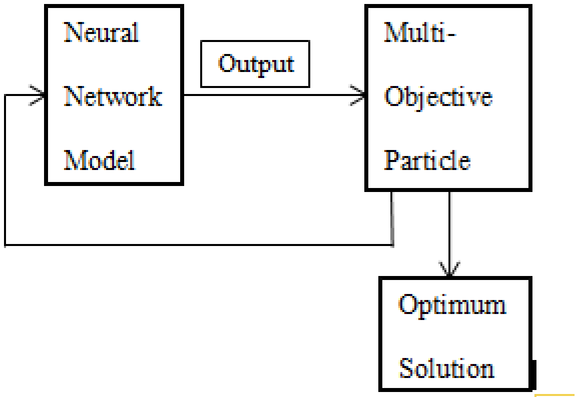


Figure 2.1 Swarm Intelligence Neural Network System [51]

Inputs provided to the neural network are feed, speed, and depth of cut while the output is surface roughness. A Neural network with sufficient accuracy integrated with multi-objective particle swarm optimizer will provide useful information to the user during selection of machining parameters. Karpat and Ozel used this Swarm-intelligence Neural Network (SINNS) to minimize surface roughness [51]. Data collection dictates the degree of accuracy of the results [36]. One of the most effective methods in experimental planning is Design of Experiments (DoE). It is used by the researchers in order to conduct the fractional factorial experiments that collect statistically significant data with minimum possible repetitions. In order to understand the contribution of the process parameters and other factors affecting the resultant characteristic of interest by researchers, variance analysis is carried out using ANOVA. Besides finding the optimum cutting parameters this can be used to find out parameters affecting the process under scrutiny [40]. Researchers (e.g. Tzeng and Yang [34]; Hascalik and Caydas [37]; Dagwa [39] ; Yang and Tarng [40]) concluded that feed was the most dominant parameter, followed by speed, and then depth on the turning process.

2.3 Design of Experiments

Experiments are carried out in order to have better understanding of the process, effects of process parameters on the final outcome or on some phenomenon. In any case many results can be drawn from a large extent depending on the manner in which the data is collected. For efficient and effective performance, a scientific approach towards planning is recommended. Design of Experiment results in systematic process planning, so that appropriate data is collected which is then analyzed by statistical methods so that a valid conclusion is obtained. DoE is an objective approach for selecting and analyzing the data with possible experimental errors. This technique is capable of studying several process parameters in one trial which reduces the work load that would be done for each parameter at one time. The minimum number of trials required increases with the increase in factors under study.

During the experiments conducted by Hascalik and Caydas [37] for optimizing the process parameters of the turning process, a 244% improvement in surface roughness was observed and about a 335% improvement for tool life was observed using the optimum parameters. Using the Taguchi DoE, optimum parameters were found out for the turning operation and accuracy of prediction for surface roughness was observed to be up to 96.44%, for experiments conducted by Dagwa [39]. Use of DOE was also extended up to drilling operations for improvement of surface roughness by Zhang and Chen [38] when satisfactory results were obtained. Yang and Tarng [40] discovered for their experimental set up and resultant surface roughness with optimum parameters was improved by 250% compared to that of the initial parameters.

3. EXPERIMENTATION

3.1 Experimental Set up

3.1.1 Introduction

The information is provided in this chapter is about experimental set up used for force acquisition. This setup is used for acquiring the cutting forces and the surface roughness generated during turning process of metal work piece for a wide range machining conditions. The cutting forces and surface roughness obtained after the experiments from this set up are used as resource to create simulation of turning process by using artificial neural network (ANN) technique. The acquired data is used for analysis of the process and to find out the cutting parameters which results in most efficient outcomes. It can also be used for further analytical purposes such as tool and material comparison, sensory input for adaptive process control etc. Before we proceed to describe the intricacies of the setup, a list of the various components used in this experimental are listed below.

Table 3.1 Turning force and surface roughness measurement experimental setup components

Sr. No.	Component	Specifications
1	3-Component Force Dynamometer:	Kistler Type 9257B
2	Shielded Connecting Cable	Kistler Type 1687B5
3	Charge Amplifier	Kistler Type 5010
4	CB-68LP	National Instruments
5	Type SH6868 Shielded Cable Assembly	National Instruments
6	NI PCI-6036E	National Instruments
7	IBM PC, sufficient capacity or 100 % compatible for hardware interfacing.	
8	Matlab GUI	MATLAB
9	TR100 Surface Roughness tester	

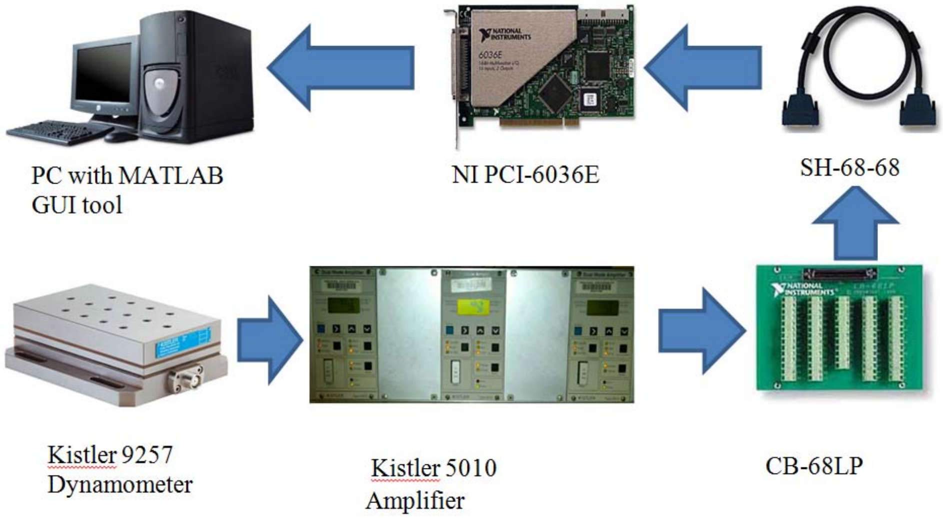


Figure 3.1 Force acquisition experimental setup schematic

3.1.2 Part Descriptions

The experimental set up begins with one of the most important part,viz. the three-component force dynamometer (Item 1, Table 1) by Kistler$^{□}$, Type 9257B. This dynamometer is quartz based, and is used for measuring three orthogonal components of force as referred in articles [47, 48]. It has great rigidity, high natural frequency, and, high resolution. High resolution is preferred as it records very small dynamic changes.

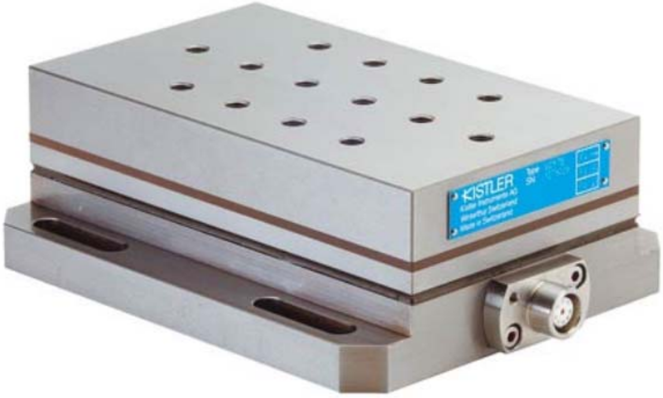


Figure 3.2 Kistler Type 9257B

The quartz based dynamometer operates as follows. The electrical signals are generated from the forces acting on quartz element. These generated signals are in proportion with the forces. The resulting displacements due to these forces are very small. Rigidity coefficients are C_x,C_y > 1 kN/□m and C_z > 2kN/□m. Therefore force components are measured without any major displacement. Sensitivity is 7.95 pC/N (pC = pico-Coulomb, a unit of electric charge, N = Newton, a unit of force) for X- and Y-axes and 3.74pC/N for Z- axis, which is very high. Between the base plate and the top plate four, three component force sensors are fitted with high pre-load. Each sensor has three pairs of quartz plates, two for responding to pressure in X and Y direction and one for responding pressure in Z direction. In order to allow the multicomponent measurements of forces and moments, the outputs of these four sensors are connected inside the dynamometer.

The electric signals from dynamometer are sent to the charge amplifier. It is an electronic unit. It is used to convert a charge from piezoelectric transducer to low impedance output voltage, utilizing the high-gain voltage amplifier with negative capacitative feedback. It is necessary to shield the charge induced by the dynamometer due to its very small magnitude (few pC/Newton force). The reason for using specialized cable i.e. Type SH6868 Shielded Cable Assembly is that, there is concern that it could be exposed to external electromagnetic interference.

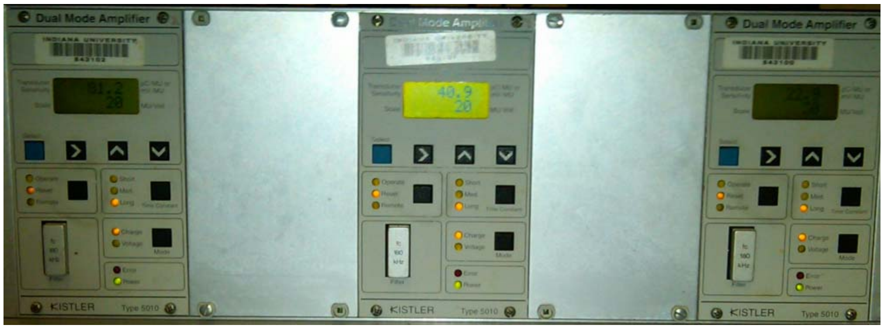


Figure 3.3 Kistler 5010 Amplifier

After an amplifier does its work, charge from the dynamometer is converted to voltage, which is in measurable data. For conversion from an analogue format to a digital format, National instruments CB-68LP connection block is used. It allows simple connection to NI acquisition cards through ribbon cable. This data is then stored for analytical purposes.

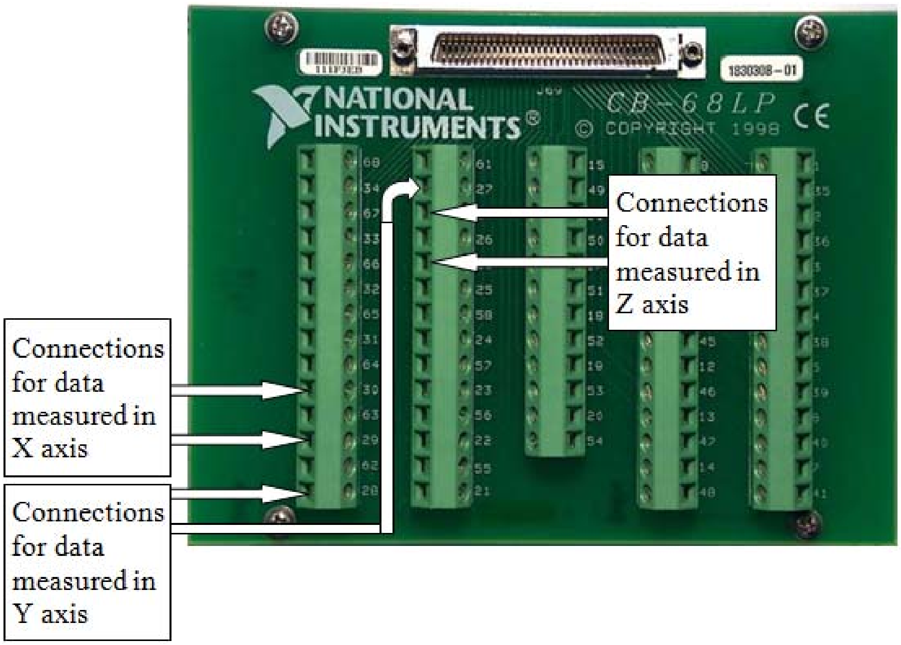

NATIONAL INSTRUMENTS
CB-68LP
© COPYRIGHT 1998
Connections for data measured in Z axis
Connections for data measured in X axis
Connections for data measured in Y axis

Figure 3.4 CB-68LP connection block for Analogue to digital conversion

From the CB-68LP connection block, a SH68-68 Shielded cable is attached to NI PCI-6036E. It is a multifunction data acquisition cart. NI PCI-6036E has two 16-bit analogue outputs; eight digital I/O lines; two 24-bit counters. This was chosen due to availability of resources.

Figure 3.5 NI PCI-6036 multifunction data acquisition cart

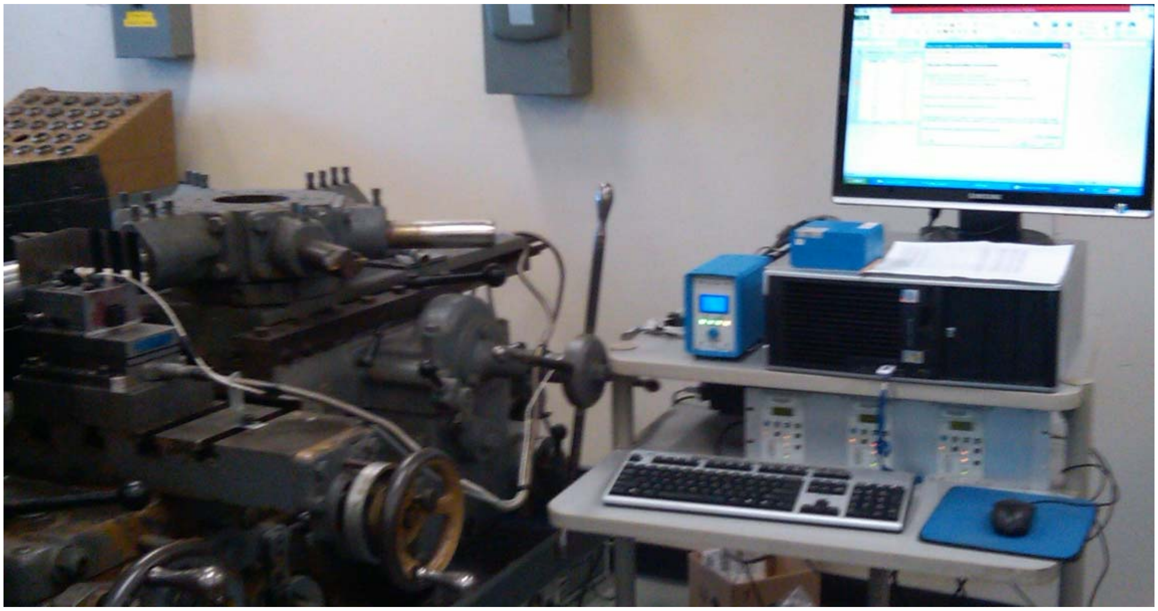


Figure 3.6 Complete set up of force acquisition system

3.1.3 Data Acquisition Concepts

Necessary characteristics of the process to acquire the data are described below [49].

- Sampling rate

It defines the number of samples per unit time acquired from continuous signal to make discrete signal. In order to have properly digitized signal, sampling rate is required to be twice the rate of that of maximum frequency component of the signal as per Nyquist sampling theorem. Lower sampling rates may lead to incomplete recovery of information from the sampled signal. When this requirement is met problems such as aliasing are avoided.

- Resolution

Number of bits (binary) used for representation. This determines the accuracy /least count of acquisition. For example, 12-bit means 2^{12} = 4096 parts of the range. This for a 10V range the resolution is 0.0025V.This translates into an accuracy of 0.25 N when the scaling factor on the charge amplifier is set to 100 Newton/Volt (i.e. measuring forces in the range of ±1000N)

- Multiplexing

It is a method by which multiple analog message signals or digital data streams are combined into one signal using same ADC chip. In the experimental set up 3 force components are measured. Board however is capable of measuring sixteen such single ended channels.

- Range

This dictates the maximum and the minimum values of the voltage level ADC can quantisize. However there is limit to which the hardware is exposed to the voltage (in this case ±10V). Signal conditioning does allow to control the maximum voltage exposure. Scaling factor is used at the charged amplifier to avoid overstepping of the voltage signal at the acquisition card.

- Over-voltage protection

This feature provides the protection to the experimental set up for unexpected outcomes such as tool breakage or too large cutting force due to excessive depth of cut or even collision of parts. The safety factor is deployed to safeguard the equipment in case of any accident.

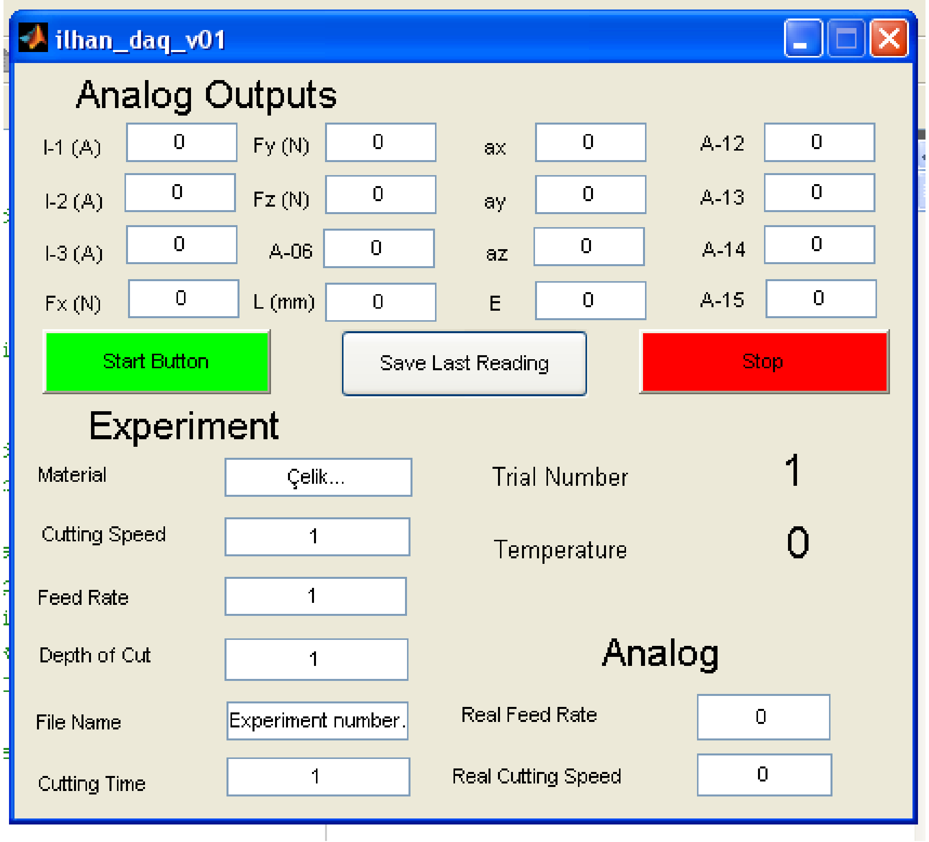
- Data Transfers

Since a large amount of data is handled, this characteristic holds lot of importance. This technique facilitates data transfer without any problems. It includes Direct Memory Access (DMA), and programmed I/O.

3.1.4 MATLAB GUI Implementation

MATLAB GUI (Guided User Interface) is tool from MATLAB. It is used as application software platform [50]. MATLAB GUI is chosen because of its availability. Also, since the hardware used during the experimental set up is from NI and compatibility of MATLAB GUI with NI hardware has been proved to be excellent. The rate at which the readings are taken is too high. In order to have ease of analysis, data from the three separate channels is stored and documented in the spreadsheet format. By doing this, the objective of acquiring and storage of the data in usable format is achieved. The technique used for data acquisition is Circular buffer. It is a data structure where single, fixed-size buffer is used to connect the ends. Buffering data streams is done easily in this structure.

Once the program is stopped after pushing the stop button, a text file is created with name that was the respective experimental number designated at the start of experiment. Acquired input is converted into string, and this data is stored in a line at that time and then this is continued until the operation is stopped. The benefit of having this is that, there is no error in data logging. For every iteration, the converted data is stored in spreadsheet string of the file. When the STOP button is pressed it forces the program to stop data acquisition loop and close the file.



ilhan_daq_v01

Analog Outputs

I-1 (A) 0 Fy (N) 0 ax 0 A-12 0
I-2 (A) 0 Fz (N) 0 ay 0 A-13 0
I-3 (A) 0 A-06 0 az 0 A-14 0
Fx (N) 0 L (mm) 0 E 0 A-15 0

Experiment

Material Çelik...
Cutting Speed 1
Feed Rate 1
Depth of Cut 1
File Name Experiment number.
Cutting Time 1

Trial Number 1
Temperature 0

Analog

Real Feed Rate 0
Real Cutting Speed 0

Figure 3.7 GUI Interface for experimental setup

3.1.5 Data Processing

The data recorded is then filtered to avoid time delay errors during operations. The actual force value is taken as the average of forces recorded from the iterations during that time interval. Same procedure is followed for the forces obtained from other two channels. This ensures the accurate collection of data and reduces the possibility of errors. Similarly while measuring the surface roughness, surface roughness tester TR100 is positioned along the axis passing through the center of workpiece. This procedure is carried out three times at 120 degrees interval and final value is recorded from the average of the three reading.

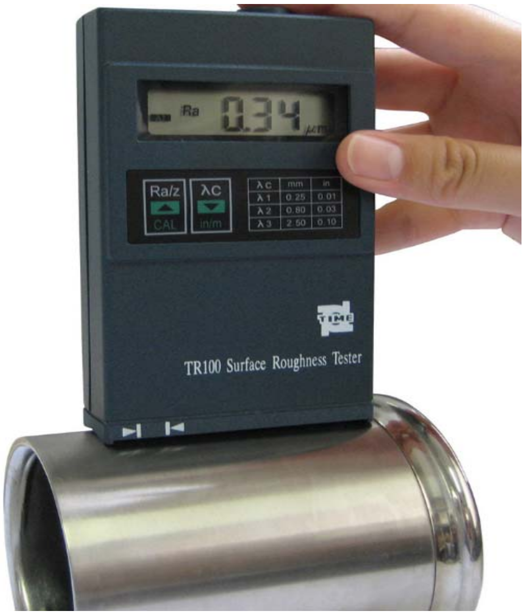


Figure3.8 TR100 Surface tester for measurement of surface roughness

3.1.6 Calibration of Setup

In order to ensure the accuracy of the acquired data, it is necessary to perform validation and benchmarking tests. It was performed on the experimental setup used for data acquisition. For testing, trial and error method was used to validate the accuracy of setup for reading forces acting on tool. A USB S-block load cell was attached along the dynamometer for testing. The load cell displays the force acting on the dynamometer, which is then compared to the force read by the setup. The setup specifications are modified if necessary until the readings match with readings on lad cell display. This was carried out for the three directions to ensure that forces being measured in X, Y and Z direction are accurate. This calibration was not only performed at the start of trials, but also in between trials to ensure the accuracy. This testing and calibration can be seen in Figure below.

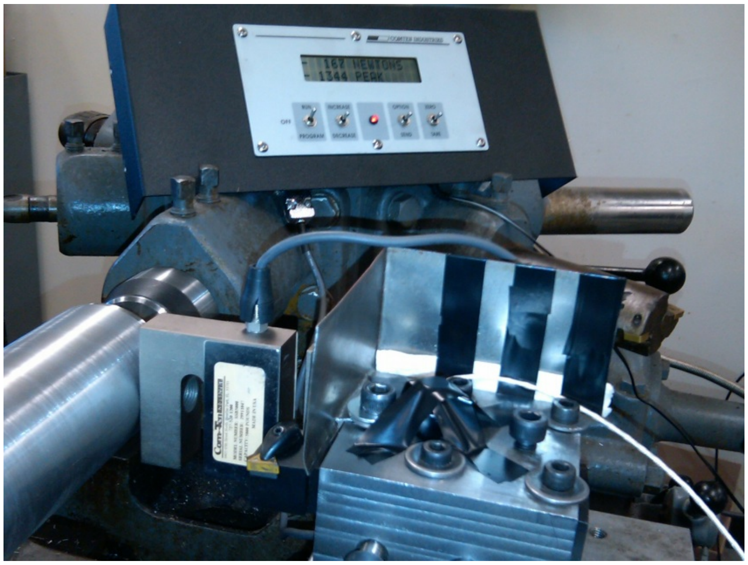


Figure3.9 calibration process of the setup for force measurement

During use of the TR100 surface roughness tester, it is also calibrated time after time from the provided apparatus. This is done to ensure the accuracy of surface roughness measurement readings.

3.2 Experiments

This chapter mainly describes how experiments were carried out. Experimentation was done to obtain the data required to develop the neural network. Tests were carried out on AISI 4140 steel. 12 speed Jones and Lamson Lathe model was used for turning operation. The specimen with a diameter of 60mm, 500mm length and hardened 35 HRC is used. The tool used for this is one that is most commonly used for turning process DTGNR 163 C 0° Lead Angle 60° Triangle insert. It is product of Kennametal.

There are many process parameters that affect the performance of the turning process. Out of these parameters, spindle speed, feed and depth of cut have been varied in different set of combinations. Its effect on surface roughness and cutting process is recorded.

3.2.1 Statistical Design of Experiments

In order to limit the cost of experimentation, statistical design of experiments technique is used. It helps to determine optimum number of experiments needed to model the process. It is the link between the statistical design and engineering applications. Design of experiments approach is used by researchers for selection of optimum parameters [34, 35]. The objective of this method is to determine a less expensive way to provide sufficient information for simulating the turning operation. Any process can be categorized into three main aspects from process improvement or control i.e. Control factors , noise factors and system response. Control factors are the ones which dominate the performance of the process. Noise factors are the parameters that are difficult or rather expensive to control. Control factors for turning process are identified by the effect they have on system response, which in this case is represented by surface roughness and cutting forces. Tool geometry, workpiece material, tool material are among the other control factors. Considering the process limitations, some of these factors are more economical to vary. Characteristics such as Spindle speed, feed rate, and depth of cut, which have most effect on turning process and are easier to vary. These parameters are defined as principal control factors.

The experiments are carried out for different combinations of the chosen parameters. These parameters are as follows.

1. Spindle speed.
2. Feed rate.
3. Depth of cut.

Once these parameters are identified, next step is to select the operating values for them. The ranges of the values selected are recommendations from material data handbook, to avoid the variance due to tool wear. The objective of this study is to choose the most economical parameters without compromising the accuracy of the model of the turning operation. The number of levels of the parameter values directly affects the experimentation cost.

- Spindle speed

This parameter signifies the speed with which the spindle of the machine. The direction of the rotation is anticlockwise. It is also referred as cutting velocity υ. It is measured in terms of meters per minute (m/min).

Four values over possible range of selection are selected are as follows

S1= 433 rpm (corresponds to □ = 104 m/min)

S2= 622 rpm (corresponds to □ = 149 m/min)

S3= 881 rpm (corresponds to □ = 211 m/min)

S4= 1264 rpm (corresponds to □ = 305 m/min)

- Feed rate

It is the rate at which the tool will advance for every revolution of the workpiece. The feed rate is determined by the size of the chip that the tool can withstand. It is measured in terms of mm/revolutions.

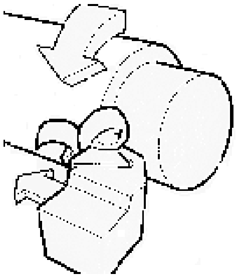


Figure 3.10 Feed direction of the tool

F1 = 0.127 mm per revolution.

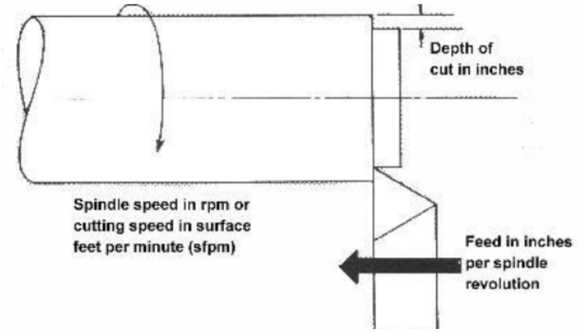
F2 = 0.1778 mm per revolution.

F3 = 0.381 mm per revolution.

F4 = 0.5588 mm per revolution.

- Depth of Cut

It is the radial distance by which the tool is immersed into the workpiece. It is measured in terms of millimeter.


Depth of cut in inches
Spindle speed in rpm or cutting speed in surface feet per minute (sfpm)
Feed in inches per spindle revolution

“Figure3.11 pictorial representation of depth of cut.”

D1 = 0.254 mm

D2 = 0.762 mm

D3 = 1.27 mm

D4 = 1.778 mm

Orthogonal Arrays (OA) are utilized so as to reduce the number of experiments. Orthogonal Arrays are equally spaced intervalsof control factor values. It is used so as to obtain result as that of the detailed experimental trials.

3.2.2 Set of Experiments

As stated earlier, the spindle speed, feed rate and the depth of cut are selected as principal control factors for this study. During the experiment trials, values of these parameters are varied and cutting force and surface roughness values are recorded for each trial. Selected parameters are very crucial in order to simulate the accurate model of the turning process. Since four levels of parameter values are selected, there will be 4^3 =

64 experiments. Range of the values selected for the spindle speed, feed rate and depth of cut are based on recommendation made by Walsh [31]. Equally spaced levels are used as shown in following table.

Table 3.2 Indicating levels for different values of turning parameters

Parameter	Level 1	Level 2	Level 3	Level 4
Speed ‘v’(m/min)	104	149	211	305
Feed rate ‘f’ (mm/rev)	0.127	0.1778	0.381	0.5588
Depth of cut ‘a’ (mm)	0.254	0.762	1.27	1.778

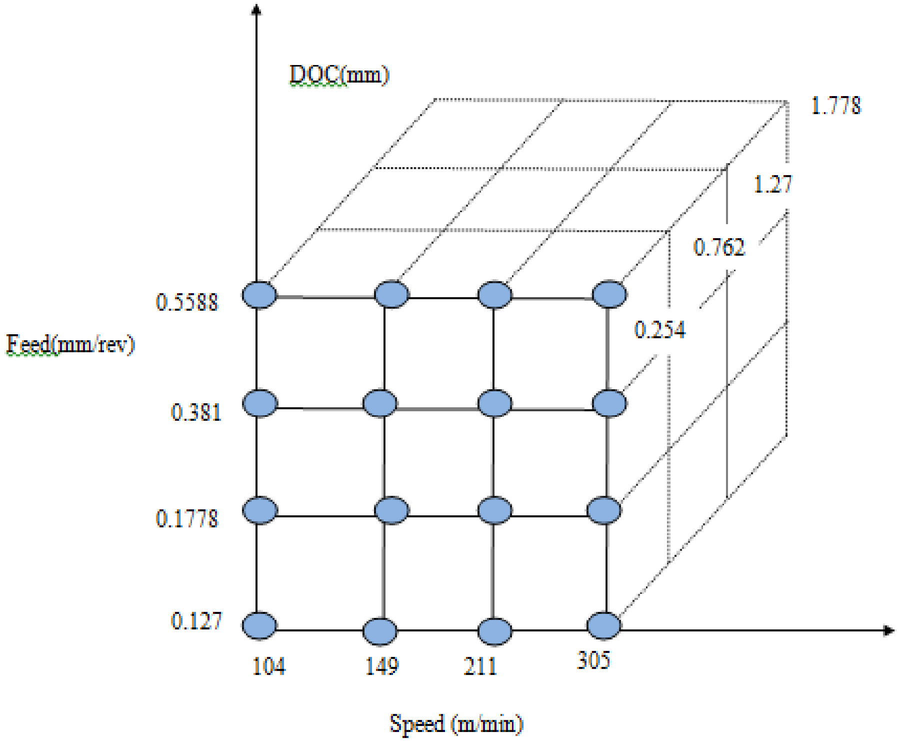


Figure 3.12 Pictorial representation set of experiments

3.2.3 Noise Factors

Noise factors are the parameters that affect the performance of the turning process. Elements such as tool wear, vibrations of the machine-external vibration, material defects, ambiant temprature effects, white noise (due to electronic devices that are part of the set up) are considered among noise factors. Sometimes with same input parameters the output is oberved to be different if the noise factors are dominant. A few ways to reduce effect of such factors are, using a coolant, using vibration absorbers, using recommended input values. The use of coolant reduces effect of cutting temprature where as using recommended values reduce the tool wear.

A study was conducted to determine which of the control factors influence the surface roughness and the cutting forces the most. During this study, the output values of surface roughness and cutting forces are plotted for corresponding of the control factors values. For every trial plotted on the graph, two of the three control factors are kept varying and third is kept constant. Levels of the two variable control factors are indicated in the legend provided along with each graph. It is clear from the Figure 3.15 and 3.16 that feed rate influences the the surface roughness and cutting forces more, than the spindle speed and the depth of cut.

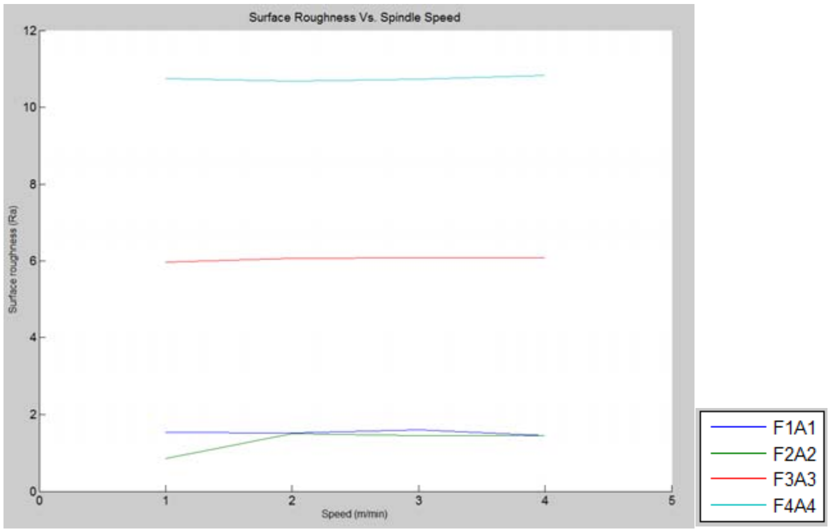


Figure 3.13 Graphical representation of spindle speed effect on surface roughness

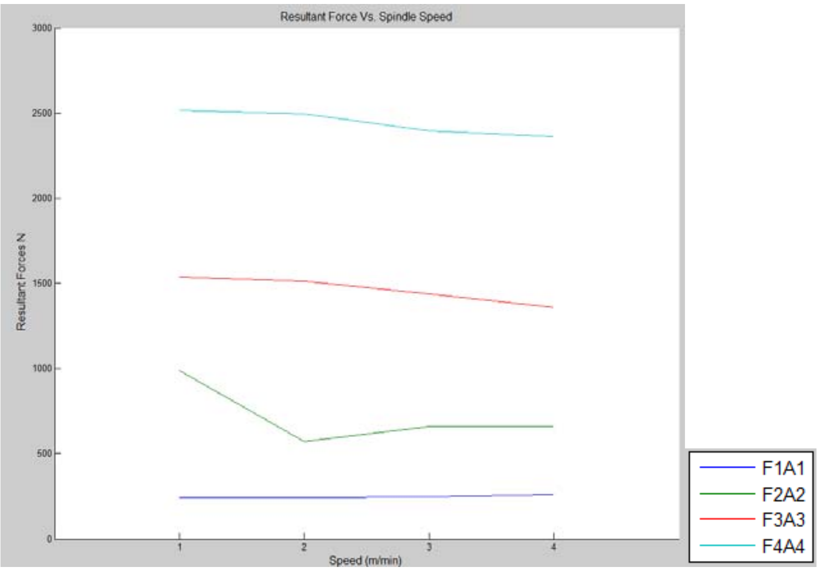


Figure 3.14 Graphical representation of spindle speed effect on cutting forces

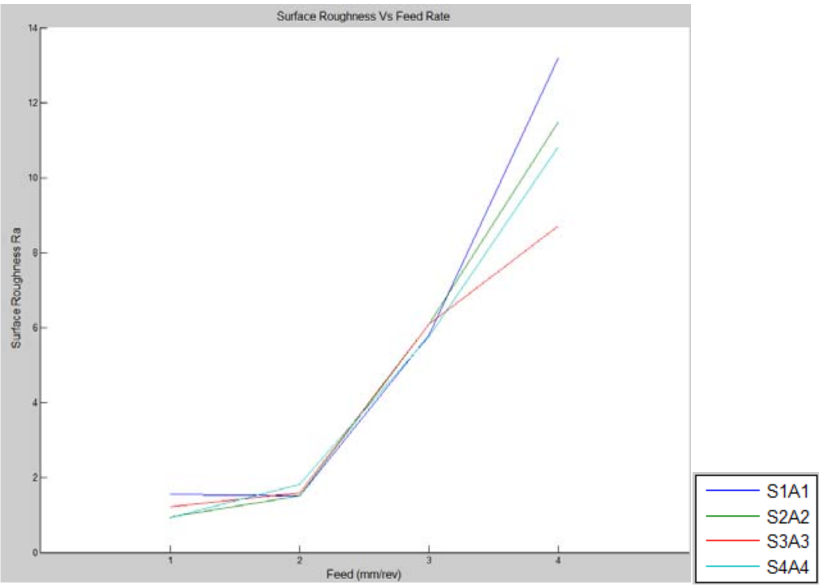


Figure 3.15 Graphical representation of feed rate effect on surface roughness

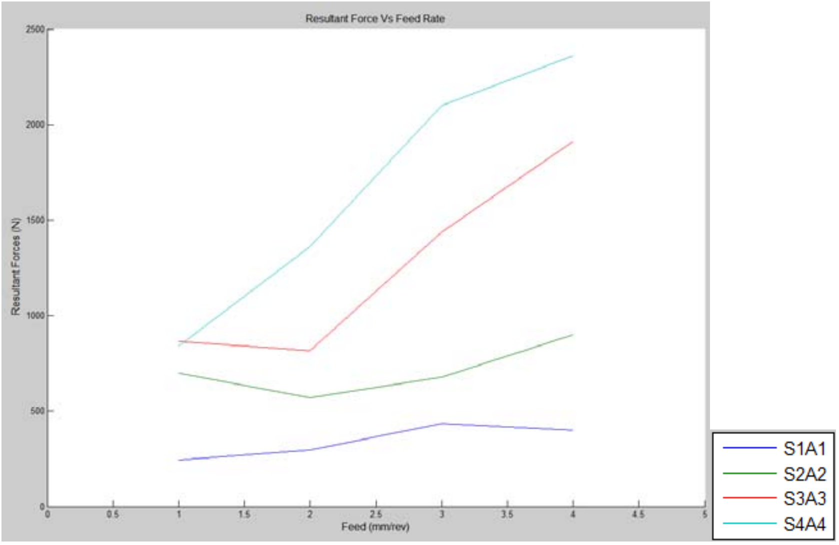


Figure 3.16 Graphical representation of feed rate effect on cutting forces

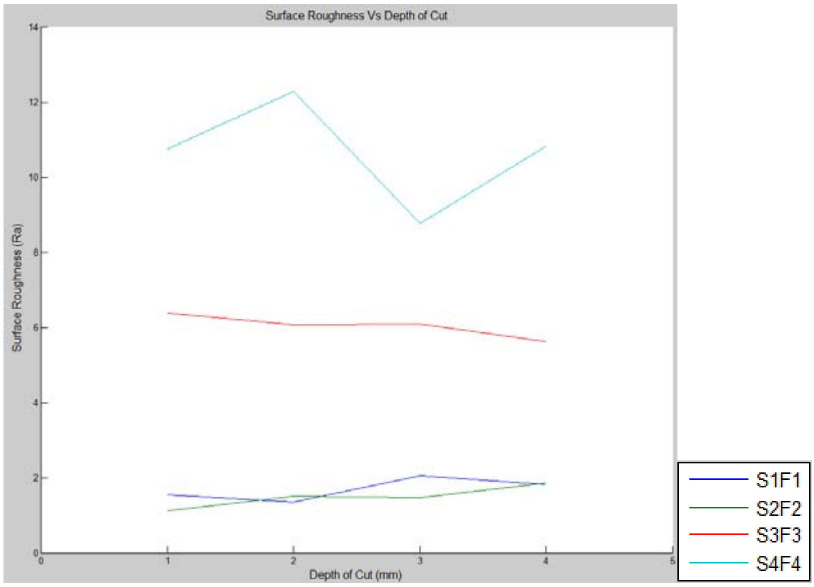


Figure3.17 Graphical representation of depth of cut effect on surface roughness

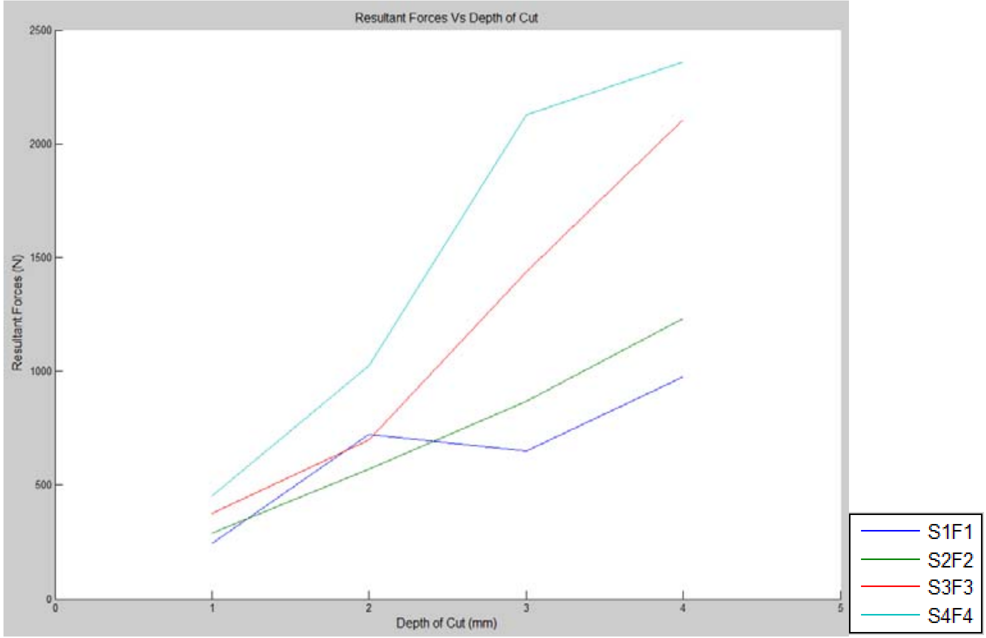


Figure3.18 Graphical representation of depth of cut effect on cutting forces

3.2.4 Data Analysis

3.2.4.1 Forces Calculations

Once the trials are over, the data is stored in the excel sheets. Forces are recorded in X, Y and Z direction. Resultant force R is calculated using the following equation:

$$Fr = \sqrt{Fx^2 + Fy^2 + Fz^2} \tag{3.1}$$

This “Fr” is vector quantity, which represents the forces recorded during each trial. As shown in equation 3.1. “Fr” takes into consideration, forces in all the three directions. It is very important to consider the resultant force. The workpiece material always has some surface irregularities such as hard points or vacancies (depression). When tool comes across such irregularities, cutting forces displays sudden jump or drop in recorded values. In spite of these patterns, the resultant force captures the real characteristics.

4. ANN MODEL OF TURNING

4.1 Introduction

Simulation of a turning process is one of the important steps during this experimentation, without which optimization will not be feasible. Simulation should be modeled taking into account all the relevant process control factors. It should have reasonable accuracy. As seen in literature review, many processes are modeled mathematically. Unfortunately modeling of turning process using mathematical function to obtain surface roughness is a bit difficult task. Many researchers adopted this technique to predict similar outcome [17-20, 24-30]. Relation between inputs or control factors and surface roughness, forces is non-linear. We used Neural Network approach to model the process. Artificial Neural Network uses data obtained from experimentation in order to predict the outcomes.. This also helps to attain reliable results as there are no assumptions on functional relationship such as quadratic, cubic etc.

4.2 Artificial Neural Networks

Use of ANNs can be traced back to many applications in a wide spectrum of fields, especially for applications which are difficult to generate mathematically. In the manufacturing field, besides turning operations, experiments were carried out to predict outcomes of milling, drilling machine etc. Geological, construction, analysis based applications have also adopted ANN approach. For output prediction within the metal cutting applications, ANN is utilized to predict different parameters that are rather difficult to evaluate using mathematical/analytical method e.g. cutting forces, surface roughness, tool path etc. Before discussing details of application of ANN technique for simulation of turning, we need to discuss details of ANN technique. Neural networks consist of processing elements which are also known as neurons. Processing elements act as a middleman between accepting input information and producing output information. Mathematically, mapping of input vectors is carried out on output vectors. The reason of using ANN to simulate a model is that, the model can be claimed to generic and complete so as to predict fairly accurate results.

4.2.1 Elements of Artificial Neural Networks

ANN consists of following things:

1. Processing Elements (PE) or Neurons.
2. Connection weights.

Combining these two elements will result into topology of the Neural Network.

4.2.1.1 Processing Elements/Neurons

Each Processing Element performs simple computations. In case of an Input P.E., it receives input and produces output value. Value of output is dependent on the input value supplied to the P.E. Computation can be divided into two types. The first component is linear component is called input function In_i and second component which is non-linear is called activation function g. For other types of P.E. all inputs are combined before the P.E.:

$$in_i = \sum_j W_{j,i} * a_j = W_i * a_i \tag{4.1}$$

Computation step where a P.E. calculates the activation function, g, as a result of input function value:

$$a_i \longleftarrow g(in_i) = g\left(\sum_j W_{j,i} \ a_j\right) \tag{4.2}$$

a_i Activation value of unit I (also output of the unit)

$W_{i,j}$ Weight on the link from unit j to unit i

W_i Weight from unit I to the output in perceptron

g Activation function

Following are the important qualities of Processing Elements. These qualities facilitate parallel operation of neural network.

1. Only local information is required. Rest of the required information is within the P.E. to produce output from the available input. No other information is required by P.E.
2. Output value is generated, and provided to following P.E.s or to the output of the network.

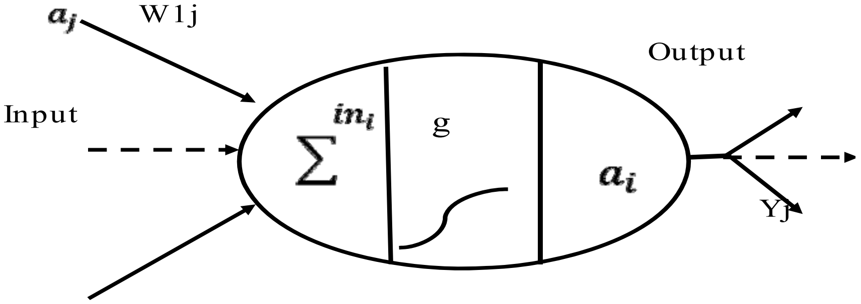


Figure 4.1 Processing element architecture

4.2.1.2 Network Connection Weights

Weight is the numeric value assigned to connection between Processing Elements. Weights are responsible for the storage of the changes that the neural network goes through while training. Learning of the network usually takes place when weights are updated considering the predicted and actual values. Biases are special case of network weight in general.

4.2.2 Training Methods

The aim of training the neural network is to make it more evolved and have strong relation with Inputs and outputs. We have selected Back Propagation as training method for NN model. Following are the reasons to justify the selection.

1. Standard Back Propagation is a gradient descent algorithm. Name Back propagation refers to the way in which gradient is calculated for non-linear multi-layer Neural Network.
2. It provides fairly accurate results for the Input data that it has not been exposed to. If a new input is presented, it will lead to an output which is fairly accurate for input vectors used in training that are similar to new input being presented.
3. The order in which inputs are presented to the NN does not affect accuracy of the training procedure. This is achieved because an adaptation is being done only at the end of each epoch.

Second reason mentioned above, is key reason why we chose this training method. We have discussed the design of experiment study, where we selected particular step values of machining parameters in order to achieve low cost of experimentation.

4.3 Back Propagation Algorithm

Back-propagation is the most popular method of learning in multilayer networks. If there is a difference between the predicted value and an actual value, which is referred as error, network connection weights (i.e. one of the main constituents of the network) are adjusted so as to reduce the error. However, the important thing is to narrow down the weight that needs to be adjusted in order to improve the result. It is also important to understand that, values are provided for training as well as for comparison of the network. In another way the network weight is responsible for error. Back propagation algorithm has sensible approach to divide and identify the contribution of individual weight. Error is defined as:

$$E_{rri} = T_i - O_i \tag{4.3}$$

E_{rri} Error i.e. difference between target and output

T_i It is the target value provided to the network

O_i It is the output value calculated by the network.

As mentioned earlier, when there is an error, connection weights are adjusted. E.g. the weight update rule for connection weight from I to j is calculated as:

$$W_{j,i} \longleftarrow W_{j,i} + \alpha * a_j * E_{rri} * g' (in_i) \tag{4.4}$$

α Is it the learning rate of the network.

g' Is derivative of the function g.

a_j Is the hidden processing elements.

For the simplicity of the calculations the equations can be rewritten as:

$$W_{j,i} \longleftarrow W_{j,i} + \alpha * a_j * \alpha * \Delta_i \tag{4.5}$$

Hence for output nodes

$$\Delta_i = E_{rri} * g' (in_i) \tag{4.6}$$

Error back propagation is carried out in the following way. Suppose the hidden node j is responsible for part of error or fraction of error Δ_i in output nodes. Since, Δ_i values are dependent on connection weights between hidden nodes and output nodes, this is propagated back to provide Δ_j values for the hidden layer. Hence in this way propagation of the Δ values is carried out to reestablish the weights and attain more accuracy. Propagation can be described in the equation form as follows.

$$\Delta_j = g'(in_j) \sum_i W_{j,i} \Delta_i \qquad (4.7)$$

Δ_j For errors among j nodes, calculated similar to that of node "i "as shown above.

We have discussed the updating procedure of connection weights among the hidden layer and output layer. Procedure for updating the connection weight between hidden layer and input layer is almost identical to that of output layer.

$$W_{k,j} \leftarrow W_{k,j} + \alpha * I_k * \Delta_j \qquad (4.8)$$

I_k Is the Input processing elements.

Work of back propagation algorithm can be summarized as follows.

1. Calculate the error between target value and output value of the network.
2. Calculate the value of Δ from the available values.
3. Propagate the Δ values to the layer before output layer and further.
4. While doing so, keep updating the connection weights between consecutive layers.

4.3.1 Training and Validation

The provided experimental data is divided among three stages i.e. training, testing and validation respectively. This data subdivision is carried out by inbuilt characteristic of matlab neural network tool. During training, the connection weights of the processing elements are updated to predict results. This is done with help of target values supplied by the user. During the learning phase of the network, pairs of input and output are supplied to the network. Inputs are used to create the input vectors and output vectors are calculated which are then compared to output values of the data supplied by the user. If the errors are zero, no learning takes place. Next step is testing, during this stage already trained network's prediction is gauged with actual data and error is calculated. This error is estimation of how well the NN model is predicting the outputs. During this stage we get to understand how well the network has been trained. The network is trained until the errors are to the minimum level. Next stage is validation, during this stage data which is unknown to the network model is presented and NN model is made to predict the outcomes. During this stage we get to know whether use of this NN model is practical or not. All these stages are equally important as each stage indicated performance of NN model at various levels.

4.3.2 Predictive Surface Roughness Model

Selection of the output variables is done carefully depending on the requirements or expectations of the user. For Neural Network model to simulate turning process and estimate the expected relevant variables, input parameters selection is very crucial. The accuracy of prediction of the network could hamper if selected input parameters are not influential with reference to the expected output. The output parameters that we need the NN model to predict are as follows:

1. Cutting forces in Z -Direction.
2. Cutting forces in Y -Direction.
3. Cutting forces in X -Direction
4. Surface roughness R_a.

For optimization purposes, resultant cutting force is considered as a constraint, as excessive resultant cutting force could result into tool deterioration, which would eventually affect the surface roughness. Thus, it would be sufficient just to be able to predict the maximum resultant force accurately.

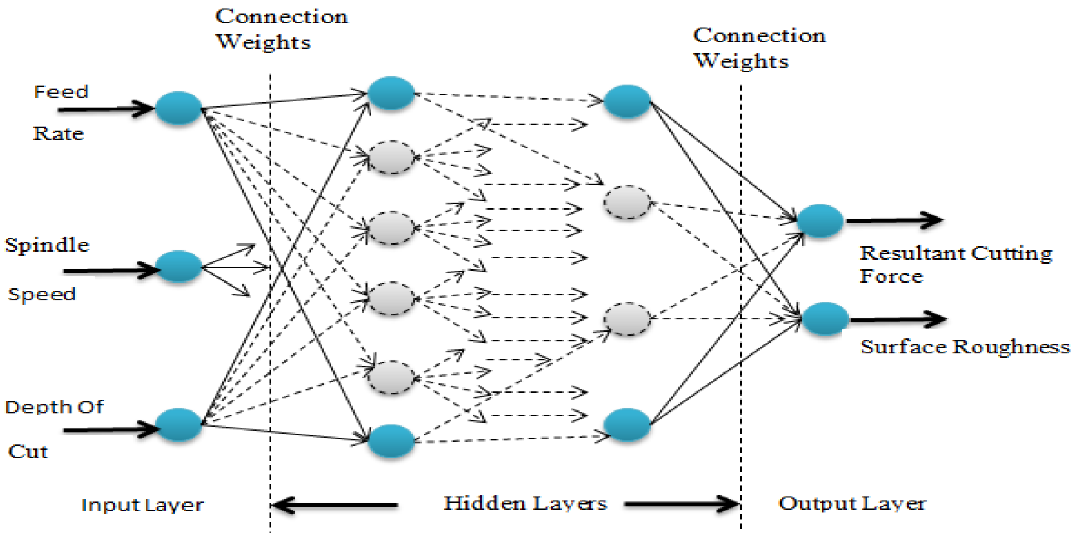


Figure 4.2 Neural network topology

As shown in Figure 4.2, Size of the input layer and the output layer is fixed. The vector length of input vector is 3 components and for the output vector it is 4 components. Hence this indicates the topology of the network to be 3 - X_i – 4. X Represents the number of processing elements and “i” represent number of hidden layers. In order to Figure out the number of processing elements in the hidden layer, we conducted following study. Since we were not sure what should be the number of processing elements, and number of hidden layers of finalized Neural Network model, we conducted a network training of various NN models with different hidden layers and different number of processing element. After comparison of the accuracy for prediction, 3-15-15-4 combination was found to be most accurate. This included two hidden layers with 15 processing elements each. The activation function of the two hidden layers is sigmoid function, whereas output layer has pure linear function. We will be discussing this in detail in section about network topology.

4.3.3 Data Preparation

Training the neural network requires a set of data in input and output format. In order to provide the data in such form it is very important to process it. The way in which data is recorded, dictates how much work needs to be done for processing it. In this case, the data is recorded as force component in X-axis direction, Y-axis direction and Z- axis directions. Processing is done as follows.

To facilitate the training of the NN model to predict the designated outputs, it is very important that the data should be processed in pattern.

Procedure:

Step 1: For each iteration, resultant force F_r is calculated from recorded experimental data. It consists of F_x, F_y and F_z. Hence F_r is calculated as,

$$F = \sqrt{F_x^2 + F_y^2 + F_z^2} \tag{4.9}$$

Step 2: Average of the surface roughness values is recorded from the three measurements taken at 120 degrees apart. These steps are repeated for each experiment. Please refer to following table for sample calculations.

Table 4.1 Sample of the data processing

F_x	F_y	F_z	F_r	$R_{a\_}1$	$R_{a\_}2$	$R_{a\_}3$	$R_{a_}$average
42.89585	104.2802	135.7829	176.4976	0.9	0.0.938	0.907	0.915
47.6057	117.241	133.665	184.06	0.887	0.92	0.908	0.905
53.57264	129.3938	147.9112	203.6922	0.9	1.1	1.06	1.02
63.547	140.682	140.658	208.8404	1.1	0.95	0.965	1.005
49.89166	130.1915	176.8473	225.1976	1.12	1.5	1.355	1.325
53.8106	141.713	166.2751	225.0013	1.5	1.415	1.45	1.455
56.4738	152.892	161.2431	229.2697	1.125	1.512	1.518	1.385
59.34957	164.911	152.5892	232.3822	1.21	1.421	1.464	1.365
40.8008	170.1425	180.5421	251.4133	5.236	7.129	6.73	6.365
28.23982	145.5337	178.2087	231.81	7.142	6.03	5.593	6.255

Step 3: Training/Testing pattern vectors are now formed. Each pattern is formed with an input condition vector,

$$\text{Input Vector} = \begin{bmatrix} Feed\ Rate \\ Spindle\ Speed \\ Depth\ of\ Cut \end{bmatrix}$$

And the corresponding target vector,

$$\text{Target Vector} = \begin{bmatrix} Resultant\ Cutting\ Force \\ Surface\ roughness \end{bmatrix}$$

Step 4: For the NN model to simulate the turning process accurately, it is required that the data used for training is normalized. Normalization of the data brings consistency within the database and reduction of redundant data occurs.

$$X = (\, X_R \text{ - } X_{Min}\,)\ \frac{X_{N\,max} - X_{N\,min}}{X_{max} - X_{min}} + X_{N\,min} \tag{4.10}$$

Whereas,

X = Normalized value of the variable.

X_R = Real value of the variable.

X_{max} = Maximum value of the variable (Upper limit of the parameter boundary).

X_{Min} = Minimum value of the variable (Lower limit of the parameter boundary).

$X_{N\ max}$ = Maximum normalized value of the parameter. ($X_{N\ max}$ < = 1)

$X_{N\ min}$ = Minimum normalized value of the parameter. ($X_{N\ min}$ >= 0.1)

4.4 Network Topology

NN Model is created using Matlab Neural Network toolbox. Matlab tool facilitates ease of simulation and modeling. As discussed earlier, size of input and output vector is decided. Trials are first conducted by randomly selecting number of processing elements. Judgment of the accuracy of prediction is done on the basis of the mean square error at the end of the training. The range of selection of the processing elements was narrowed down carefully, on the basis of result of performance for prediction. Among the range of 10 to 20 neurons, NN model was observed to perform with very good accuracy for prediction. The data used for this study is recorded for one of the five inserts used during the experimentation. Since all the inserts have most of the important characteristics identical, this data could be used for selection of topology of Neural Network. Following table depicts the performance of Neural Network Models ranging between 10-20 processing elements. NN models are created with one hidden layer and varying number of processing elements or neurons. As per White's theorem, one layer with non-linear activation function is enough to map non-linear functional relationship in a fairly accurate way [54].

Table 4.2 Network Comparison for best validation performance

Number of Neurons in hidden layer	Validation Performance	epoch	Reason For Stopping	Network Configuration
15	0.00257	2	Validation Stopped/Max Iterations	3--15--2
16	0.0029	5	Validation Stopped/Max Iterations	3--16--2
14	0.00333	11	Validation Stopped/Max Iterations	3--17--2
17	0.00342	1	Validation Stopped/Max Iterations	3--17--2
13	0.00405	2	Validation Stopped/Max Iterations	3--13--2
18	0.00564	1	Validation Stopped/Max Iterations	3--18--2
12	0.00567	5	Validation Stopped/Max Iterations	3--12--2
19	0.0077	1	Validation Stopped/Max Iterations	3--19--2
20	0.00851	4	Validation Stopped/Max Iterations	3--20--2
11	0.0098	5	Validation Stopped/Max Iterations	3--11-2

The arrangement of the table is done in descending order of the validation performance measure. Hence, first combination i.e. having 15 processing elements is selected. Another reason why the network size was chosen to be limited to such a number was that as the size of network becomes larger, generalization characteristics suffer due to increased number of connections. Also computational expense increases in proportion to the size of the network. In latter part, two hidden layers were introduced. This brought more accuracy in parameter prediction. As limited by the Matlab Neural network tool, the number of neurons for both layers is simultaneously selected.

Table 4.3 Effect of number of hidden layer on NN Model Performance

Neurons in Hidden Layer	Validation Performance	Epoch	Reason for Stopping
15-15	0.00126	0	Max Iterations/Validation Stopped
16-16	0.00154	4	Max Iterations/Validation Stopped
17-17	0.00191	0	Max Iterations/Validation Stopped
14-14	0.00261	4	Max Iterations/Validation Stopped
13-13	0.00266	22	Max Iterations/Validation Stopped

As shown in the table, increasing the number of hidden layers does not significantly increase the performance of the Neural Network Model. Hence network configuration was chosen to be 3-15-15-2. Similar configuration is to be used from now on for predicting the performance of rest of the four types of insert.

4.4.1 Network Testing and Simulation

Once the network structure is finalized and data to be used for functional requirements of the NN model is converted into useful format, training, testing, and validation of the NN model can be started. Matlab NN toolbox has characteristic of dividing the available data into 70 percent for training, 15 percent for testing and another 15 percent for validation of the NN model.

4.4.1.1 Network Testing

Once the network is trained to the level where the predicted results are fairly accurate, testing is carried out to assure that predicted results are in proximity to actual values. Mean Squared Error (M.S.E.) is used to judge the accuracy of the prediction during training, as well as testing and validation. It was also noticed that performance did not necessarily improve even when network error was low. In the following Figure 4.3, you can notice that, roughly after 4 epochs, the training error continued to decrease even when the performance of testing and validation were somewhat stagnant. This can be referred to the effect known as "overfitting".

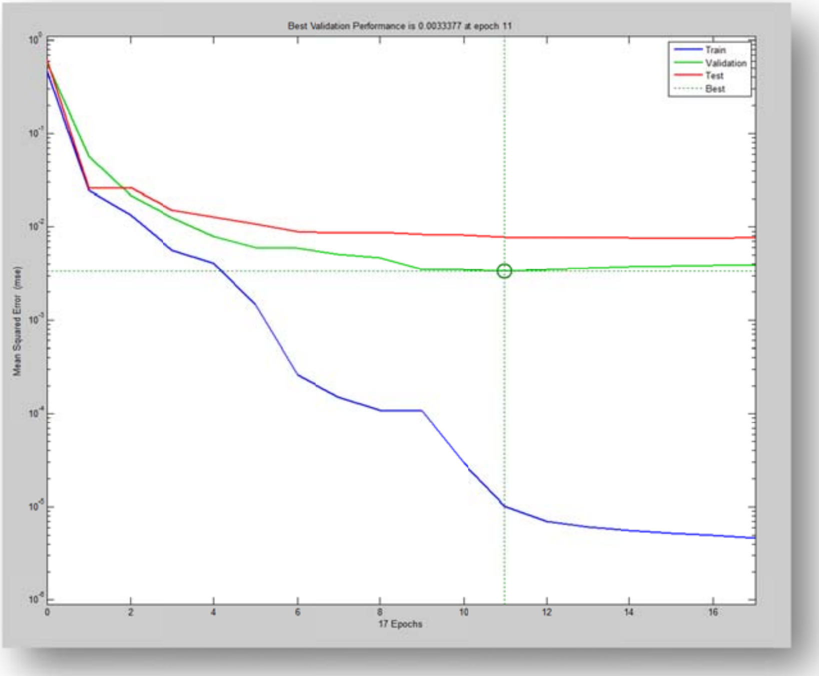


Figure 4.3 Training of NN Model gauged by MSE

From results of the study conducted on network topology, performance of NN model which is most accurate is shown in Figure 4.3. It is our best choice of network topology. With this selection of network topology, numbers of layers, processing elements, generalization characteristics are preserved. It was also noticed that training time is also significantly reduced as there are lesser iterations every time.

Following Figure 4.5 depicts the comparison between actual and simulated data for surface roughness. It is noticed that except for rare occasions, simulated surface roughness values for the designated parameters are in acceptable proximity with actual values. This representation therefore agrees with the conclusion that, high accuracy of prediction is attained by Neural Network model after successful completion of training criteria i.e. with the value of the MSE being within acceptable range as well as agreeable performance measure. Hence from the results it is inferred that performance of the NN model is acceptable.

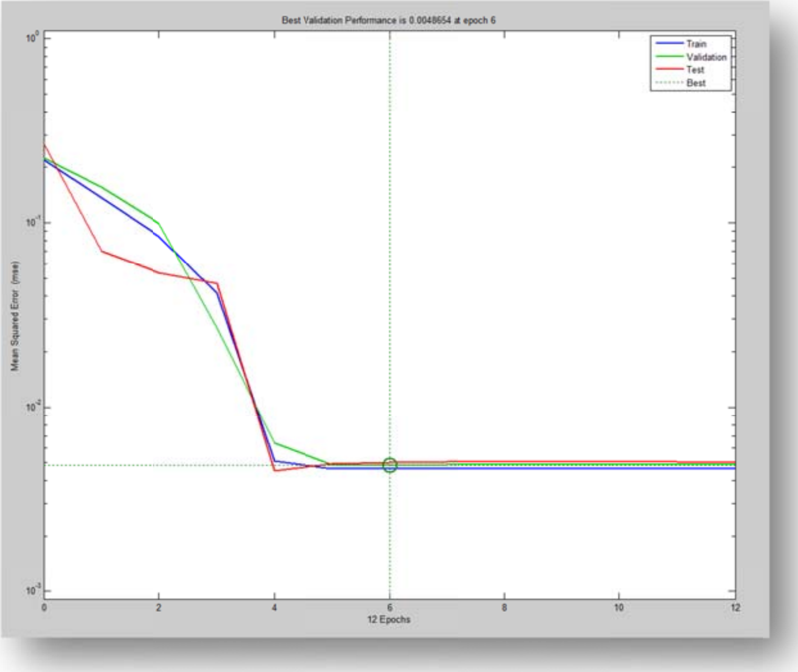


Figure 4.4 Acceptable training performance by selected network topology

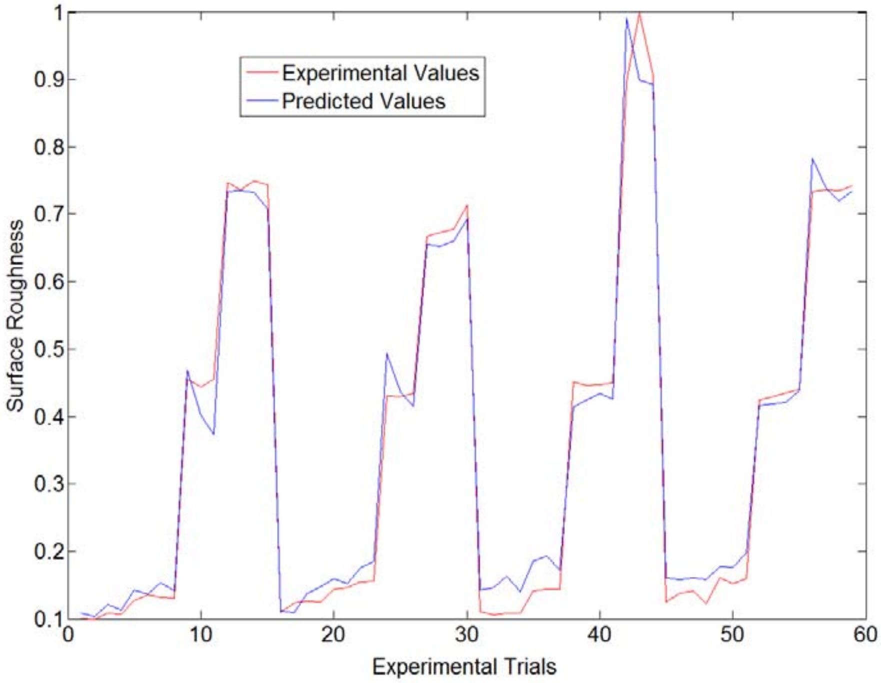


Figure 4.5 Comparison of Actual values and Values predicted by selected Best NN model for Insert FF

Figure 4.6 represents a linear regression analysis between the network response and the network output. It can be inferred that NN model does good mapping. 15 percent of the data which was used for validation was not used for training at all. Hence performance of these machining conditions is something that the Neural Network model has never experienced before. Therefore, we can consider this mapping to be true and representing functional relationship.

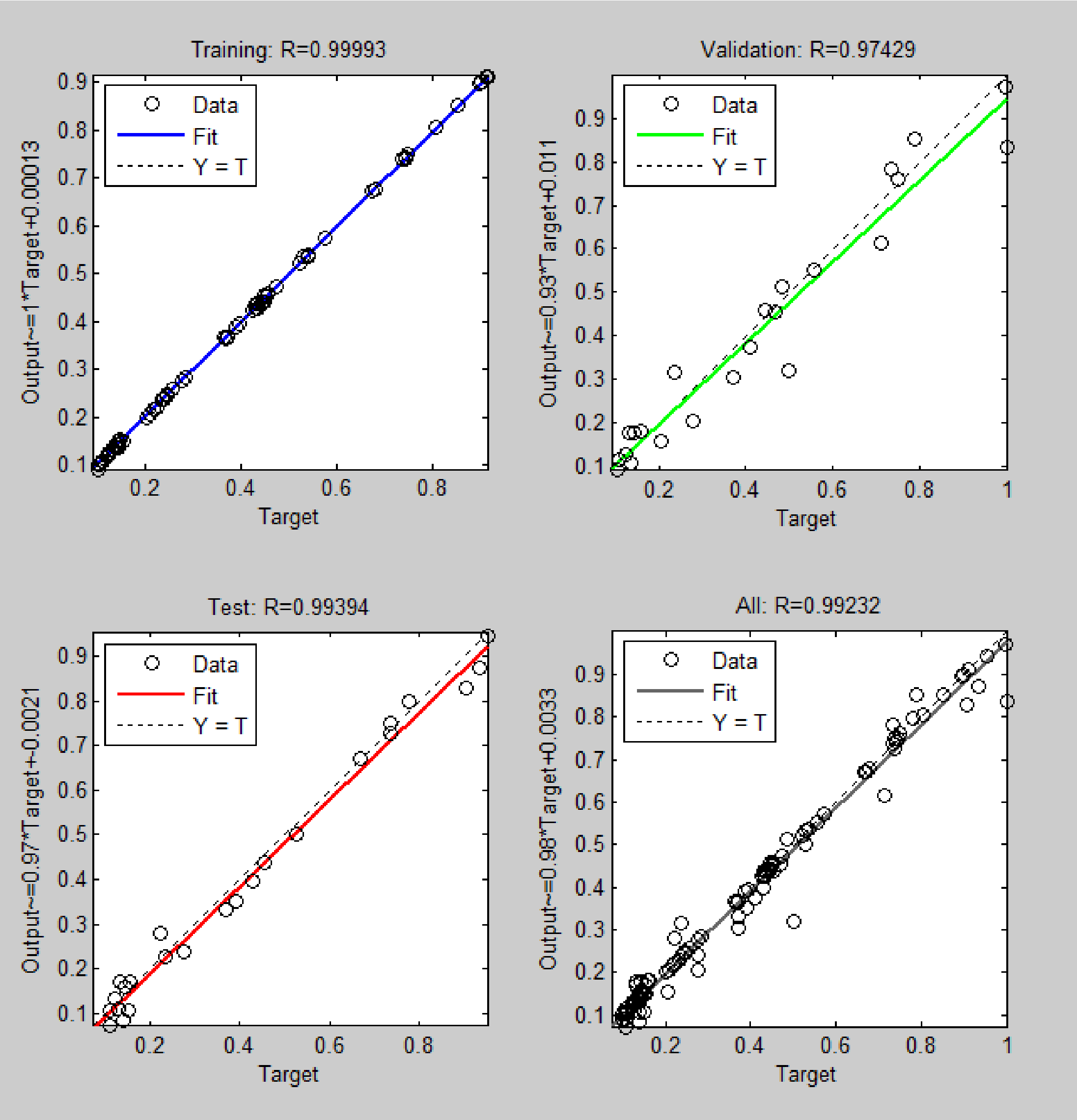


Figure 4.6 Regression Analysis of response for Surface roughness prediction

Similarly, the NN models were created for four other types of insert. Performance and values of R for training, testing and validation are also included in the table so as give a clear understanding. Closer the value of R is to 9, better is the performance.

Table 4.4 Comparison of NN performance created for five different inserts

Insert Type	Validation Performance	R Training	R Testing	R Validation	Avg R
FF	0.00126	0.999	0.997	0.979	0.991667
FN	0.00149	0.998	0.992	0.959	0.983
MN	0.00126	0.998	0.992	0.959	0.983
RN	0.00258	0.984	0.946	0.973	0.967667
RP	0.00345	0.998	0.985	0.932	0.971667

4.4.1.2 Design Details of ANNs

For all the experiments carried out for selection of best NN model for particular insert, details of ANN design is given below.

- Network type feed-forward back propagation, subsequent layer getting inputs only from adjacent layers.
- Number of inputs fixed to 3. Feed, speed, depth of cut.
- Number of Hidden Layers varied between 1 and 2.
- Type of training function used: TRAINLM (Levenberg-Marquardt Algorithm). It is robust technique and it finds solution even if it starts very far off accurate results. It is built in function in Matlab NN tool.
- Mean Square Error (MSE) function is used to judge the performance of the NN model and determine weight changes. Maximum number of epochs fixed to 100 for proper training. Performance goal to be set as 0. Total number of 6 validation checks are carried out.

4.4.2 Network Simulation

For checking the validity of obtained results, the Neural Network model is used to predict values of machining parameters that it has not yet encountered during training. Thus we can understand the accuracy with which selected Neural Network is performing. 10% data from available actual results is randomly chosen for validation. Network is simulated using selected values. Output is then plotted for each Neural Network representing each type of insert. Actual values of the surface roughness for those values of machining parameters is also plotted on the same graph so as to have better visuals of accuracy of simulation or prediction.

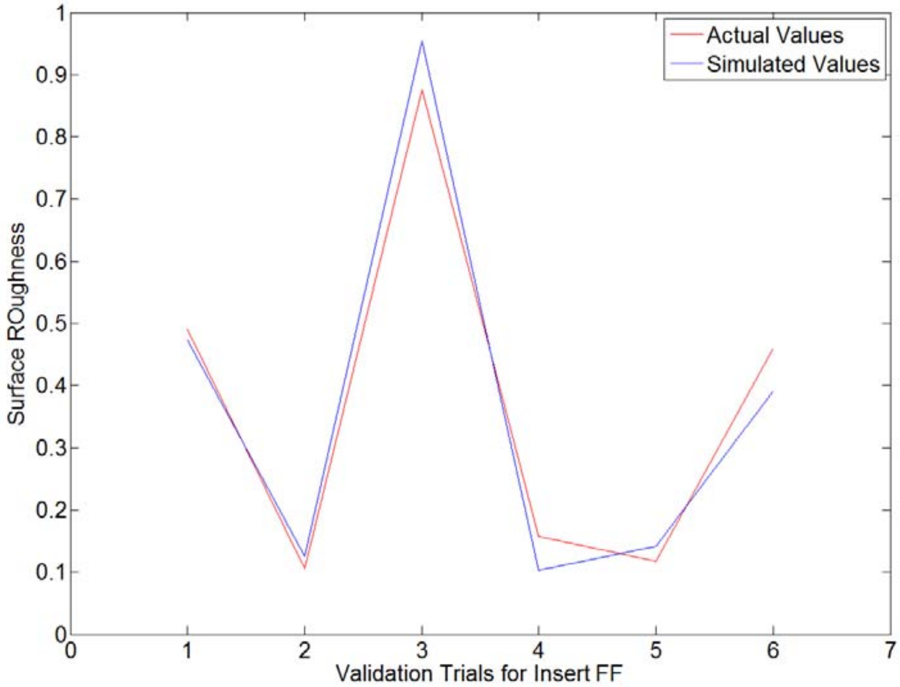


Figure 4.7 Comparison of actual and predicted surface roughness values for Insert TNMG160408FF

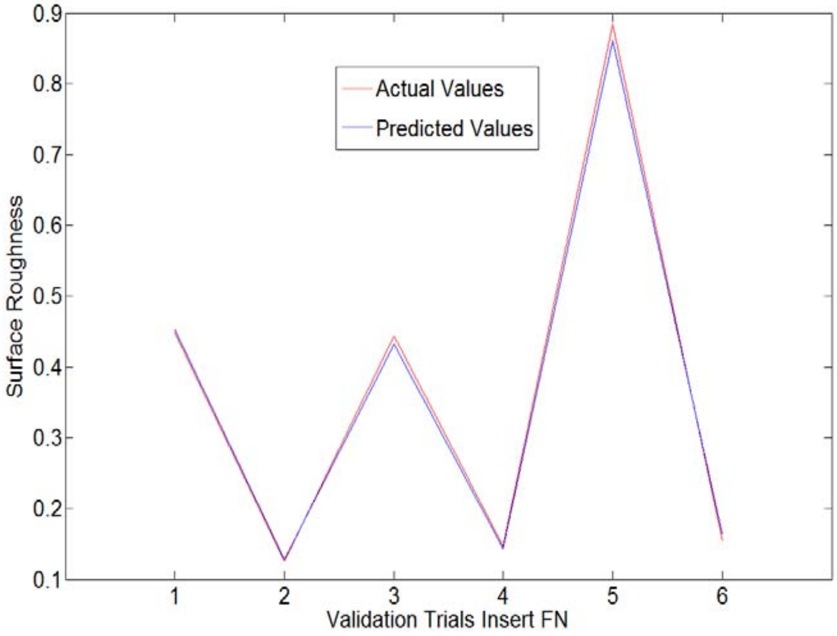


Figure 4.8 Comparison of actual and predicted surface roughness values for Insert TNMG160408FN

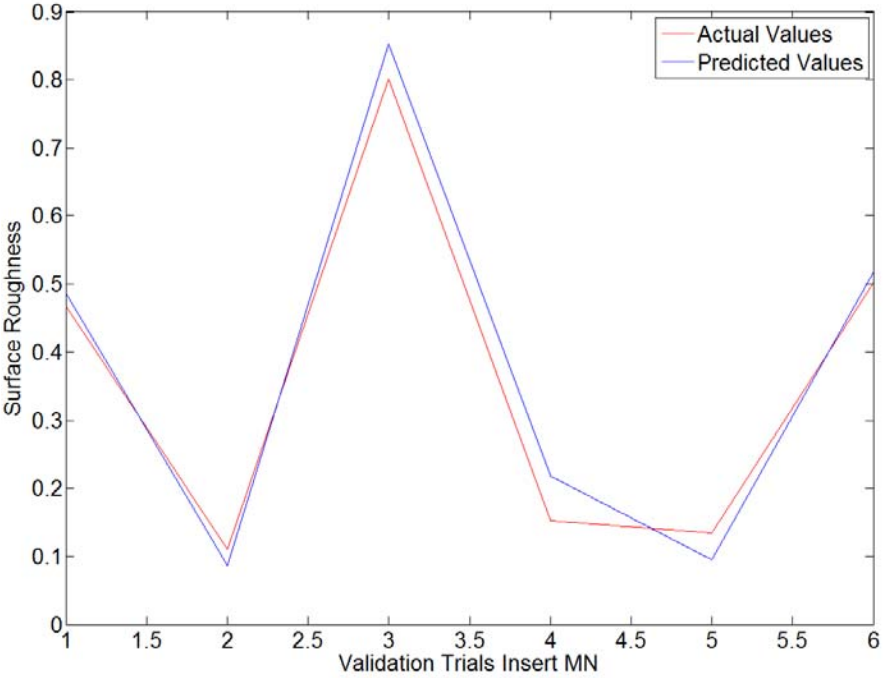


Figure 4.9 Comparison of actual and predicted surface roughness values for Insert TNMG160408MN

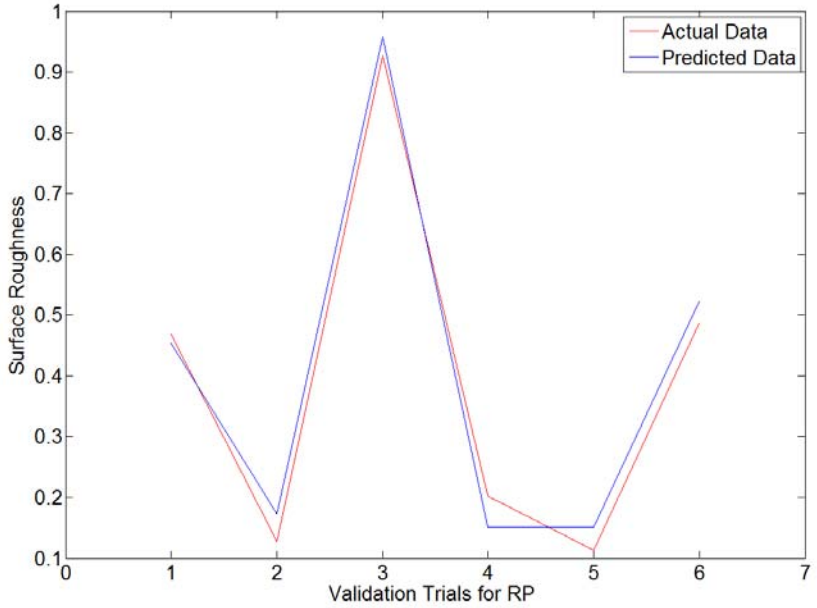


Figure 4.10 Comparison of actual and predicted surface roughness values for Insert TNMG160408RP

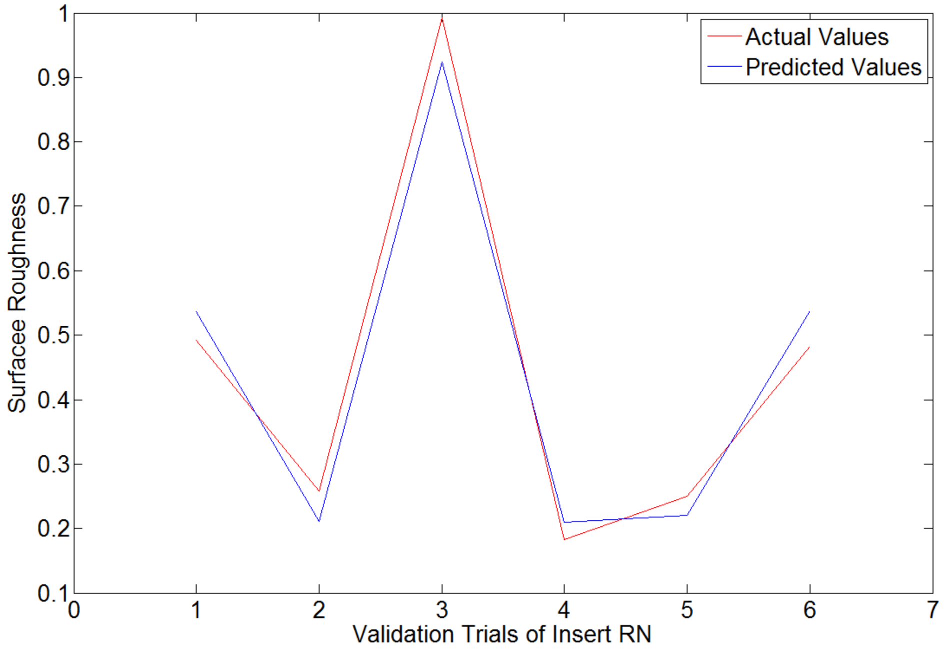


Figure 4.11 Comparison of actual and predicted surface roughness values for Insert TNMG160408R

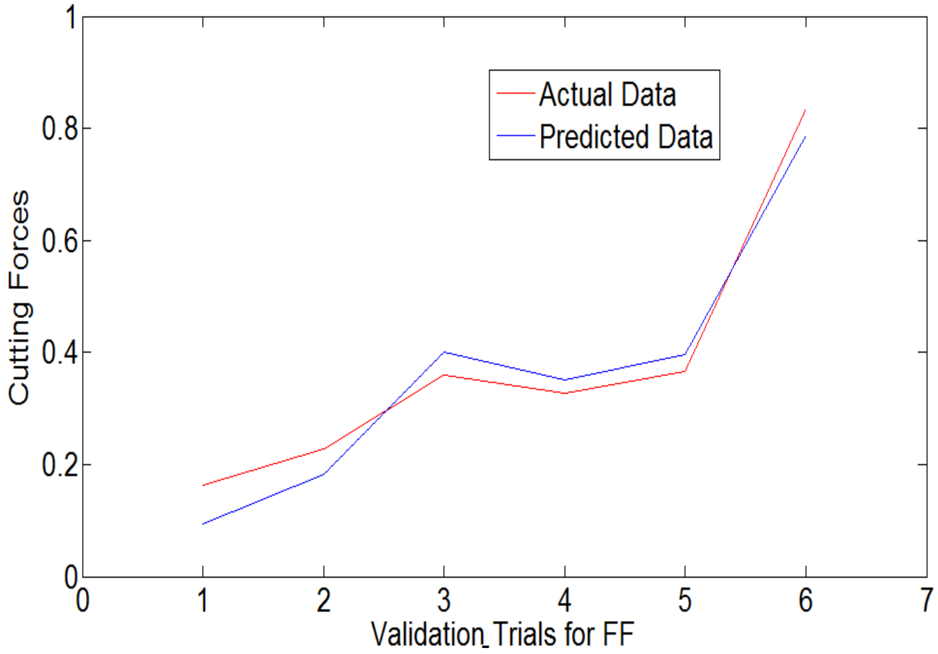


Figure 4.12 Comparison of actual and predicted cutting forces values for Insert TNMG160408FF

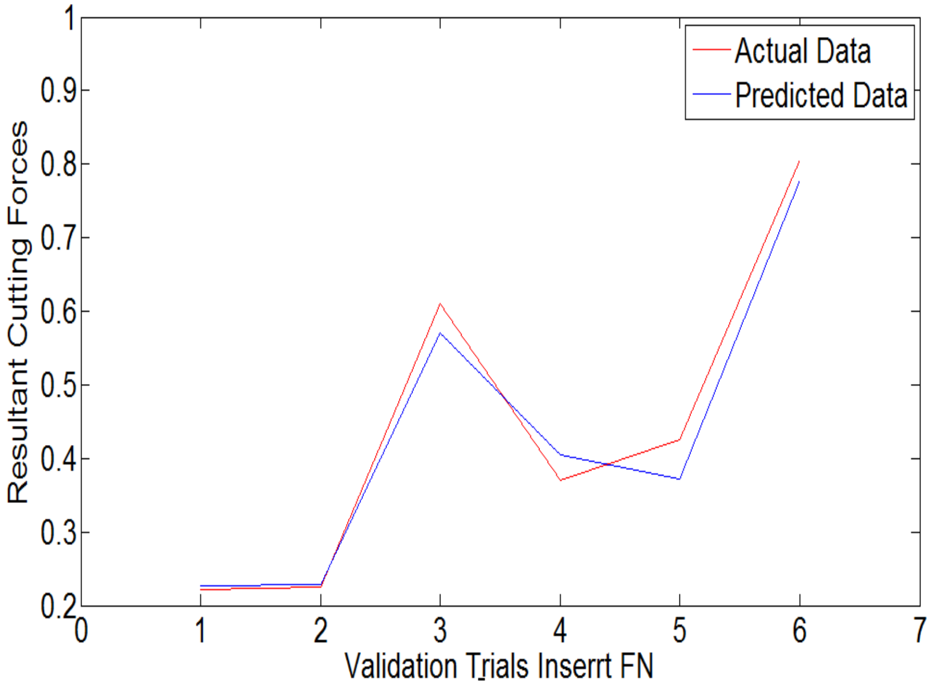


Figure 4.13 Comparison of actual and predicted cutting forces values for Insert TNMG160408FN

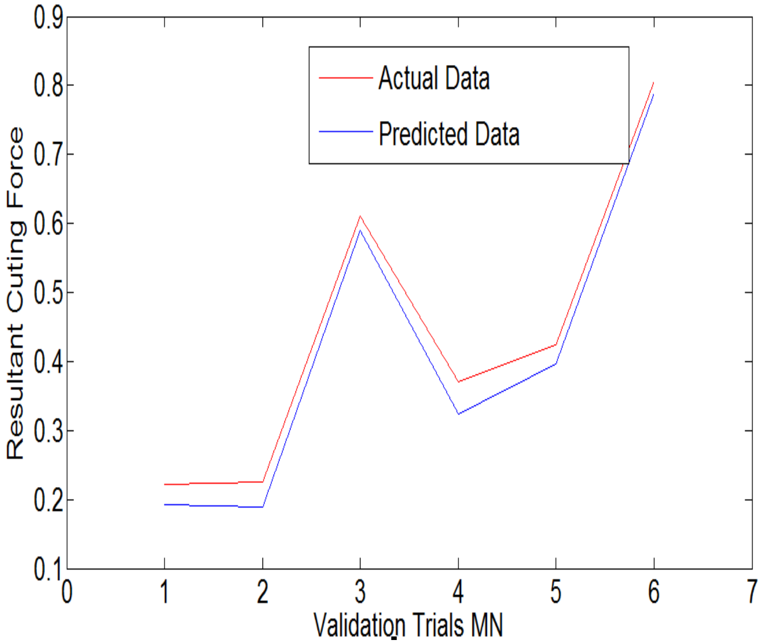


Figure 4.14 Comparison of actual and predicted cutting forces values for Insert TNMG160408MN

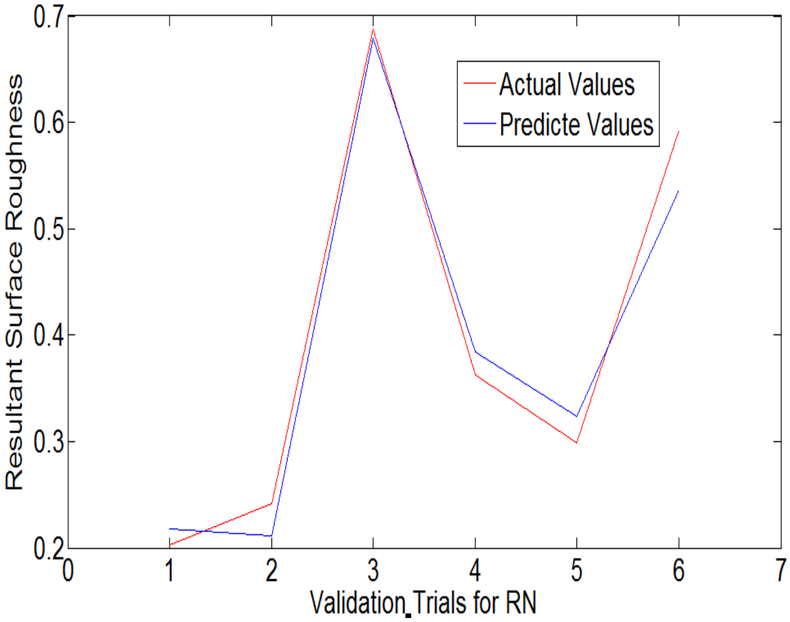


Figure 4.15 Comparison of actual and predicted cutting forces values for Insert TNMG160408RN

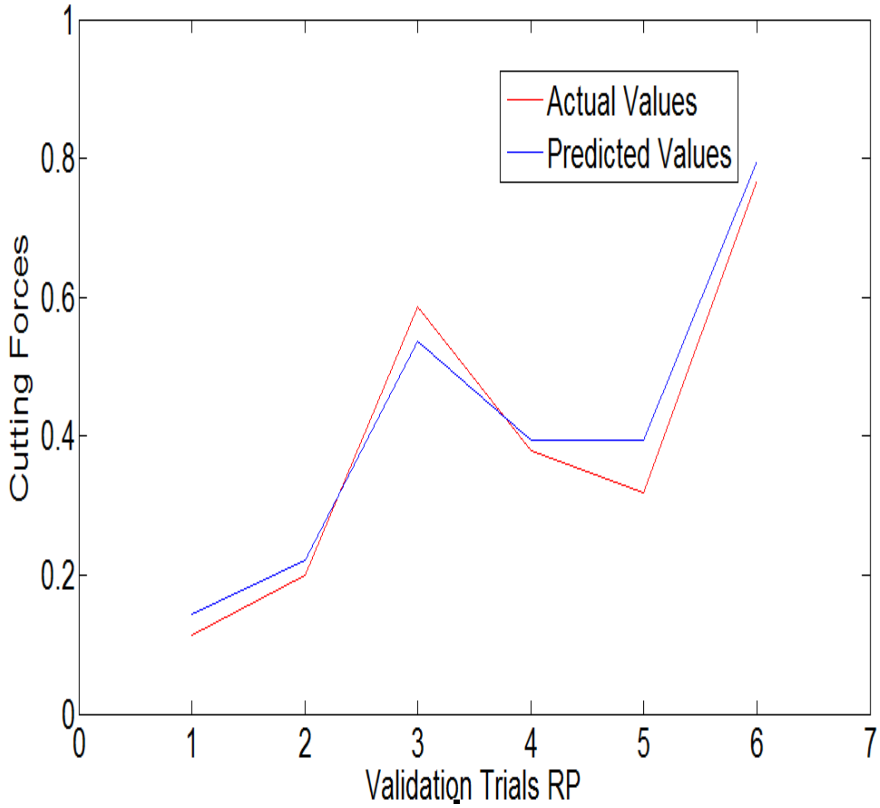


Figure 4.16 Comparison of actual and predicted cutting forces values for Insert TNMG160408RP

Table 4.5 Percentage error for values predicted by Insert FF

Parameter	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6
Resultant Cutting Force	-8.2867	-4.13058	3.879744	7.23306	8.119455	-5.79581
Surface Roughness	7.791287	4.046172	-2.7304	-6.77041	6.06633	8.368709

Table 4.6 Percentage error for values predicted by Insert FN

Parameter	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6
Resultant Cutting Force	2.650696	1.481016	-3.2581	5.444909	-7.85738	-3.35351
Surface Roughness	1.235022	3.2922	3.705668	-7.80028	4.698368	-4.68559

Table 4.7 Percentage error for values predicted by Insert MN

Parameter	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6
Resultant Cutting Force	-4.871	-4.03071	-3.28241	3.931796	-6.56621	-2.58617
Surface Roughness	-7.77751	-3.5498	6.196944	-6.48454	3.850617	4.933518

Table 4.8 Percentage error for values predicted by Insert RN

Parameter	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6
Resultant Cutting Force	7.050412	-12.4813	-1.35157	6.010696	8.2058	-9.62814
Surface Roughness	2.611493	-3.26777	-2.26538	7.773411	-4.54434	-7.04594

Table 4.9 Percentage error for values predicted by Insert RP

Parameter	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6
Resultant Cutting Force	-3.66102	-9.77929	9.209145	-3.78384	-1.95213	-3.71535
Surface Roughness	-0.94109	9.323578	-4.66953	-8.76684	-2.56426	3.288333

NN models are created to predict surface roughness and cutting forces. Results of validation of these NN models makes it is easy to understand that these models accurately predict values for unseen data. Hence these NN models can be used for further experimentation for optimization of machining parameters.

5. PSO (PARTICLE SWARM OPTIMIZATION TECHNIQUE FOR OPTIMIZATION OF TURNING PROCESS PARAMETERS)

5.1 Introduction

Once a reasonable level of accuracy is attained by the neural network model of turning process, it is used effectively for optimization of the process parameters. Calculating the optimum cutting parameters of the turning process is a very important step of this study. In the literature review section a lot of emphasis is done on the comparison among different optimization methods and selection of this technique. Considering the complicated nature of the objective, to optimize input cutting parameters of turning process to minimize surface roughness, Particle Swarm Optimization technique proves to be very helpful.

5.2 Optimization Model

In general, objective of the optimization problem is to achieve either minimum or the maximum value of the parameter, which is the target. This objective can be demonstrated using mathematical function, program code, as well as using neural network. This objective which is to be either minimized or maximized is subjected to partial constraints.

5.2.1 Objective Function

During the literature review, several objective functions were found that are used for machining optimization. Cutting time, tool life, production cost etc. have been used as frames to optimize turning parameters. However use of these functions is limited considering applicability and generalization. For example, mathematical equations representing cutting cost (Equation 2.1), tool cost (Equation 2.5), and material removal rate (Equation 2.7) are subjective because the parameters in these equations are obtained using deterministic approximations, or its applicability is very limited. For this reason, these parameters are only useful when all the process parameters are well known to the user. However not every parameter or outcome can be described analytically; in which cases the evolutionary Neural Network technique is used for predicting the output.

It has been observed that actual surface roughness measured from experimental data does not match theoretical values calculated from existing analytical formulation. This is because of characteristics such as adhesion, ploughing, and geometrical effects, which cannot be easily modeled using analytical models [51].

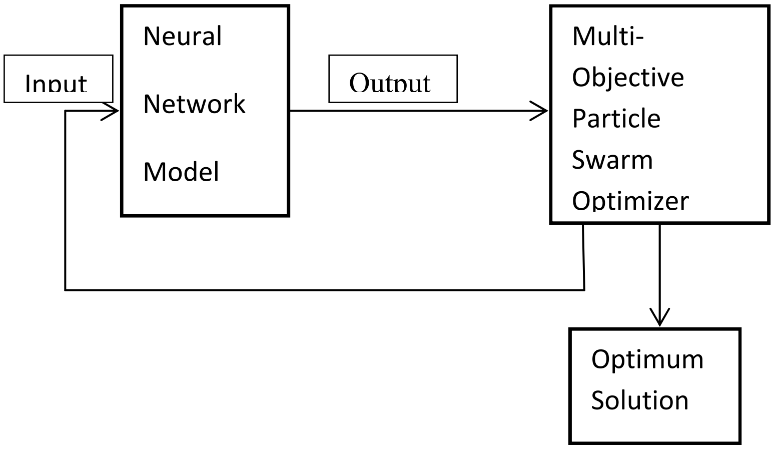


Figure 5.1 Swarm Intelligence Neural Network System [51]

Inputs provided to the Neural Network model are feed rate, spindle speed and depth of cut while the output is surface roughness and resultant cutting force. Neural Network model with sufficient accuracy integrated with multi-objective particle swarm optimizer will provide useful information to the user during selection of machining parameters.

5.2.2 Problem Constraints

For any optimization problem, besides objective function, constraints is the next important thing. For the optimization problem, to produce realistic and meaningful results, these constraints must be satisfied. These constraints are nothing but mathematically defined limitations for the process and are used to define the range within which the parameters must operate. For the turning process, constraints could be classified under parameter boundaries, tool life constraints, operating constraints (i.e. cutting force constraint and/or power constraint). Chip-tool interface constraints are also utilized in some optimization problems. The focus of this experiment deals with parameter boundaries such as range of feed rate, spindle speed and depth of cut. However, these constraints can be subdivided into input and output constraints. Input constraints are for limiting the values of input parameters. Values that are randomly selected by the PSO program for inputs are selected with these inputs. Another type of constraints is output constraints, resultant cutting force is considered among these constraints. If the predicted value of resultant cutting force is beyond the maximum value then the corresponding input set is also discarded.

1. The available range of cutting speed(m/min) is expressed in terms of upper and lower boundary

   $$V_{Min} < V < V_{Max} \tag{5.1}$$

   We provide lower boundary value to avoid formation of built up edges, whereas the upper boundary is for the safety of the operator.

2. Available range of feed rate (mm/rev) is expressed in terms of upper and lower boundary.

   $$f_{Min} < f < f_{Max} \tag{5.2}$$

3. Range of Depth of cut available is,

$$d_{Min} < d < d_{Max} \tag{5.3}$$

4. The maximum cutting force permitted by the cutting system is the maximum allowable force that either a tool or work piece can withstand. This constraint allows the force prediction model to set up the condition that, if the predicted force value is beyond the maximum cutting force permitted, then the user has to modify the values of feed or depth of cut in order to avoid causing the tool to be overstressed. In other words maximum force allowable can be used as physical limitation of the optimization process. This force is necessary as it limits the deflection of the tool or work piece. Deflection of the tool or work piece can induce the dimensional errors. With maximum value of allowable force being known it can also be used to limit the power consumed by the machine in certain cases. Hence the maximum cutting force can be given as

$$F_{Cutting} = k_1 * f_r^u * d_r^v < F_{Max} \tag{5.4}$$

$F_{Cutting}$ = Cutting force with current input parameters

k_1, u, v = Constants of the cutting force equations.

f_r = Feed rate in turning

d_r = Depth of cut in turning

F_{Max} = Maximum allowable force.

In this study, out of all the force characteristics calculated by the force model, resultant force was found to be maximum. The following Figure depicts how resultant force acts on the tool. In turning process, resultant force has components in 3 directions i.e. X, Y, and Z axis direction. This resultant force is used to calculate the maximum allowable stress on the tool without causing any deflection. Safety factor selection is also very helpful as it defines the margin of security.

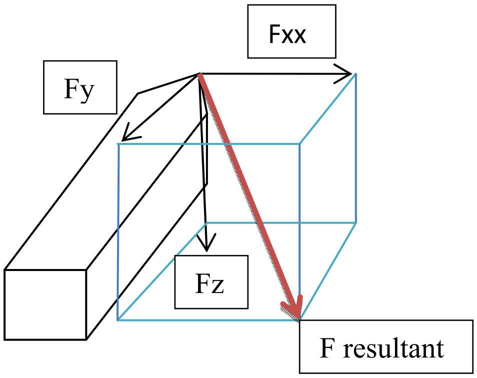


Figure 5.2 Components of resultant cutting force

In another study of the components of resultant cutting force it was noticed that forces in Z direction are the most dominant of them all. As matter of fact, force component in Z direction constitutes about 80% of the resultant force. This was proved in the following study. In this study resultant force was calculated using following formula.

$$F_{Resultant} = \sqrt{F_x^2 + F_y^2 + F_z^2} \qquad (5.5)$$

Now the cutting force in Z direction is plotted along with values of resultant force for some trials. Selection of trials was done randomly.

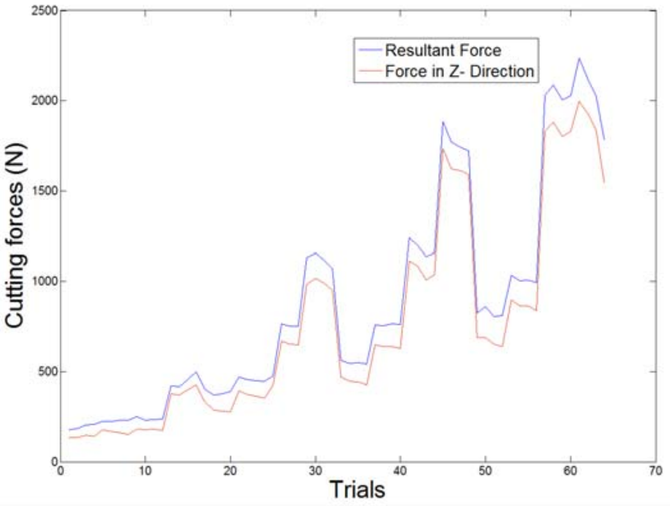


Figure 5.3 Comparison of magnitude of resultant cutting forces and force in Z-direction

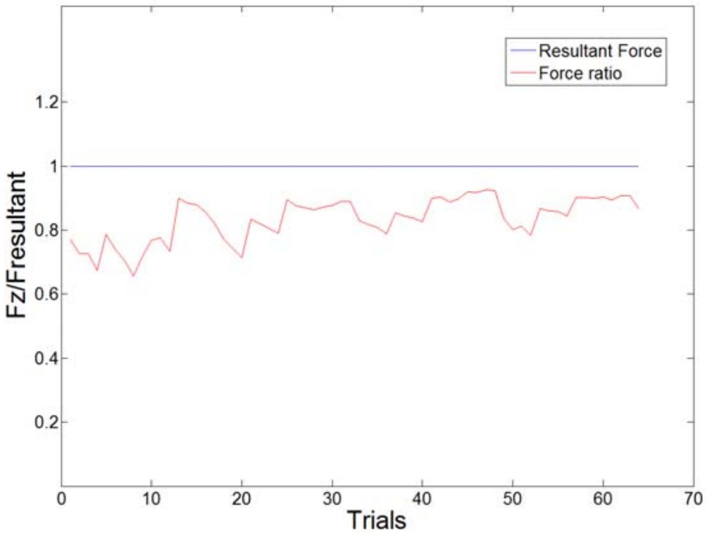


Figure 5.4 Maximum Force component Ratio

The F_{Max} in the Equation 5.4 is a constraint for the optimization problem, as keeping the force beyond the value where tool can break is one of the objectives. This constraint also interferes with the tolerance and factor of safety. When resultant cutting force is ensured to be always less than F_{Max}, we are limiting the maximum component forces going beyond such magnitude.

5.3 Optimization Methodology

The objective of this study is to obtain minimum surface roughness while optimizing the control factors i.e. input parameters of the turning process simulation model. The objective function used for optimization objective is constructed using Neural Network model. Using different optimization techniques for multiple parameter optimization task, there is always a concern that a path of the optimization process should be concluding at global minimum rather than stopped at local minima. As this optimization objective is complex and non-linear, we have chosen to apply the "Evolutionary Techniques" of optimization. The Particle Swarm Optimization (PSO) technique has shown great caliber while dealing with multi-parameter optimization tasks and demonstrated excellent results. PSO also holds advantage over genetic algorithms for optimization objectives of similar nature. In genetic algorithms, chromosomes share the information so that the whole community gradually moves to area where best solution could be spotted. In PSO the information is sent out by best particle and then the whole group converges around the same point [52]. In PSO, each particle has assigned velocity with which it traverses through the solution space. Using PSO technique it is possible to search the optimum solution in multiple dimensions.

The Particle Swarm Optimization is a stochastic optimization technique. The PSO is originated from the research of food hunting of birds, how the course of flight flock of birds would always scatter and gather as whole with individuals always keeping most suitable distance. It is found that there is social information sharing mechanism in biological communities, which is important from their evolution point of view. This provides basis for formulation of PSO [52]. Specifically, the PSO technique is mainly inspired in social behaviors where the search for a "common best solution" drives the whole set of particles.

The PSO technique has ability to deal with problems related to non-differentiality, high dimensions, multiple optima as well as nonlinearity. In PSO technique each single solution is a particle in the search place. These particles hold a fitness value which is evaluated by fitness function. During the optimization process, particles traverse along with this path with these velocities within the search space. This search space is limited by the constraints applied to the problem by process limitations. The best position of the particle found during this process, is the best solution found by that particle. Also, the best position of all the particles as a group is considered as best solution. Best position is called as P_{Best}, and best solution is called as G_{Best}. Every particle, as it travels within the search space keeps updating these solutions.

In Evolutionary Techniques, search for potential solution is carried out in allotted space as opposed to traditional search techniques. Using population interaction, these techniques provide rapid solution. Population based search methods can be defined as:

$$P' = m(f(P)) \tag{5.6}$$

Where, P is multiple sets of positions in search space also called population, f is fitness function, 'P' is a vector that signifies the optimal value of each member of P and m is manipulation function of the population.

In PSO technique, each particle is a point in solution space. Therefore, for total of N particles of swarm, position of I^{th} particle will be X_I. While I=1,2,3,4……….,N. Then X_I can be represented as follows.

$$X_I = (x_{i1}, x_{i2}, x_{i3}, \dots , x_{1N}) \tag{5.7}$$

As mentioned before, for every particle, P_{Best} i.e. best value based upon outcome of fitness function (Neural Network model in this case) is considered as "previous best position" and it's represented as:

$$P_I = (p_{i1}, p_{i2}, p_{i3}, \dots, p_{iN}) \tag{5.8}$$

G_{Best} i.e. best position for the whole group of particles. G implies total group. Hence G_{Best} is represented as:

$$P_G = (P_{g1}, p_{g2}, p_{g3}, \dots , p_{gN}) \tag{5.9}$$

Another fundamental characteristic of the swarm particle is the velocity. It is the speed with which the particle traverses through the solution space. Velocity vector is represented as:

$V_I = (v_{i1}, v_{i2}, v_{i3}, \dots , v_{iN})$ (5.10)

For every step of the process, velocity is calculated based on previous best individual position as well as best position of the group. This is calculated using following formula.

$v_{in} = w*v_{in}+c_1*rand_1*(p_{in} - x_{in}) +c_2*rand_2*(p_{gn} - x_{in})$ (5.11)

Where,
W: inertia weight
C1 and c2: two positive constants
Rand1 and rand2: two random functions in the range of [0,1]

This equation consist three parts. First part is previous velocity of the swarm which represents present stage. Second part is cognition model and third part is social modal. These three parts together determines the ability of searching the solution space [52]. As the particle moves to the next position, it is calculated as:

$x_{in} = x_{in} +v_{in}$ (5.12)

The performance of every swarm is gauged by using fitness function.

For every swarm particle, the values of feed rate, spindle speed and depth of cut define the position and weather the swarm has reached its optimum position or not. While doing this, the particle will keep traversing in the three dimensional space. The performance of course will be defined by objective, which in this case will give the surface roughness.

The optimization procedure can be summarized as follows:

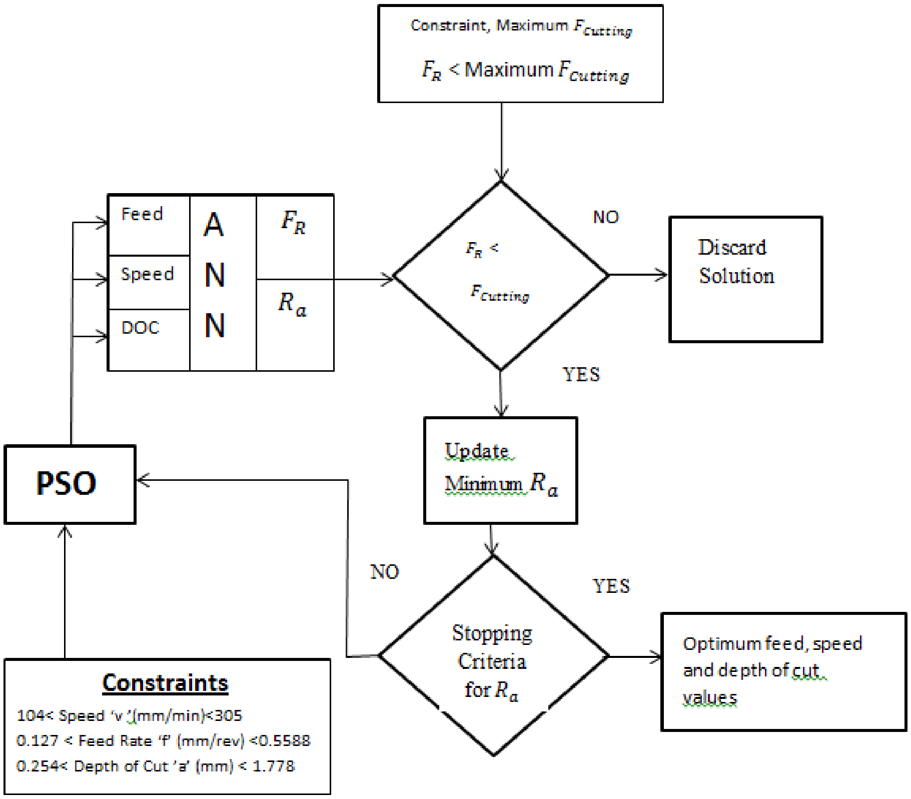

Constraint, Maximum $F_{Cutting}$
F_R < Maximum $F_{Cutting}$
Feed
Speed
DOC
A
N
N
F_R
R_a
F_R <
$F_{Cutting}$
NO
Discard Solution
YES
Update Minimum R_a
PSO
NO
YES
Stopping Criteria for R_a
Optimum feed, speed and depth of cut values
Constraints
104< Speed 'v '(mm/min)<305
0.127 < Feed Rate 'f' (mm/rev) <0.5588
0.254< Depth of Cut 'a' (mm) < 1.778

Figure 5.5 ANN – PSO Relation, pictorial representation

Where,

R_a : Surface Roughness

R_{Force} : Resultant cutting force

Feed : Feed rate values for the process (mm/rev)

Speed : Speed or revolution of the machine (m/min)

DOC : Depth of Cut (mm)

ANN : Artificial Neural network

Input : Recorded Input data from trials used for training ANN model

Output : Recorded output data from trials used for training ANN model

PSO : Particle swarm optimization

This Figure above is the representation of course of action of decision making, evaluation and result is calculated during the PSO process. As shown in the Figure, Input and output matrices which consists data from experimentation, are used to train and validate the neural network. Range of values of Feed rate, spindle speed and depth of cut are used by NN model to predict surface roughness and cutting forces values. The next step is using the values predicted by the NN model for the PSO process. Boundaries are set up for the choice of input values i.e. feed, speed and depth of cut which are called constrains in problem definition. This is followed by applying the PSO methodology until the desired value for surface roughness is reached for corresponding input values. Until the desired roughness is achieved, the process parameters are re-defined and new iteration is started. This goes on until all the stopping criteria for the optimization process is being met.

5.3.1 Optimization Program Code

For the PSO program, to perform well and produce optimum results, problem constraints are very essential. Constraining the control factors which are feed rate, spindle speed and depth of cut, is part of user inputs. The other aspects are, data perpetration, and, application of PSO and finding optimum results. Choice of a number of particles to do the optima search as well as selection the criteria for most number of iterations are also considered as important elements categorized under user inputs. User inputs selected for PSO code is as follows.

$0.127 <$ Feed Rate (mm/rev) < 0.5588
$0.254 <$ Depth of Cut (mm) < 1.778
$104 <$ Speed (m/min) < 305
Number of particles: 20 [53]

Once the problem constraints and process constants or other user inputs are assigned, optimization procedure can be started. It can be roughly divided into four important aspects.

- Objective function and its evaluation
- Calculation of particle best position of the particle (P_{Best})
- Calculation of global best position (G_{Best})
- Stopping Criteria

We have discussed about objective function and its evaluation in previous topics. The calculations for particle best and global best positions are most crucial part of PSO technique. The stopping criteria have been established for the optimization program to stop the iteration. It is, condition 1, the difference between the surface roughness values for the consecutive iterations should be less than the epsilon value mention in the code. Condition 2, checking condition 1, for next 50 iteration to ensure consistency. The Figure 5.7 represents the values of roughness obtained from the objective function at different iterations kept reducing until the stopping criteria were achieved. Surface roughness values are observed to drop at high rate until first 25 iteration. After 25 iterations surface roughness values has very low rate of change and after 150 iterations it hardly changes.
The necessary equations for doing these calculations are available in particle swarm optimization procedure. Finally, the program plots for the final results and output values can be seen as shown in Figure 5.8 onwards. Time taken by the PSO program is approximately 4-5 minutes, which is convenient considering how long it takes to converge for the PSO for other applications. It can be considered that the PSO program used for this study is efficient when it comes to time taken for calculations.

Table 5.1 Representing the values of characteristic constant [45, 46]

Characteristic constants	C1	C2	Epsilon
Value	1.49445	1.49445	0.00009

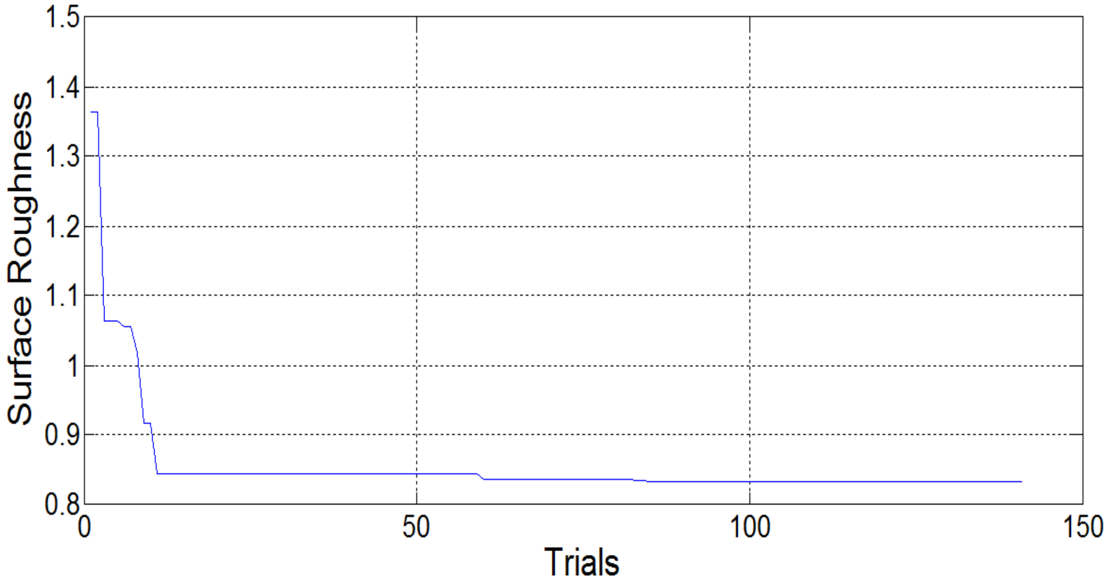


Figure 5.6 Minimum surface roughness vs. trials or number of iterations

Values used in Table 5.1 are as follows, C1 is cognition learning rate, C2 is social learning rate and epsilon is the cutoff value or termination criteria for the PSO program. The learning rate is used as 1.49445 as it is recommended for obtaining better results by author of the method [53].

5.3.2 Optimization Results

The optimization program generates the positions of the particles for every iteration until the stopping criteria are satisfied. During these iterations particles change their positions within the workspace that is limited by the constrains. New positions are calculated using the available set of formulae. With every iteration movement of the particles becomes oriented towards the optimum positions. The change in positions and orientations can be observed in the Figures 5.8 onwards. Graphs represent the positions of the particles at the start, middle and end of the iterations.

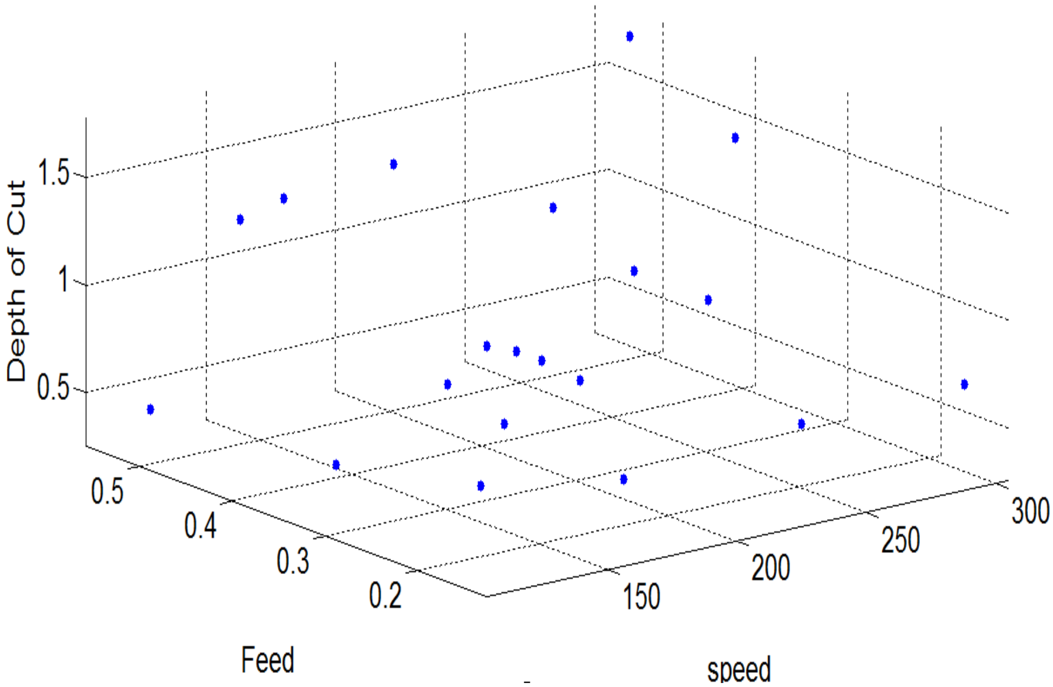


Figure 5.7 Initial positions of particles (First Iteration)

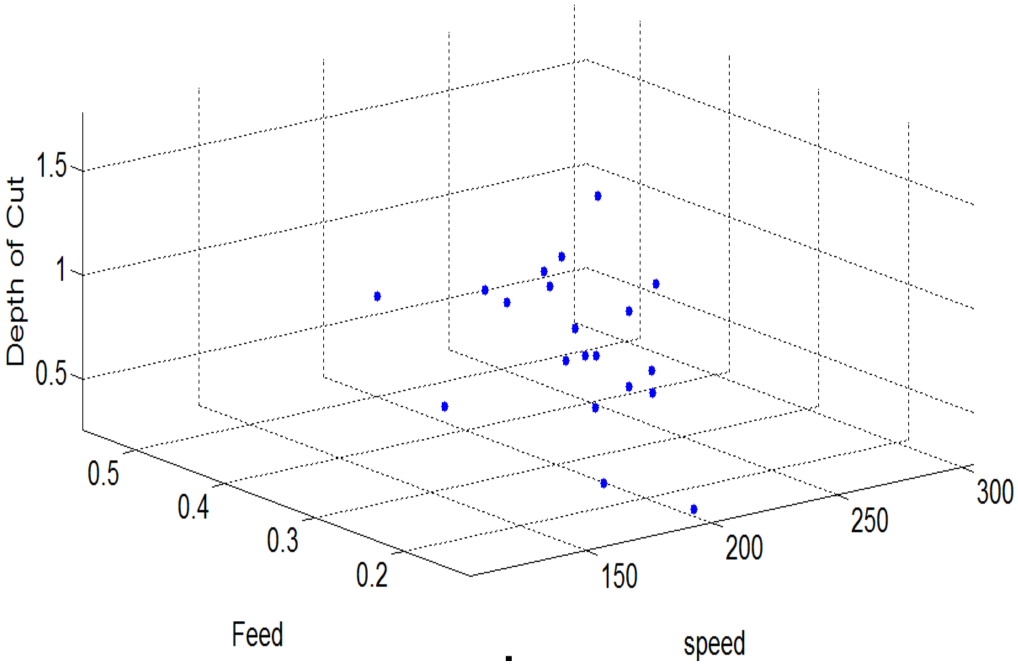


Figure 5.8 Particles travel path in the middle of the process (Iteration 25)

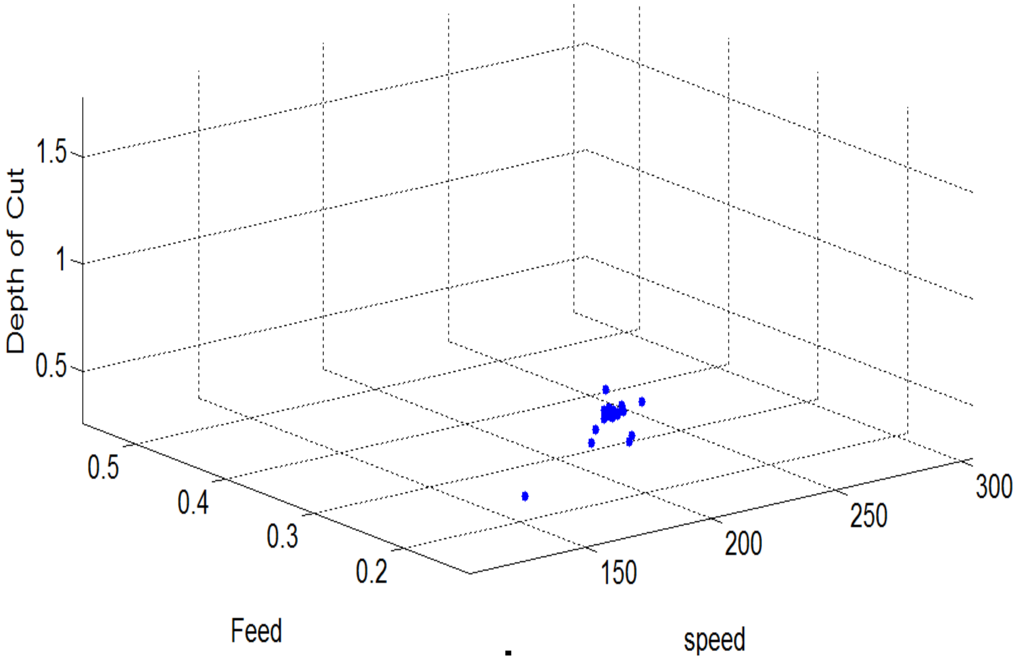


Figure 5.9 Particles travel path in the middle of the process (Iteration 55)

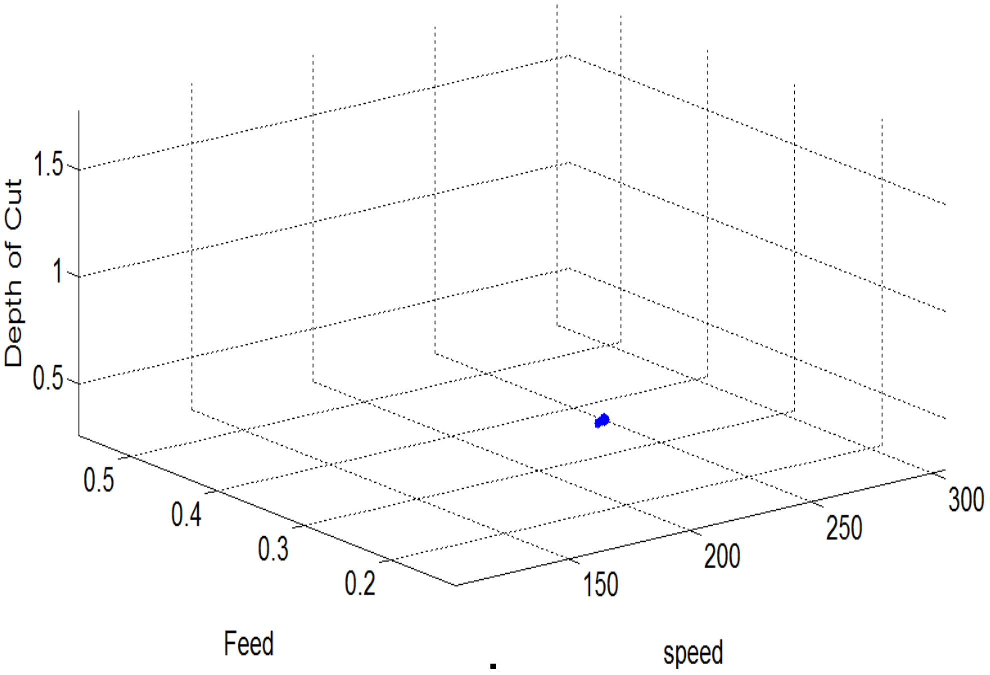


Figure 5.10 Final positions of the particles (Iteration 125)

The 3D graphs are plotted, having spindle speed, feed rate, and, depth of cut on X, Y and Z axis respectively. For every point in solution space, there is set of values of control factors. In these Figures, the “●“ sign represents the location of each particle in the solution space. The positions and path of travel of these particles represent the travel of particles to optimum position. In study carried out during Chapter 3, for most influential parameter among the input parameters of the process i.e. Feed rate, Spindle speed and Depth of cut, most influential parameter responsible for surface roughness is feed rate. Hence the values for feed rate and its corresponding value of surface roughness for every iteration is plotted in the following graph. Also, graph displaying speed and its effect on surface roughness is displayed below.

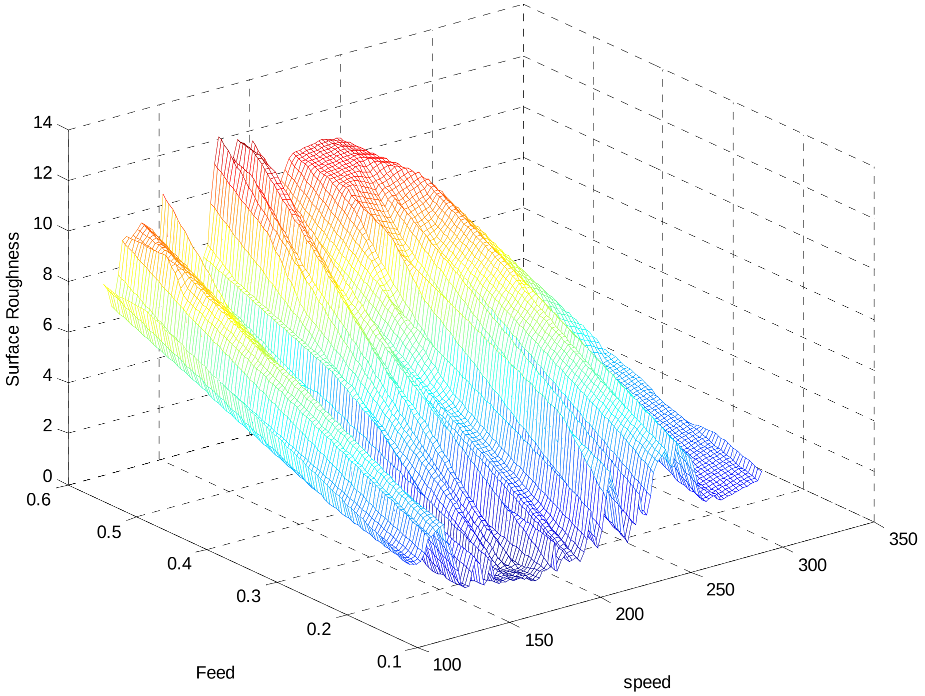


Figure 5.11 Surface roughness vs. Feed vs. speed for iterations

With increase in the feed rate, surface roughness is observed to be lowered, especially when the feed rate is within 0.13 mm/rev to 0.16 mm/rev. However, in the middle of process for calculation of optimization point, the program reached local minimum, since PSO technique is capable of handling such circumstances program could calculate further until global minimum is reached. Similarly while studying the effects of speed on surface roughness surface roughness quality actually observed to been reduced, especially within the range of 170 m/min to 195 m/min. Results obtained from the code based on PSO algorithm are mentioned below in the results table. The code stops working when all the stopping conditions are satisfied. The corresponding parameters at that point are optimum cutting parameter for achieving minimum surface roughness. Along with the surface roughness, resultant cutting force with optimum cutting parameters is also determined so as to make sure it does not crosses the maximum force bearing capacity of the cutting tool.

Table 5.2 Optimum cutting conditions for corresponding objectives

Cutting Parameter	**Corresponding Optimum Value**
Feed Rate (mm/rev)	0.1533
Speed (m/min)	176.6065
Depth of cut (mm)	0.8178

Table 5.3 Minimum values of characteristics attained

Resultant Parameters	**Corresponding Minimum Value**
Surface Roughness R_a (μm)	0.8357
Resultant Cutting Force R_f (N)	631.5206

5.4 Extended Optimization Model for Turning Procedure

While choosing the cutting parameters for turning process, process constraints have to be considered. In some cases, constraints are more right e.g. diameter of the work piece, specific choice of depth of cut etc. It could be possible that sometimes selection of depth of cut is not the choice of the operator but the process constraint. Constraint of selection of depth of cut value is mostly seen during finishing operation or when there are tighter manufacturing tolerances. Considering the tool manufacturer's recommendations or material properties such as chip formation it is mandatory to follow these limitations for safety. To make the optimization model compatible for these conditions, we have chosen to extend the scope of process constraints to selection of specific depth of cut. PSO procedure is used for this extension of model. Constrains regarding feed and speed are kept similar to that of previous optimization exercise i.e.

$0.127 <$ Feed Rate (mm/rev) < 0.5588
$104 <$ Speed (m/min) < 305

In previous optimization model, the solution space was three dimensional, whereas for extended model it is 2 dimensional since depth of cut is now a constraint input by a user, for trial purposes we used depth of cut equal to 0.75 mm. Extended model can also be used to cross check validity of the results of three dimension optimization model optimization model and vice-a-versa. Code for extended optimization model consists similar set of equations as that of mentioned above for calculating the positions, velocities, particle best and global best values. Following Figure is the graph of the surface roughness values obtained from each iteration of two dimensional optimization model.

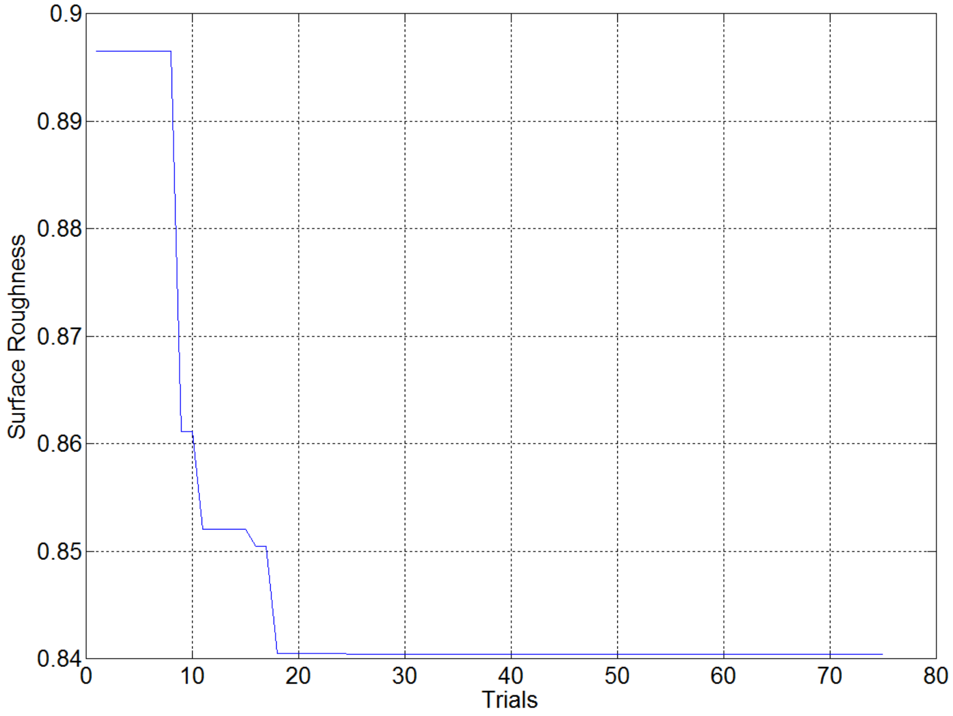


Figure 5.12 Change in the values of surface roughness with progressing iteration

Major change in the values occurs among iteration 5 to iteration 20, iteration 21 onwards, program continues until all the stopping criteria are satisfied, iteration 76 is the last iteration. At this point, the model concludes to have found optimum position. During this procedure, particles keep changing their position depending upon, history of particle path, and, latest calculations for optimum position. Resulting particle positions are displayed in the graphs below. First graph shows particle positions for the very first iterations, second shows position of the particles when the optimization calculations are in progress, and, last graph showing the particle path merging at the optimum position. Particles are scattered at first, then they reorient themselves in order to search for the optimum position.

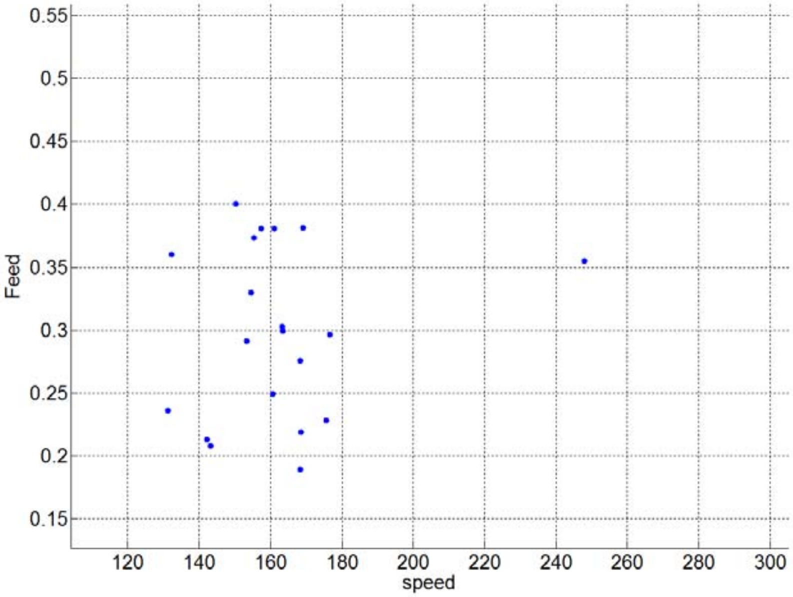


Figure 5.13 Particle positions during start of optimization procedure (Iteration 1)

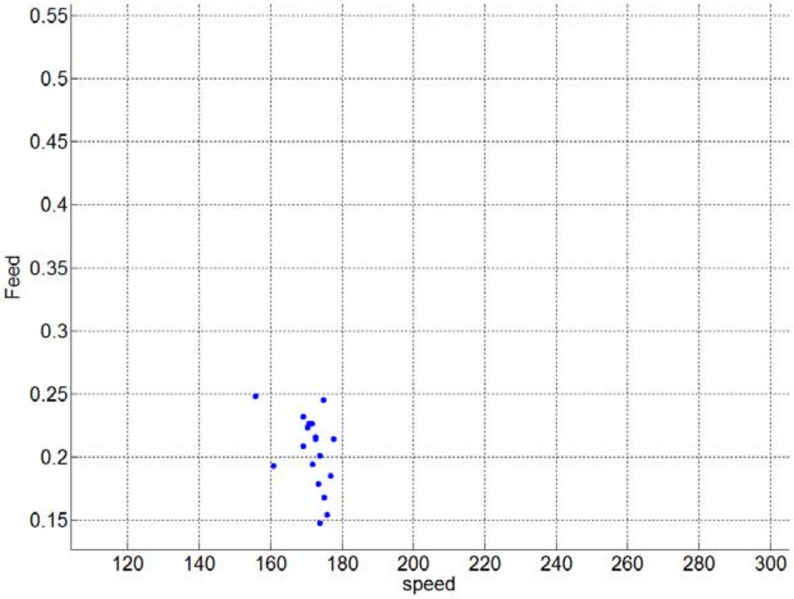


Figure 5.14 Particle position during middle of the optimization procedure (Iteration 15)

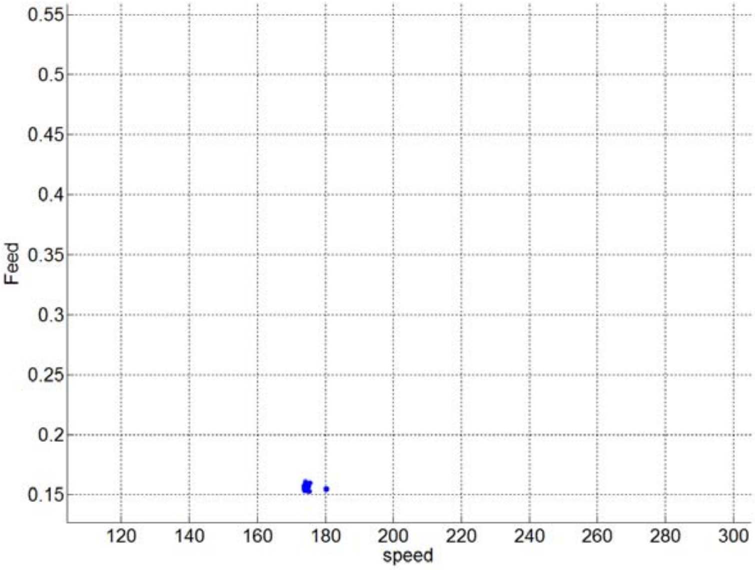


Figure 5.15 particle positions close to end of iterations (Iteration 50)

Values of spindle speed and feed rate for respective iterations are plotted so as to have clear understanding about their effect on surface roughness. It is very clear from the graphs that lower values of surface roughness can be achieved with lower values of the feed rate, especially within the range of 0.155 mm/rev up to 0.165 mm/rev. Spindle speed has similar effect on surface roughness. As spindle speed increases up to 170 m/min surface roughness values are observed to be very low which is acceptable and feasible.

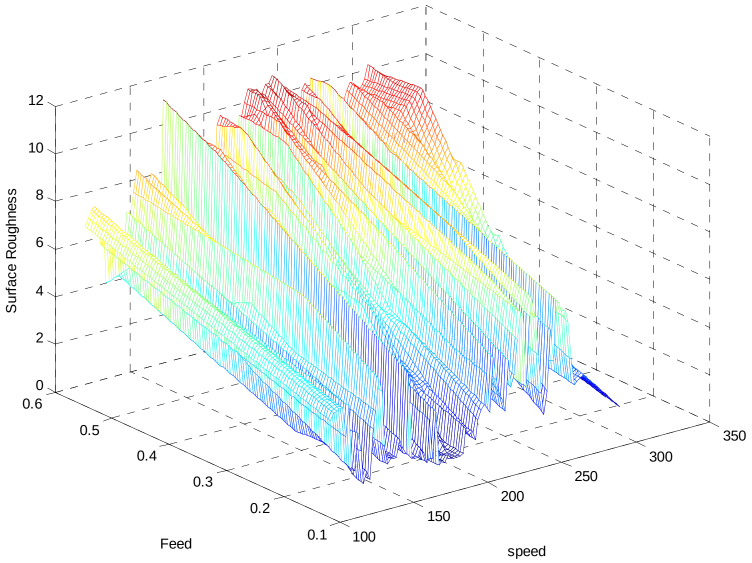


Figure 5.16 Effect of feed, speed on surface roughness

Multiple values of depth of cut are used to find out the optimum values of other parameters. Following table depicts the result of these trials for achieving minimum surface roughness and corresponding resultant force.

Table 5.4 Optimum cutting parameter values obtained from different depth of cut values

	Optimum Parameter Value			
Depth of cut (mm)	**Feed (mm/rev)**	**Speed (m/min)**	**Surface Roughness(µm)**	**Resultant Force(N)**
0.65	0.1686	171.4845	0.8950	493.3968
0.75	0.1592	174.2946	0.8404	581.9895
0.85	0.1495	175.5681	0.8437	656.5289
0.95	0.1270	183.9613	0.9194	697.5184
1.5	0.1792	304.9819	1.5926	1228

These values confirm that the optimum values of control parameters to ensure minimum surface roughness and resultant force for given choice of depth of cut are correct. For value of depth of cut which was considered to be its optimum, in three dimensional optimization process i.e. when depth of cut is not a constraint, when used for two dimensional optimization process, values obtained for other two control parameters are in proximity. This ensures the optimization model to be useful even when limitations such as depth of cut is presented because process or material or manufacturing constraints.

Generally the surface roughness requirements are characterized according to the type of application the manufactured part is going to be used for e.g. aerospace, health care, automobile, medical industry (especially for medial operations) needs surface quality to be very fine, whereas for welding process, parts that are out of reach for operator etc. do not have high standards when it comes to surface quality. Predicted optimum values from either three input parameters model or two input parameter model are between 0.65 to 0.85 microns. These values for surface roughness are well within the acceptable range in the industry.

6. CONCLUSIONS AND FUTURE RESEARCH

6.1 Conclusions

Considering the increasing need to use turning process in manufacturing industry, existing advanced manufacturing systems have not done so well, when it comes to providing optimum solutions. Existing systems have contributed in process planning. Considering the multidimensional nature of the turning process, input and output process parameters have highly non-linear relationship. Analytical and mathematical alternatives were presented so as to simulate the process in order to understand nature of relationship between the input and the output process parameters. These approaches had very limited applicability due to its limitations, and, their generalization characteristics were very limited. However, Artificial Intelligence techniques have shown promising characteristics in spite of being expensive. In this thesis, Artificial Neural Network, which is Artificial Intelligence technique, is utilized. This thesis deals with effective simulation of turning process and machining parameter optimization. Accomplishments are summarized as follows.

- An Artificial Neural Network technique is used to simulate accurate and generic model of turning process. This model is successfully implementation of ANN model is carried out. The ANN modeling technique is more accurate as compared to any other modeling technique used so far. This model was successfully trained to predict/estimate the desired outputs. Generalization characteristics of this technique are better than other alternative techniques used before. This technique, can be used not only to simulate turning process, but also other manufacturing process.
- Matlab Neural Network tool box is used in order to simulate the turning process, which would be very useful to the machine operator for selection of process parameters.

- Feed forward back propagation type of Neural Network was found useful in modeling turning process because, its ability to predict the process outputs for data it has never seen before. It is very efficient NN model type. Specialty of Back propagation technique is that, when an error is detected between predicted and actual values, it can be traced back to the processing elements responsible for it, and correction can be made. It has very good generalization capabilities. It is proved to be very useful while using NN model along with optimization algorithm for calculations of optimum machining characteristics.
- Statistical Design of Experiments proved to be an effective method, to reduce the amount of required training data for the model to learn to predict accurately. DOE approach reduces experimentation cost and time as only fraction of the data is sufficient for modeling the process. Selection of four levels of machining parameters/control factors was sufficient information for training of ANN model. Orthogonal Arrays were implemented successfully.
- Since the data used for training the ANN model was recorded from actual experimentation, the ANN model takes into account static effects of the process. For validation of performance of ANN model, real time data was utilized, choosing randomly from the available recorded readings. Predicted results were in agreement with actual results, which support our claim of accurate modeling of the process.
- Objective function used for optimization task, utilizes ANN model to predict the outcomes of the turning process i.e. surface roughness and resultant cutting force. Particle Swarm Optimization algorithm was used for calculations of optimum input parameter values for the turning process in order to minimize the surface roughness and resultant cutting force. Effectiveness of this procedure is demonstrated by its comparison with extended model for optimization process using similar objective function but more constraints.
- The proposed methodology can also be used for many other machining processes as it allows the user to evaluate multidimensional scenario, and helps to predict most efficient set of process parameters with sufficient accuracy.

- Results obtained from optimization exercise have more benefits as that if from recommendations from material data handbook.

6.2 Future Research

Even though, the ANN model paired along with PSO technique provides satisfactory results, number of advancements could be done to the ANN model in order to improve its abilities for proposed objective.

- The performance of the ANN model for turning process simulation could further be improved by including additional machining related variables as control factors. This would increase the accuracy and predictability of the model significantly. In fact, parameters such as vibrations of the tool, temperature of the cutting surface would contribute greatly towards improving the generalization ability of the model. Also more factors can be introduced to measure the performance of the process, such as, geometric tolerance, chip thickness etc.
- A comprehensive Taguchi method could be used to analyze the available data while reducing the effect of the noise factors involved in turning operation. This would improve the overall efficiency of the model as well as the understanding of this important machining operation.
- For further improvement in the model, surface roughness can be considered as a constraint and the objective function can be chosen to minimize time.

LIST OF REFERENCES

LIST OF REFERENCES

[1] M. F. Zaeh and F. Schwarz, "Modeling and Simulation of Process and Structure Interactions Considering Turning Operations," 2009.

[2] C. Van Luttervelt, T. Childs, I. Jawahir, F. Klocke, P. Venuvinod, Y. Altintas, E. Armarego, D. Dornfeld, I. Grabec, and J. Leopold, "Present Situation and Future Trends in Modelling of Machining Operations Progress Report of the CIRP Working Group 'Modelling of Machining Operations'," *CIRP Annals-Manufacturing Technology,* vol. 47, pp. 587-626, 1998.

[3] P. L. B. Oxley, "Mechanics of Machining: an Analytical Approach to Assessing Machinability.(Retroactive Coverage)," *Ellis Horwood Limited(UK), 1989,* p. 242, 1989.

[4] S. Hoppe, "Experimental and numerical analysis of chip formation in metal cutting," *Universitätsbibliothek*, 2003.

[5] I. N. Tansel, "Simulation of turning operations," *International Journal of Machine Tools and Manufacture,* vol. 30, pp. 535-547, 1990.

[6] L. Lauderbaugh and A. Ulsoy, "Dynamic modeling for control of the milling process," *J. Eng. Ind.(Trans. ASME),* vol. 110, pp. 367-375, 1988.

[7] R. Jerard and b. Fussell, "Process simulation and feed rate selection for three axis sculptured surface machining ," *Int. J. Manufacturing research,* vol. 1, pp. 136-157, 2006.

[8] E. Ceretti, C. Lazzaroni, L. Menegardo, and T. Altan, "Turning simulations using a three-dimensional FEM code," *Journal of Materials Processing Technology,* vol. 98, pp. 99-103, 2000.

[9] Y. Guo and C. Liu, "3D FEA modeling of hard turning," *Journal of manufacturing science and engineering,* vol. 124, p. 189, 2002.

[10] F. Klocke, C. Brecher, S. Witt, and P. Frank, "Integrated simulation of machine tool and process interaction for turning," *Steel research international,* vol. 78, pp. 831-838, 2007.

[11] G. E. Dieter and D. Bacon, *Mechanical metallurgy,* vol. 2: McGraw-Hill New York, 1986.

[12] J. Lorentzon and N. Jarrvtrat, "Tool wear geometry updating in Inconel 718 turning simulation," *CIRP International Workshop on Modeling Machining Operations.*2006.

[13] H. Bil et al., "Simulation of orthogonal metal cutting by finite element method," *CIRP International Workshop on Modelling machining operation,.*2004.

[14] T. Schermann, J. Marsolek, C. Schmidt, and J. Fleischer, "Aspects of the simulation of a cutting process with ABAQUS/Explicit in-cluding the interaction between the cutting process and the dynamic be-havior of the machine tool," 2006.

[15] J. Lorentzon and N. Jarrvtrat, "Tool wear geometry updating in Inconel 718 turning simulation," *CIRP International Workshop on Modeling Machining Operations,*2006.

[16] P. Arrazola, and F. Le Maitra, "Finite element modeling for oblique cutting," *CIRP International Workshop on Modeling Machining Operations,*2006.

[17] K. Risbood, U. Dixit, and A. Sahasrabudhe, "Prediction of surface roughness and dimensional deviation by measuring cutting forces and vibrations in turning process," *Journal of Materials Processing Technology,* vol. 132, pp. 203-214, 2003.

[18] W. Lin, B. Lee, and C. Wu, "Modeling the surface roughness and cutting force for turning," *Journal of Materials Processing Technology,* vol. 108, pp. 286-293, 2001.

[19] J. P. Davim, V. Gaitonde, and S. 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.

[20] P. J. Núñez Lopez, J. Simao, J. Arenas, and C. De la Cruz, "Surface roughness characterisation using cutting force analysis, regression and neural network prediction models," 2006, pp. 211-216.

[21] B. Suksawat, "Chip form classification and main cutting force prediction of cast nylon in turning operation using artificial neural network," 2010, pp. 172-175.

[22] S. Zare Chavoshi and M. Tajdari, "Surface roughness modelling in hard turning operation of AISI 4140 using CBN cutting tool," *International Journal of Material Forming,* vol. 3, pp. 233-239, 2010.

[23] E. D. Kirby, J. C. Chen, and J. Z. Zhang, "Development of a fuzzy-nets-based in-process surface roughness adaptive control system in turning operations," *Expert Systems with Applications,* vol. 30, pp. 592-604, 2006.

[24] C. Natarajan, S. Muthu, and P. Karuppuswamy, "Prediction and analysis of surface roughness characteristics of a non-ferrous material using ANN in CNC turning," *The International Journal of Advanced Manufacturing Technology,* pp. 1-9, 2011.

[25] V. S. Sharma, S. Dhiman, R. Sehgal, and S. Sharma, "Estimation of cutting forces and surface roughness for hard turning using neural networks," *Journal of Intelligent Manufacturing,* vol. 19, pp. 473-483, 2008.

[26] E. Ezugwu, D. Fadare, J. Bonney, R. Da Silva, and W. Sales, "Modelling the correlation between cutting and process parameters in high-speed machining of Inconel 718 alloy using an artificial neural network," *International Journal of Machine Tools and Manufacture,* vol. 45, pp. 1375-1385, 2005.

[27] H. El-Mounayri, J. F. Briceno, and M. Gadallah, "A new artificial neural network approach to modeling ball-end milling," *The International Journal of Advanced Manufacturing Technology,* vol. 47, pp. 527-534, 2010.

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

[29] T. Özel and Y. Karpat, "Predictive modeling of surface roughness and tool wear in hard turning using regression and neural networks," *International Journal of Machine Tools and Manufacture,* vol. 45, pp. 467-479, 2005.

[30] H. El-Mounayri and H. Deng, "A generic and innovative approach for integrated simulation and optimisation of end milling using solid modelling and neural network," *International Journal of Computer Integrated Manufacturing,* vol. 23, pp. 40-60, 2010.

[31] J. Briceno, “Modeling and optimization of milling using computational intelligence and design of experiments,” M.S. Thesis, Dept. of Mechanical Engineering, IUPUI, 2001.

[32] V. Tandon, H. El-Mounayri, and H. Kishawy, "NC end milling optimization using evolutionary computation," *International Journal of Machine Tools and Manufacture,* vol. 42, pp. 595-605, 2002.

[33] C. T. O. N. Su and M. U. C. Chen, "Computer-aided optimization of multi-pass turning operations for continuous forms on CNC lathes," *IIE transactions,* vol. 31, pp. 583-596, 1999.

[34] C. J. Tzeng and Y. K. Yang, "Determination of optimal parameters for SKD11 CNC turning process," *Materials and Manufacturing Processes,* vol. 23, pp. 363-368, 2008.

[35] K. Palanikumar, L. Karunamoorthy, and R. Karthikeyan, "Optimizing the machining parameters for minimum surface roughness in turning of GFRP composites using the design of experiments," *Cailiao Kexue Yu Jishu(J. Mater. Sci. Technol.)(China),* vol. 20, pp. 373-378, 2004.

[36] P. Benardos and G. Vosniakos, "Prediction of surface roughness in CNC face milling using neural networks and Taguchi's design of experiments," *Robotics and Computer-Integrated Manufacturing,* vol. 18, pp. 343-354, 2002.

[37] A. Hasçalık and U. Çaydaş, "Optimization of turning parameters for surface roughness and tool life based on the Taguchi method," *The International Journal of Advanced Manufacturing Technology,* vol. 38, pp. 896-903, 2008.

[38] J. Z. Zhang and J. C. Chen, "Surface roughness optimization in a drilling operation using the Taguchi design method," *Materials and Manufacturing Processes,* vol. 24, pp. 459-467, 2009.

[39] I. M. Dagwa, "Surface Roughness Optimization of some Machining Parameters in Turning Operations Using Taguchi Method," *Advanced Materials Research,* vol. 62, pp. 613-620, 2009.

[40] W. Yang and Y. Tarng, "Design optimization of cutting parameters for turning operations based on the Taguchi method," *Journal of Materials Processing Technology,* vol. 84, pp. 122-129, 1998.

[41] E. D. Kirby, Z. Zhang, J. C. Chen, and J. Chen, "Optimizing surface finish in a turning operation using the Taguchi parameter design method," *The International Journal of Advanced Manufacturing Technology,* vol. 30, pp. 1021-1029, 2006.

[42] J. Kopec et al, “Optimum machining parameters for achieving desired surface roughness in fine turning of cold pre-formed steel workpiece," *The International Journal of Machining tools and Manufacturers,* pp. 707-716, 2002.

[43] I. Jawahir and X. Wang, "Development of hybrid predictive models and optimization techniques for machining operations," *Journal of Materials Processing Technology,* vol. 185, pp. 46-59, 2007.

[44] S. Satishkumar et al., “Optimization of depth of cut for multi –pass turning using nontraditional techniques,” *Int. Journal of Adv. Manuf. Technology,* pp. 230-238, 2006.

[45] X. Hu and R. Eberhart, "Tracking dynamic system with PSO," *Proceeding of workshop on PSO,* Purdue School of Engineering and Technology, Indianapolis, 2001.

[46] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," vol. 1, pp, 94-100, 2001.

[47] J. Chen and B. Huang, "An in-process neural network-based surface roughness prediction (INN-SRP) system using a dynamometer in end milling operations," *The International Journal of Advanced Manufacturing Technology,* vol. 21, pp. 339-347, 2003.

[48] V. Tandon, “Closing the gap between CAD/CAM and end milling,” M.S. Thesis, Dept. of Mechanical Engineering, IUPUI, 2000.

[49] G. W. Johnson, *LabVIEW graphical programming: practical applications in instrumentation and control*: McGraw-Hill School Education Group, 1997.

[50] J. Ghani, M. Rizal, M. Nuawi, M. Ghazali, and C. Haron, "Monitoring online cutting tool wear using low-cost technique and user-friendly GUI," *Wear,* vol. 271, pp. 2619-2624, 2011.

[51] Y. Karpat and T. Özel, "Multi-objective optimization for turning processes using neural network modeling and dynamic-neighborhood particle swarm optimization," *The International Journal of Advanced Manufacturing Technology,* vol. 35, pp. 234-247, 2007.

[52] L. B. Zhang, C. G. Zhou, X. Liu, Z. Ma, M. Ma, and Y. Liang, "Solving multi objective optimization problems using particle swarm optimization," 2003, pp. 2400-2405 Vol. 4.

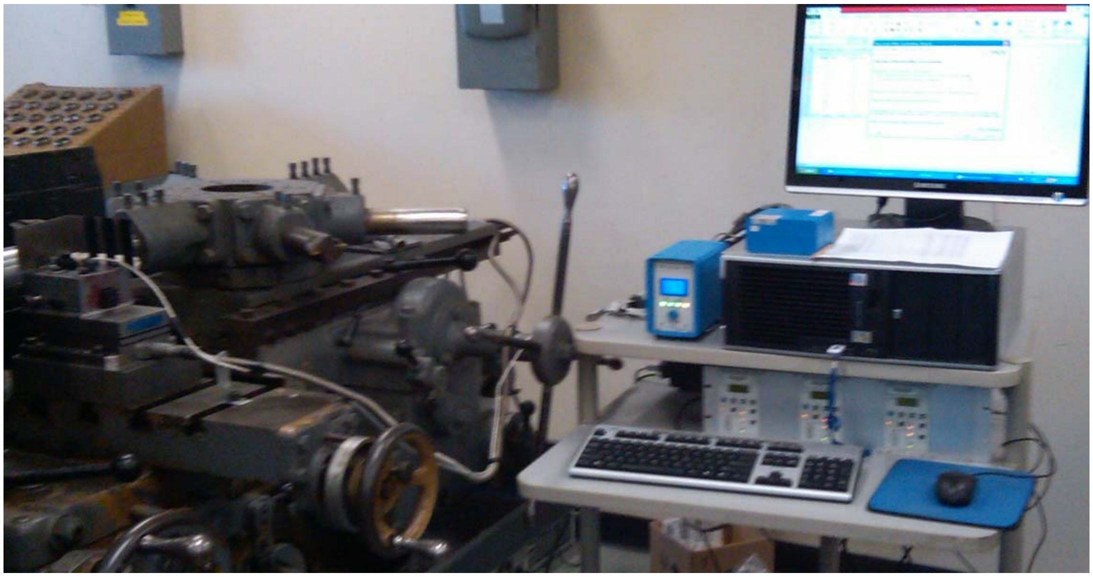
[53] X. Hu and R. Eberhart, "Solving constrained nonlinear optimization problems with particle swarm optimization," pp. 203-206, 2002.

[54] R. C. Eberhart and Y. Shi, *Computational Intelligence: Concepts to Implementations*: Morgan Kaufmann, 2007.

APPENDICES

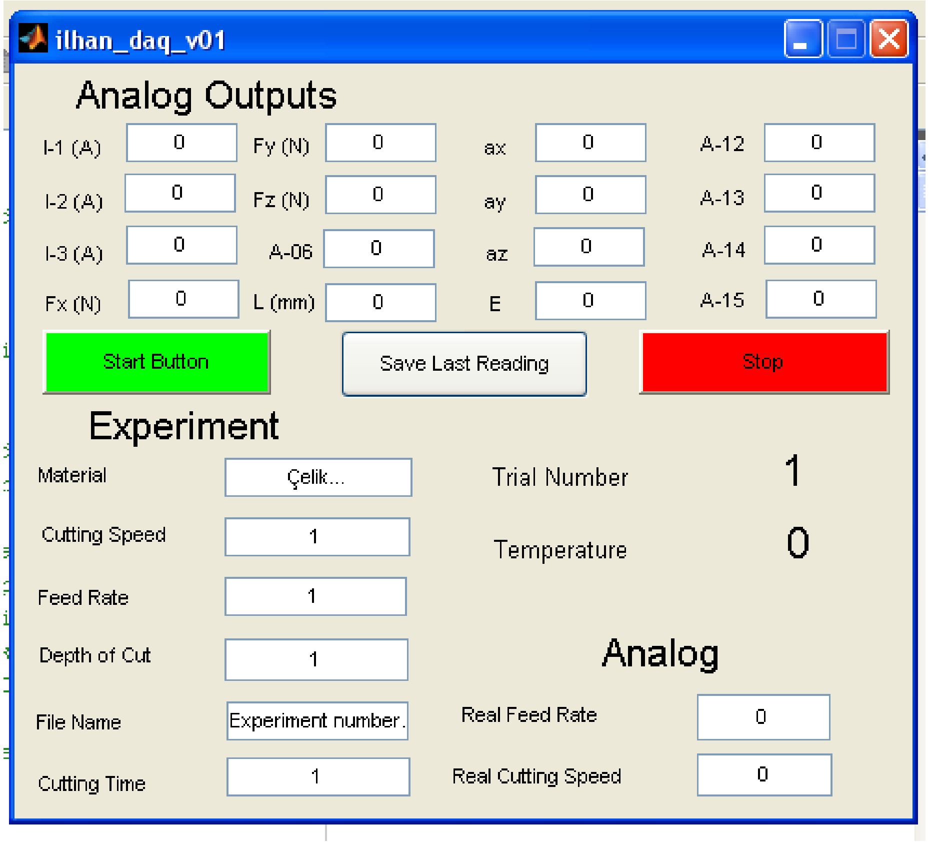
Appendix A

A.1 Figure Experimental Setup



A.2 Figure of surface roughness

A.3 Matlab GUI interface



ilhan_daq_v01

Analog Outputs

I-1 (A)	0	Fy (N)	0	ax	0	A-12	0
I-2 (A)	0	Fz (N)	0	ay	0	A-13	0
I-3 (A)	0	A-06	0	az	0	A-14	0
Fx (N)	0	L (mm)	0	E	0	A-15	0

Experiment

Material	Çelik...	Trial Number	1
Cutting Speed	1	Temperature	0
Feed Rate	1		
Depth of Cut	1		
File Name	Experiment number.	Real Feed Rate	0
Cutting Time	1	Real Cutting Speed	0

Analog

Appendix B

Data used for ANN model training

B.1 Data obtained by using TNMG160408FF Insert (Total 64 Experiments).

V [m/min]	f [mm/rev]	a [mm]	Fr [N]	Ra [mic]
104	0.127	0.254	243.2984	1.535
149	0.127	0.254	245.3579	1.53
211	0.127	0.254	247.1539	1.6
305	0.127	0.254	260.0561	1.46
104	0.1778	0.254	297.8278	1.505
149	0.1778	0.254	290.2299	1.11
211	0.1778	0.254	263.4923	1.205
305	0.1778	0.254	294.5171	1.105
104	0.381	0.254	434.7561	5.805
149	0.381	0.254	402.1734	6.185
211	0.381	0.254	376.5565	6.375
305	0.381	0.254	389.3196	6.08
104	0.5588	0.254	400.6619	13.195
149	0.5588	0.254	388.8305	13.015
211	0.5588	0.254	516.9701	11.84
305	0.5588	0.254	452.4397	10.75
104	0.127	0.762	723.0048	1.34
149	0.127	0.762	698.1973	0.94
211	0.127	0.762	484.0585	0.8
305	0.127	0.762	565.3121	0.88
104	0.1778	0.762	989.4154	0.86
149	0.1778	0.762	570.9566	1.505
211	0.1778	0.762	658.3087	1.45
305	0.1778	0.762	658.3087	1.45
104	0.381	0.762	717.9953	6.045
149	0.381	0.762	678.1831	6.09
211	0.381	0.762	698.0793	6.07
305	0.381	0.762	878.6915	6.005
104	0.5588	0.762	1130.169	10.415
149	0.5588	0.762	899.7007	11.48

211	0.5588	0.762	1061.591	11.99
305	0.5588	0.762	1024.9	12.29
104	0.127	1.27	648.7546	2.05
149	0.127	1.27	617.1317	1.08
211	0.127	1.27	864.9301	1.21
305	0.127	1.27	887.4968	1.04
104	0.1778	1.27	849.6834	1.635
149	0.1778	1.27	867.0112	1.465
211	0.1778	1.27	837.5391	1.43
305	0.1778	1.27	817.9469	1.59
104	0.381	1.27	1536.711	5.965
149	0.381	1.27	1514.56	6.065
211	0.381	1.27	1436.408	6.09
305	0.381	1.27	1359.68	6.09
104	0.5588	1.27	2309.101	8.04
149	0.5588	1.27	2293.674	8.5
211	0.5588	1.27	1910.68	8.715
305	0.5588	1.27	2126.174	8.76
104	0.127	1.778	975.4165	1.815
149	0.127	1.778	916.5075	1.05
211	0.127	1.778	858.7402	0.88
305	0.127	1.778	841.8924	0.915
104	0.1778	1.778	1384.94	1.825
149	0.1778	1.778	1232.859	1.845
211	0.1778	1.778	1193.418	1.83
305	0.1778	1.778	1362.437	1.81
104	0.381	1.778	2317.2	6.08
149	0.381	1.778	2234.191	5.845
211	0.381	1.778	2106.505	5.62
305	0.381	1.778	2100.3	5.755
104	0.5588	1.778	2515.127	10.745
149	0.5588	1.778	2493.63	10.685
211	0.5588	1.778	2393.944	10.735
305	0.5588	1.778	2360.337	10.825

B.2 Data obtained by using T TNMG160408FN Insert (Total 64 Experiments).

V [m/min]	f [mm/rev]	a [mm]	Fr [N]	Ra [mic]
104	0.127	0.254	176.4976	0.915
149	0.127	0.254	184.06	0.905
211	0.127	0.254	203.6922	1.02
305	0.127	0.254	208.8404	1.005
104	0.1778	0.254	225.1976	1.325
149	0.1778	0.254	225.0013	1.455
211	0.1778	0.254	229.2697	1.385
305	0.1778	0.254	232.3822	1.365
104	0.381	0.254	251.4133	6.365
149	0.381	0.254	231.81	6.255
211	0.381	0.254	233.1164	6.185
305	0.381	0.254	237.0063	6.355
104	0.5588	0.254	418.8359	10.835
149	0.5588	0.254	417.2522	10.675
211	0.5588	0.254	456.064	10.875
305	0.5588	0.254	498.6787	10.775
104	0.127	0.762	404.1575	1.05
149	0.127	0.762	371.813	1.265
211	0.127	0.762	376.433	1.305
305	0.127	0.762	389.4393	1.285
104	0.1778	0.762	471.3972	1.575
149	0.1778	0.762	455.9982	1.605
211	0.1778	0.762	449.765	1.745
305	0.1778	0.762	446.5688	1.76
104	0.381	0.762	475.3344	5.985
149	0.381	0.762	765.6536	5.955
211	0.381	0.762	751.8645	6.035
305	0.381	0.762	751.3596	6.185
104	0.5588	0.762	1131.43	9.615
149	0.5588	0.762	1157.544	9.69
211	0.5588	0.762	1116.691	9.775
305	0.5588	0.762	1070.133	10.32
104	0.127	1.27	563.5931	1.055
149	0.127	1.27	544.8958	0.995
211	0.127	1.27	547.8406	1.025

305	0.127	1.27	542.352	1.03
104	0.1778	1.27	759.1936	1.635
149	0.1778	1.27	756.3223	1.54
211	0.1778	1.27	764.7327	1.58
305	0.1778	1.27	762.0848	1.58
104	0.381	1.27	1240.515	6.295
149	0.381	1.27	1198.06	6.22
211	0.381	1.27	1136.2	6.235
305	0.381	1.27	1155.712	6.265
104	0.5588	1.27	1885.155	13.13
149	0.5588	1.27	1770.551	12.93
211	0.5588	1.27	1739.753	14.715
305	0.5588	1.27	1720.182	13.275
104	0.127	1.778	826.1699	1.285
149	0.127	1.778	860.3836	1.48
211	0.127	1.778	804.7218	1.54
305	0.127	1.778	812.6658	1.245
104	0.1778	1.778	1034.456	1.83
149	0.1778	1.778	1004.354	1.7
211	0.1778	1.778	1009.088	1.74
305	0.1778	1.778	992.9036	1.815
104	0.381	1.778	2031.139	5.89
149	0.381	1.778	2086.126	5.95
211	0.381	1.778	2004.061	6.045
305	0.381	1.778	2028.425	6.125
104	0.5588	1.778	2234.978	10.635
149	0.5588	1.778	2124.658	10.68

B.3 Data obtained by using TNMG160408MN Insert (Total 64 Experiments).

V	f	a		Ra
[m/min]	[mm/rev]	[mm]	Fr [N]	[mic]
104	0.127	0.254	232.7851	0.98
149	0.127	0.254	215.0054	0.99
211	0.127	0.254	204.5181	1.14
305	0.127	0.254	205.1765	1.215
104	0.1778	0.254	261.7645	1.61
149	0.1778	0.254	256.9448	1.74
211	0.1778	0.254	261.3797	1.785
305	0.1778	0.254	273.4742	1.905
104	0.381	0.254	492.0043	6.075
149	0.381	0.254	486.4665	6.035
211	0.381	0.254	493.4933	5.995
305	0.381	0.254	503.1991	6.12
104	0.5588	0.254	580.1256	10.555
149	0.5588	0.254	557.9881	10.66
211	0.5588	0.254	551.404	10.795
305	0.5588	0.254	554.0336	10.645
104	0.127	0.762	476.6533	0.77
149	0.127	0.762	482.983	0.92
211	0.127	0.762	492.4869	0.89
305	0.127	0.762	494.4332	0.915
104	0.1778	0.762	718.1974	1.44
149	0.1778	0.762	617.4347	1.16
211	0.1778	0.762	602.948	1.17
305	0.1778	0.762	607.0305	1.195
104	0.381	0.762	1116.749	6.005
149	0.381	0.762	725.3309	2.82
211	0.381	0.762	831.7228	3.065
305	0.381	0.762	889.9171	2.985
104	0.5588	0.762	1247.143	13.69
149	0.5588	0.762	1391.601	10.845
211	0.5588	0.762	1384.344	10.66
305	0.5588	0.762	1415.657	10.66
104	0.127	1.27	727.0507	2.905
149	0.127	1.27	735.0775	2.005
211	0.127	1.27	740.0475	1.4

305	0.127	1.27	582.5965	0.94
104	0.1778	1.27	714.9104	1.525
149	0.1778	1.27	849.5926	1.5
211	0.1778	1.27	873.2018	1.53
305	0.1778	1.27	833.6926	1.53
104	0.381	1.27	1038.294	5.985
149	0.381	1.27	1349.799	5.975
211	0.381	1.27	1348.344	6.09
305	0.381	1.27	1357.224	6.03
104	0.5588	1.27	1502.186	10.69
149	0.5588	1.27	1823.172	10.685
211	0.5588	1.27	1864.331	10.605
305	0.5588	1.27	1897.419	10.515
104	0.127	1.778	824.2799	1.325
149	0.127	1.778	960.7853	1.265
211	0.127	1.778	973.0137	1.07
305	0.127	1.778	1063.346	1.155
104	0.1778	1.778	1333.019	1.545
149	0.1778	1.778	1079.412	1.89
211	0.1778	1.778	1068.126	1.79
305	0.1778	1.778	1078.206	1.845
104	0.381	1.778	2290.555	6.08
149	0.381	1.778	1872.895	6.22
211	0.381	1.778	1918.959	6.365
305	0.381	1.778	1846.432	6.555
104	0.5588	1.778	2299.973	13.46
149	0.5588	1.778	2247.833	10.775
211	0.5588	1.778	2255.602	10.92
305	0.5588	1.778	2273.899	10.76

B.4 Data obtained by using TNMG160408RN Insert (Total 64 Experiments).

V [m/min]	f [mm/rev]	a [mm]	Fr [N]	Ra [mic]
104	0.127	0.254	256.4234	7.115
149	0.127	0.254	249.425	2.915
211	0.127	0.254	258.9729	1.285
305	0.127	0.254	269.3727	0.965
104	0.1778	0.254	301.9586	1.705
149	0.1778	0.254	295.774	1.705
211	0.1778	0.254	299.2893	1.705
305	0.1778	0.254	327.1704	1.595
104	0.381	0.254	454.6752	5.85
149	0.381	0.254	474.9194	6.09
211	0.381	0.254	475.0501	6.01
305	0.381	0.254	337.5326	5.655
104	0.5588	0.254	462.484	10.29
149	0.5588	0.254	448.6837	10.775
211	0.5588	0.254	419.4318	10.745
305	0.5588	0.254	322.1354	5.92
104	0.127	0.762	518.614	1.1
149	0.127	0.762	535.4864	1.27
211	0.127	0.762	519.0662	0.975
305	0.127	0.762	560.2053	2.995
104	0.1778	0.762	639.1152	1.765
149	0.1778	0.762	563.2618	1.755
211	0.1778	0.762	562.941	1.74
305	0.1778	0.762	584.2943	1.685
104	0.381	0.762	1096.362	6.095
149	0.381	0.762	969.197	6
211	0.381	0.762	946.4824	6.15
305	0.381	0.762	928.4021	6.085
104	0.5588	0.762	1198.476	12.8
149	0.5588	0.762	1536.67	12.7
211	0.5588	0.762	1479.314	12.625
305	0.5588	0.762	1550.05	9.68
104	0.127	1.27	678.2507	1.425
149	0.127	1.27	658.8935	1.345
211	0.127	1.27	670.5301	1.19

305	0.127	1.27	675.5719	1.265
104	0.1778	1.27	878.3353	2.2
149	0.1778	1.27	876.6171	1.36
211	0.1778	1.27	874.7851	1.44
305	0.1778	1.27	823.9774	2.01
104	0.381	1.27	1411.397	5.865
149	0.381	1.27	1395.535	5.935
211	0.381	1.27	1276.464	6.085
305	0.381	1.27	1137.226	6.005
104	0.5588	1.27	1626.739	10.465
149	0.5588	1.27	1861.853	10.53
211	0.5588	1.27	1786.142	10.525
305	0.5588	1.27	1741.143	11.57
104	0.127	1.778	701.5861	1.355
149	0.127	1.778	685.5452	2.89
211	0.127	1.778	692.527	2.335
305	0.127	1.778	648.3117	0.91
104	0.1778	1.778	943.2886	1.665
149	0.1778	1.778	916.0447	1.55
211	0.1778	1.778	500.327	1.565
305	0.1778	1.778	886.1075	1.705
104	0.381	1.778	1408.485	6.215
149	0.381	1.778	1356.748	6.11
211	0.381	1.778	1314.332	6.215
305	0.381	1.778	1328.579	5.96
104	0.5588	1.778	2219.637	10.595
149	0.5588	1.778	2139.528	10.7
211	0.5588	1.778	2024.766	10.805
305	0.5588	1.778	1873.86	12.77

B.5 Data obtained by using TNMG160408RP Insert (Total 64 Experiments).

V [m/min]	f [mm/rev]	a [mm]	Fr	Ra [mic]
104	0.127	0.254	198.2213	1.875
149	0.127	0.254	541.6884	1.475
211	0.127	0.254	546.8546	1.71
305	0.127	0.254	545.1861	1.415
104	0.1778	0.254	871.2861	1.7
149	0.1778	0.254	850.0405	1.685
211	0.1778	0.254	933.9917	1.665
305	0.1778	0.254	940.2587	1.655
104	0.381	0.254	307.3643	5.95
149	0.381	0.254	299.1693	5.86
211	0.381	0.254	272.8894	5.865
305	0.381	0.254	263.985	5.73
104	0.5588	0.254	361.1175	10.655
149	0.5588	0.254	349.8009	10.105
211	0.5588	0.254	362.9384	12.74
305	0.5588	0.254	388.2512	12.875
104	0.127	0.762	459.3533	1.145
149	0.127	0.762	449.0858	1.53
211	0.127	0.762	453.1013	1.385
305	0.127	0.762	475.3982	1.25
104	0.1778	0.762	576.0125	1.975
149	0.1778	0.762	549.3174	1.985
211	0.1778	0.762	514.177	1.6
305	0.1778	0.762	529.216	1.48
104	0.381	0.762	899.3617	6.71
149	0.381	0.762	829.4624	6.715
211	0.381	0.762	810.264	6.86
305	0.381	0.762	825.3543	6.535
104	0.5588	0.762	1226.68	13.07
149	0.5588	0.762	1200.747	12.075
211	0.5588	0.762	1131.292	12.015
305	0.5588	0.762	1201.543	11.83
104	0.127	1.27	738.8621	4.33
149	0.127	1.27	696.7606	4.855
211	0.127	1.27	697.0091	3.69
305	0.127	1.27	682.3967	5

104	0.1778	1.27	886.6719	1.815
149	0.1778	1.27	855.2141	1.8
211	0.1778	1.27	872.3791	1.82
305	0.1778	1.27	873.4873	2.265
104	0.381	1.27	1405.721	6.445
149	0.381	1.27	1354.527	6.1
211	0.381	1.27	1335.515	5.985
305	0.381	1.27	1347.431	6.01
104	0.5588	1.27	1736.427	10.61
149	0.5588	1.27	1680.138	10.48
211	0.5588	1.27	1642.456	10.12
305	0.5588	1.27	1287.062	10.725
104	0.127	1.778	893.3393	1.4
149	0.127	1.778	869.5298	1.045
211	0.127	1.778	835.6484	1.07
305	0.127	1.778	826.5578	0.875
104	0.1778	1.778	1127.843	1.67
149	0.1778	1.778	1060.842	1.7
211	0.1778	1.778	1068.542	1.62
305	0.1778	1.778	1156.235	1.63
104	0.381	1.778	1918.044	5.845
149	0.381	1.778	1810.85	5.995
211	0.381	1.778	1741.729	6.055
305	0.381	1.778	1793.464	6.13
104	0.5588	1.778	2259.596	10.425
149	0.5588	1.778	2210.421	10.435
211	0.5588	1.778	2074.712	10.285
305	0.5588	1.778	2115.9	12.91

Appendix C

C.1 Constraint test for PSO code

```
function unfeas=Constraint_test(feed,spd,DOC,net)
unfeas=1;
if feed < 0.127 || feed > 0.5588
    unfeas=0;
    return
end
if spd < 104 || spd > 305
    unfeas=0;
    return
end
if DOC < 0.254 || DOC > 1.778
    unfeas=0;
    return
end
temp=sim(net,[feed spd DOC]');
Force=temp(2);
if Force >= 2515.127 || Force < 0
    unfeas=0;
    return
end
end
```

C.2 PSO code for three dimensional optimization model for turning

```
%Initialization of PSO parameters
%wmax=0.9; % Max weights used in PSO Eqn
%wmin=0.4; % Min weights used in PSO Eqn
%itmax=50; %Maximum iteration number
% Input: Feed, Speed, DOC(Depth of Cut)
% Output: Cutting force, Surface Rough
clear all
clc
load myNet
%net=network1;
c1=1.49445;
c2=1.49445;
epslon=0.00009;

correct(1)=1;
%T_dmd=55;
%W_dmd=300;
%Prev_SOC=0.98;
%Forming an array of weights which would be used during iteration and
%distributing  it equally according to the iteration numbers just as an
```

```
%line so that weights are changing at fixed rate during entire
iteration

%for iter=1:itmax
%W(iter)=wmax-((wmax-wmin)/itmax)*iter;
%end

%**********************************************************

%Initialization of positions of agents
% agents are initialized between -5,+5 randomly
a_feed=0.127;
b_feed=0.5588;
a_spd=104;
b_spd=305;
a_DOC=0.254;
b_DOC=1.778;
N=20; % The number of particles
D=3;  % The dimension of the space of the particles




for init_i=1:N
    x(init_i,1)=a_feed+(b_feed-a_feed)*rand(1,1,1); %feed
    x(init_i,2)=a_spd+(b_spd-a_spd)*rand(1,1,1); %spd
    x(init_i,3)=a_DOC+(b_DOC-a_DOC)*rand(1,1,1); %DOC
end
%Initialization of velocities of agents
%Between -5 , +5, (which can also be started from zero)

m=0;
n=1;
V(:,1)=m+(n-m)*rand(N,1,1);
V(:,2)=m+(n-m)*rand(N,1,1);
V(:,3)=m+(n-m)*rand(N,1,1);
%Evaluating the Objective function for each particle using its position
for i=1:N;
    temp = sim(net,[x(i,2) x(i,1) x(i,3)]');
    F(i,1,1) =temp(1);
%
F(i,1,1)=Objective_function(x(i,1,1),x(i,2,1),T_dmd,W_dmd,Prev_SOC);
end

%**********************************************************
%Obtaining the first minimum of all the randomly defined particles

[C,I]=min((F(:,1,1)));   % To identify minimum in that group" C" and
its position in that column" I"

B(1,1,1)=C;  % Value of Global min for that iteration
XX(1,1,1)=I; % store corresponding inputs of respective global minimum

%Storing that position values in the gbest array
```

```
gbest(1,1,1)=x(I,1,1);
gbest(1,2,1)=x(I,2,1);
gbest(1,3,1)=x(I,3,1);

%********************************************************
%Matrix composed of gbest vector

%Assinging the global best position value to each particle in its own
%storing location provided by array G
for p=1:N
    for r=1:D
        G(p,r,1)=gbest(1,r,1);
    end
end

%%%Finding the obj value corresponding to the gbest
temp = sim(net,[G(1,2,1) G(1,1,1) G(1,3,1)]');
Fbest(1,1,1)=temp(1);
%
Fbest(1,1,1)=Objective_function(G(1,1,1),G(1,2,1),T_dmd,W_dmd,Prev_SOC)
;

%Initializing pbest variable for all the particles  which is the
particles
%best position
  for i=1:N;
       pbest(i,:,1)=x(i,:,1);
  end

W(1)=0.5+(rand(1)/2);

%The new velocities of all the particles
V(:,:,2)=W(1)*V(:,:,1)+c1*rand*(pbest(:,:,1)-
x(:,:,1))+c2*rand*(G(:,:,1)-x(:,:,1));

%The new positions of all the particles
x(:,:,2)=x(:,:,1)+V(:,:,2);

%
temp=Objective_function(gbest(1,1,1),gbest(1,2,1),T_dmd,W_dmd,Prev_SOC)
;
temp = sim(net,[gbest(1,2,1) gbest(1,1,1) gbest(1,3,1)]');
Fb(1,1,1)=temp(1);
%sim('PSO_Code_Lookupsim_3',[0,0.001])
%Fb(1,1,1)=temp_3;
%**********************************************************************
****
%**********************************************************************
****
%The main loop for the Optimization STARTING
%**********************************************************************
****
%**********************************************************************
****
```

```
j=2;


ending=0;
while ending==0

W(j)=0.5+(rand(1)/2);
% Calculation of new positions

%Finding the new objective function values for all these new positions
of the particles
for i=1:N;
    temp = sim(net,[x(i,2,j) x(i,1,j) x(i,3,j)]');
    F(i,1,j)=temp(1);
    %
F(i,1,j)=Objective_function(x(i,1,j),x(i,2,j),T_dmd,W_dmd,Prev_SOC);
end

% Finding the minimum value for all the particles
%  [C,I]=min(abs(F(:,:,j))); %%CONSTRAINED PROBLEM CHANGES HERE
%  B(1,1,j)=C;
%  XX(1,1,j)=I;
% gbest(1,1,1)=x(I,1,1);
% gbest(1,2,1)=x(I,2,1);
% gbest(1,3,1)=x(I,3,1);
%
%***********************************************************************
****
%Constrained part modification for Feasibility Starts
%
%***********************************************************************
****
[F_g_sort,sort_g_index]=sort(F(:,1,j));
Feas_g=0;
i_unconst_g=1;
while Feas_g==0 andand i_unconst_g<=N

Feas_g=Constraint_test(x(sort_g_index(i_unconst_g),1,j),x(sort_g_index(
i_unconst_g),2,j), x(sort_g_index(i_unconst_g),3,j),net);
    i_unconst_g=i_unconst_g+1;
end
if i_unconst_g==j+1 || Feas_g==0
    C=Fbest(1,1,j-1);
    B(1,1,j)=C;
    gbest(1,1,j)=G(1,1,j-1);
    gbest(1,2,j)=G(1,2,j-1);
    gbest(1,3,j)=G(1,3,j-1);
else
    C=F(sort_g_index(i_unconst_g-1),1,j);
    I=sort_g_index(i_unconst_g-1);
   %Saving the best minimum value of objective function in an array for
record
    B(1,1,j)=C;
```

```
%Defining the new global best position afther this iteration which is
used
%for new iteration
    gbest(1,1,j)=x(I,1,j);
    gbest(1,2,j)=x(I,2,j);
    gbest(1,3,j)=x(I,3,j);
end
%
%**********************************************************************
****
% %Constrained part modification for Feasibility ends
%
%**********************************************************************
****
% %Finding the best objective function value corresponding to this new
global best
temp = sim(net,[gbest(1,2,j) gbest(1,1,j) gbest(1,3,j)]');
Fb(1,1,j)=temp(1);
%
Fb(1,1,j)=Objective_function(gbest(1,1,j),gbest(1,2,j),T_dmd,W_dmd,Prev
_SOC);
%Fb(1,1,j)=C;
% %sim('PSO_Code_Lookupsim_5',[0,0.001])
% %Fb(1,1,j)=temp_5;
%
% %Comparing this new best objective function corresponding to global
best
% %with the previous best objective function values corresponding to
other
% %global bests
%
%
[C,I]=min(Fb(1,1,:));

%If the new Fb(1,1,j) (corresponding to new gbest) is less then or
equal to
%this C then update the gbest accordingly
%(the less than and equal to sign might be used to sove the matlab
accuracy problem)
if Fb(1,1,j)<=C
    gbest(1,1,j)=gbest(1,1,j);
    gbest(1,2,j)=gbest(1,2,j);
    gbest(1,3,j)=gbest(1,3,j);
else
    gbest(1,1,j)=gbest(1,1,I);
    gbest(1,2,j)=gbest(1,2,I);
    gbest(1,3,j)=gbest(1,3,I);
end
%Matrix composed of gbest vector
%So now defining the new set of G variable corresponding to this
iteration
%and would be used to calculate the new velocities and positions
for p=1:N
    for r=1:D
        G(p,r,j)=gbest(1,r,j);
    end
end
%Dont know why its defined becuase it is not used anywhere

%
Fbest(1,1,j)=Objective_function(G(1,1,j),G(1,2,j),T_dmd,W_dmd,Prev_SOC)
;
temp = sim(net,[G(1,2,j) G(1,1,j) G(1,3,j)]');
Fbest(1,1,j)=temp(1);
%sim('PSO_Code_Lookupsim_6',[0,0.001])
%Fbest(1,1,j)=temp_6;

 %For each particle comparing the objective function value for its
entire
 %history and finding the particles best and then defining it
accordingly
 %so that it can be used in main optimization equations
  for i=1:N;
      %[C,I]=min(F(i,1,:)); %%CONSTRAINED PROBLEM CHANGES HERE
%**********************************************************************
****
%Constrained part modification for Feasibility Starts
%**********************************************************************
****
    [Fb_p_sort,sort_p_index]=sort(F(i,1,:));
    Feas_p=0;
    i_unconst_p=1;
    while Feas_p==0 andand i_unconst_p<=j

Feas_p=Constraint_test(x(i,1,sort_p_index(i_unconst_p)),x(i,2,sort_p_in
dex(i_unconst_p)), x(i,3,sort_p_index(i_unconst_p)),net);
       i_unconst_p=i_unconst_p+1;
    end

    if i_unconst_p==j+1 || Feas_p==0
        pbest(i,:,j)=pbest(i,:,j-1);
    else
        %C=Objective_function(pbest(i,1,j-1),pbest(i,2,j-
1),T_dmd,W_dmd,Prev_SOC);
        C=F(i,1,sort_p_index(i_unconst_p-1));
        I=sort_p_index(i_unconst_p-1);
        if F(i,1,j)<=C andand i_unconst_p-1==1
            pbest(i,:,j)=x(i,:,I);
        else
            pbest(i,:,j)=x(i,:,I);
        end
       end
  end
%**********************************************************************
****
%Constrained part modification for Feasibility ends
%**********************************************************************
****
%**********************************************************************
****
```

```
%Stopping Criterion Starting
%**********************************************************************
****
  l=1;
  repeat=1;
  while repeat ==1
      if j-l>0
          if abs(Fbest(:,:,j)-Fbest(:,:,j-l))<=epslon
              if l<50
                  l=l+1;
                  repeat=1;
              else
                   Feed=gbest(:,1,j);
                   Spd=gbest(:,2,j);
                   DOC=gbest(:,3,j);
                   ending=1;
                   repeat=0;
                   break
              end
          else
              repeat=0;
          end
      else
          repeat=0;
      end
  end

  if ending==1
      break
  end

%**********************************************************************
****
%Stopping Criterion Ending
%**********************************************************************
****


flag=0;
correct(j)=0;
for i=1:N
    if pbest(i,:,j)==gbest(:,:,j)
        correct(j)=1;
        flag=1;
        break
    end
end


%Calculating the Velocities for the next iteration using the main PSO
%Equation
V(:,:,j+1)=W(j)*V(:,:,j)+c1*rand*(pbest(:,:,j)-
x(:,:,j))+c2*rand*(G(:,:,j)-x(:,:,j));
```

```
%Calculating the next positions of the particle corresponding to th
above
%newly found velocities.

x(:,:,j+1)=x(:,:,j)+V(:,:,j+1);
j=j+1;

end
Feed(1)=gbest(:,1,j-1)
spd(1)=gbest(:,2,j-1)
DOC(1)=gbest(:,3,j-1)
temp=sim(net,[gbest(:,2,j-1) gbest(:,1,j-1) gbest(:,3,j-1)]');
Force(1)=temp(2);
Opt_values=min(Fbest(1,1,j-1))
```

C.3 Code for plotting results for PSO

```
for i=1:size(Fbest,3)
    Surface_rough(i)=Fbest(:,:,i);
    speed_plot(i)=gbest(1,2,i);
    Feed_plot(i)=gbest(1,1,i);
    DOC_plot(i)=gbest(1,3,i);
end



j=150;
Figure(1)
scatter3(x(:,2,j),x(:,1,j),x(:,3,j),500,'b','.')
axis([104 305 0.127 0.5588 0.254 1.778])
grid on
xlabel('speed')
ylabel('Feed')
zlabel('Depth of Cut')

Figure(2)
plot(Surface_rough)
grid on
xlabel('Trials')
ylabel('Surface Roughness')

Figure(3)
plot(speed_plot,Surface_rough)
grid on
xlabel('Speed')
ylabel('Surface Roughness')

Figure(4)
plot(Feed_plot,Surface_rough)
grid on
```

```
xlabel('Feed')
ylabel('Surface Roughness')

Figure(5)
plot(DOC_plot,Surface_rough)
grid on
xlabel('Depth of Cut')
ylabel('Surface Roughness')
```

C.4 PSO code for two dimensional optimization (depth of cut as added constraint)

```
%Initialization of PSO parameters
%wmax=0.9; % Max weights used in PSO Eqn
%wmin=0.4; % Min weights used in PSO Eqn
%itmax=50; %Maximum iteration number
% Input: Feed, Speed, DOC(Depth of Cut)
% Output: Cutting force, Surface Rough
clear all
clc
load myNet
%net=network1;
c1=1.49445;
c2=1.49445;
epslon=0.00009;

correct(1)=1;
%T_dmd=55;
%W_dmd=300;
%Prev_SOC=0.98;
%Forming an array of weights which would be used during iteration and
%distributing  it equally according to the iteration numbers just as an
%line so that weights are changing at fixed rate during entire
iteration

%for iter=1:itmax
%W(iter)=wmax-((wmax-wmin)/itmax)*iter;
%end

%**********************************************************

%Initialization of positions of agents
% agents are initialized between -5,+5 randomly
a_feed=0.127;
b_feed=0.5588;
a_spd=104;
b_spd=305;
a_DOC=0.254;
b_DOC=0.254;
N=20; % The number of particles
D=2;  % The dimension of the space of the particles
```

```
DOC=0.75;



for init_i=1:N
    x(init_i,1)=a_feed+(b_feed-a_feed)*rand(1,1,1); %feed
    x(init_i,2)=a_spd+(b_spd-a_spd)*rand(1,1,1); %spd
 %   x(init_i,3)=a_DOC+(b_DOC-a_DOC)*rand(1,1,1); %DOC
end
%Initialization of velocities of agents
%Between -5 , +5, (which can also be started from zero)

m=0;
n=1;
V(:,1)=m+(n-m)*rand(N,1,1);
V(:,2)=m+(n-m)*rand(N,1,1);
%V(:,3)=m+(n-m)*rand(N,1,1);
%Evaluating the Objective function for each particle using its position
for i=1:N;
    temp = sim(net,[x(i,2) x(i,1) DOC]');
    F(i,1,1) =temp(1);
%
F(i,1,1)=Objective_function(x(i,1,1),x(i,2,1),T_dmd,W_dmd,Prev_SOC);
end

%**********************************************************
%Obtaining the first minimum of all the randomly defined particles

[C,I]=min((F(:,1,1)));   % To identify minimum in that group" C" and
its position in that column" I"

B(1,1,1)=C;  % Value of Global min for that iteration
XX(1,1,1)=I; % store corresponding inputs of respective global minimum

%Storing that position values in the gbest array
gbest(1,1,1)=x(I,1,1);
gbest(1,2,1)=x(I,2,1);
%gbest(1,3,1)=x(I,3,1);

%********************************************************
%Matrix composed of gbest vector

%Assinging the global best position value to each particle in its own
%storing location provided by array G
for p=1:N
    for r=1:D
        G(p,r,1)=gbest(1,r,1);
    end
end

%%%Finding the obj value corresponding to the gbest
temp = sim(net,[G(1,2,1) G(1,1,1) DOC]');
Fbest(1,1,1)=temp(1);
```

```
%
Fbest(1,1,1)=Objective_function(G(1,1,1),G(1,2,1),T_dmd,W_dmd,Prev_SOC)
;

%Initializing pbest variable for all the particles  which is the
particles
%best position
  for i=1:N;
       pbest(i,:,1)=x(i,:,1);
  end

W(1)=0.5+(rand(1)/2);

%The new velocities of all the particles
V(:,:,2)=W(1)*V(:,:,1)+c1*rand*(pbest(:,:,1)-
x(:,:,1))+c2*rand*(G(:,:,1)-x(:,:,1));

%The new positions of all the particles
x(:,:,2)=x(:,:,1)+V(:,:,2);

%
temp=Objective_function(gbest(1,1,1),gbest(1,2,1),T_dmd,W_dmd,Prev_SOC)
;
temp = sim(net,[gbest(1,2,1) gbest(1,1,1) DOC]');
Fb(1,1,1)=temp(1);
%sim('PSO_Code_Lookupsim_3',[0,0.001])
%Fb(1,1,1)=temp_3;
%***********************************************************************
****
%***********************************************************************
****
%The main loop for the Optimization STARTING
%***********************************************************************
****
%***********************************************************************
****
j=2;


ending=0;
while ending==0

W(j)=0.5+(rand(1)/2);
% Calculation of new positions

%Finding the new objective function values for all these new positions
of the particles
for i=1:N;
    temp = sim(net,[x(i,2,j) x(i,1,j) DOC]');
    F(i,1,j)=temp(1);
    %
F(i,1,j)=Objective_function(x(i,1,j),x(i,2,j),T_dmd,W_dmd,Prev_SOC);
end
```

```
% Finding the minimum value for all the particles
%  [C,I]=min(abs(F(:,:,j))); %%CONSTRAINED PROBLEM CHANGES HERE
%  B(1,1,j)=C;
%  XX(1,1,j)=I;
% gbest(1,1,1)=x(I,1,1);
% gbest(1,2,1)=x(I,2,1);
% gbest(1,3,1)=x(I,3,1);
%
%**********************************************************************
****
%Constrained part modification for Feasibility Starts
%
%**********************************************************************
****
[F_g_sort,sort_g_index]=sort(F(:,1,j));
Feas_g=0;
i_unconst_g=1;
while Feas_g==0 andand i_unconst_g<=N

Feas_g=Constraint_test(x(sort_g_index(i_unconst_g),1,j),x(sort_g_index(
i_unconst_g),2,j), DOC,net);
    i_unconst_g=i_unconst_g+1;
end
if i_unconst_g==j+1 || Feas_g==0
    C=Fbest(1,1,j-1);
    B(1,1,j)=C;
    gbest(1,1,j)=G(1,1,j-1);
    gbest(1,2,j)=G(1,2,j-1);
   % gbest(1,3,j)=G(1,3,j-1);
else
    C=F(sort_g_index(i_unconst_g-1),1,j);
    I=sort_g_index(i_unconst_g-1);
   %Saving the best minimum value of objective function in an array for
record
    B(1,1,j)=C;
%Defining the new global best position afther this iteration which is
used
%for new iteration
    gbest(1,1,j)=x(I,1,j);
    gbest(1,2,j)=x(I,2,j);
  %  gbest(1,3,j)=x(I,3,j);
end
%
%**********************************************************************
****
% %Constrained part modification for Feasibility ends
%
%**********************************************************************
****
% %Finding the best objective function value corresponding to this new
global best
temp = sim(net,[gbest(1,2,j) gbest(1,1,j) DOC]');
Fb(1,1,j)=temp(1);
```

```
%
Fb(1,1,j)=Objective_function(gbest(1,1,j),gbest(1,2,j),T_dmd,W_dmd,Prev
_SOC);
%Fb(1,1,j)=C;
% %sim('PSO_Code_Lookupsim_5',[0,0.001])
% %Fb(1,1,j)=temp_5;
%
% %Comparing this new best objective function corresponding to global
best
% %with the previous best objective function values corresponding to
other
% %global bests
%
%
[C,I]=min(Fb(1,1,:));

%If the new Fb(1,1,j) (corresponding to new gbest) is less then or
equal to
%this C then update the gbest accordingly
%(the less than and equal to sign might be used to sove the matlab
accuracy problem)
if Fb(1,1,j)<=C
    gbest(1,1,j)=gbest(1,1,j);
    gbest(1,2,j)=gbest(1,2,j);
 %   gbest(1,3,j)=gbest(1,3,j);
else
    gbest(1,1,j)=gbest(1,1,I);
    gbest(1,2,j)=gbest(1,2,I);
%    gbest(1,3,j)=gbest(1,3,I);
end
%Matrix composed of gbest vector
%So now defining the new set of G variable corresponding to this
iteration
%and would be used to calculate the new velocities and positions
for p=1:N
    for r=1:D
        G(p,r,j)=gbest(1,r,j);
    end
end
%Dont know why its defined becuase it is not used anywhere

%
Fbest(1,1,j)=Objective_function(G(1,1,j),G(1,2,j),T_dmd,W_dmd,Prev_SOC)
;
temp = sim(net,[G(1,2,j) G(1,1,j) DOC]');
Fbest(1,1,j)=temp(1);
%sim('PSO_Code_Lookupsim_6',[0,0.001])
%Fbest(1,1,j)=temp_6;

 %For each particle comparing the objective function value for its
entire
 %history and finding the particles best and then defining it
accordingly
 %so that it can be used in main optimization equations
  for i=1:N;
```

```
      %[C,I]=min(F(i,1,:)); %%CONSTRAINED PROBLEM CHANGES HERE
%***********************************************************************
****
%Constrained part modification for Feasibility Starts
%***********************************************************************
****
    [Fb_p_sort,sort_p_index]=sort(F(i,1,:));
    Feas_p=0;
    i_unconst_p=1;
    while Feas_p==0 andand i_unconst_p<=j

Feas_p=Constraint_test(x(i,1,sort_p_index(i_unconst_p)),x(i,2,sort_p_in
dex(i_unconst_p)), DOC,net);
       i_unconst_p=i_unconst_p+1;
    end

    if i_unconst_p==j+1 || Feas_p==0
        pbest(i,:,j)=pbest(i,:,j-1);
    else
        %C=Objective_function(pbest(i,1,j-1),pbest(i,2,j-
1),T_dmd,W_dmd,Prev_SOC);
        C=F(i,1,sort_p_index(i_unconst_p-1));
        I=sort_p_index(i_unconst_p-1);
        if F(i,1,j)<=C andand i_unconst_p-1==1
            pbest(i,:,j)=x(i,:,I);
        else
            pbest(i,:,j)=x(i,:,I);
        end
       end
  end
%***********************************************************************
****
%Constrained part modification for Feasibility ends
%***********************************************************************
****
%***********************************************************************
****
%Stopping Criterion Starting
%***********************************************************************
****
  l=1;
  repeat=1;
  while repeat ==1
      if j-l>0
          if abs(Fbest(:,:,j)-Fbest(:,:,j-l))<=epslon
              if l<50
                  l=l+1;
                  repeat=1;
              else
                   Feed=gbest(:,1,j);
                   Spd=gbest(:,2,j);
                   %DOC=gbest(:,3,j);
                   ending=1;
                   repeat=0;
                   break
              end
          else
              repeat=0;
          end
      else
          repeat=0;
      end
  end

  if ending==1
      break
  end

%**********************************************************************
****
%Stopping Criterion Ending
%**********************************************************************
****

flag=0;
correct(j)=0;
for i=1:N
    if pbest(i,:,j)==gbest(:,:,j)
        correct(j)=1;
        flag=1;
        break
    end
end


%Calculating the Velocities for the next iteration using the main PSO
%Equation
V(:,:,j+1)=W(j)*V(:,:,j)+c1*rand*(pbest(:,:,j)-
x(:,:,j))+c2*rand*(G(:,:,j)-x(:,:,j));

%Calculating the next positions of the particle corresponding to th
above
%newly found velocities.

x(:,:,j+1)=x(:,:,j)+V(:,:,j+1);
j=j+1;

end
Feed(1)=gbest(:,1,j-1)
spd(1)=gbest(:,2,j-1)
%DOC(1)=gbest(:,3,j-1)
temp=sim(net,[gbest(:,2,j-1) gbest(:,1,j-1) DOC]');
Force(1)=temp(2);
Opt_values=min(Fbest(1,1,j-1))
```

C.5 Code for plotting results of two dimensional PSO study

```
for i=1:size(Fbest,3)
    Surface_rough(i)=Fbest(:,:,i);
    speed_plot(i)=gbest(1,2,i);
    Feed_plot(i)=gbest(1,1,i);
   % DOC_plot(i)=gbest(1,3,i);
end


j=10;
Figure(1)
scatter(x(:,2,j),x(:,1,j),500,'b','.')
axis([104 305 0.127 0.5588])
grid on
xlabel('speed')
ylabel('Feed')


Figure(2)
plot(Surface_rough)
grid on
xlabel('Trials')
ylabel('Surface Roughness')

Figure(3)
plot(speed_plot,Surface_rough)
grid on
xlabel('Speed')
ylabel('Surface Roughness')

Figure(4)
plot(Feed_plot,Surface_rough)
grid on
xlabel('Feed')
ylabel('Surface Roughness')

% Figure(5)
% plot(DOC_plot,Surface_rough)
% grid on
% xlabel('Depth of Cut')
% ylabel('Surface Roughness')
```