International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 2, Issue 4 (February 2013) PP: 62-68

Optimization of Cutting Parameters of Thin Ribs in High Speed Machining

Pal Pandian P.[1], Dr. Prabhu Raja V.[2], Sakthimurugan K.[3]
[1] *Research Scholar, Anna University of Technology, Coimbatore, Tamilnadu, India*
[2] *Associate Professor, Department of Mechanical Engineering, PSG College of Technology, Coimbatore, India*
[3] *Assistant Professor, Department of Mechanical Engineering, M.P. N.M.J Engineering College, Erode*

Abstract: *Machining of thin-walled parts is a key process in aerospace industry. Many components used in the aerospace industry are usually thin-walled structures. Because of their poor stiffness, thin-walled work pieces are very easy to deform under the action of cutting force in the process of cutting. Even in CNC milling, in which the tools are controlled exactly according to the contour of the thin-walled component, the wall will be thicker at the top and thinner at the root. In general, the surface dimensional error is induced mainly by the deflection of the work piece during milling, which does not remove the material as planned. The part deflection caused by the cutting force is difficult to predict and control. The main objective of this work is to achieve the minimum surface dimensional error which decreases the machining time. Therefore, the cutting parameters are to be optimized which enables the minimum possible surface dimensional error. The conditions required to achieve this in high speed milling process imply optimum cutting forces which in turn induce the part deflection. The present work is aimed at predicting cutting forces during machining and obtaining optimum cutting speed and feed rate. An Artificial Neural Network (ANN) predictive model is used to predict cutting forces during machining and Particle Swarm Optimization (PSO) algorithm is used to obtain optimum cutting speed and feed rate. The parts are modeled and effect of cutting force is applied and deflections of the work piece are found out using ANSYS. The ANN models and algorithms are developed using MATLAB.*

Keywords: *HSM, Thin walled parts or thin ribs, ANN, PSO*

I. Introduction

High-Speed Machining (HSM) is an emerging area of technology within manufacturing engineering. High speed machining is a proven stipulation characterized by low cutting forces and high metal removal. High Speed Milling is a technique used in the CNC Milling Industry that combines high spindle speeds with increased feed rates. If the machine tool spindles have rotational speeds as high as 10000-100000 rpm, the machining can be categorized as HSM [1].

Many components used in the aerospace industry are usually thin-walled structures. Because of their poor stiffness, thin-walled work pieces are very easy to deform under the action of cutting force in the process of cutting [2]. In general, the surface dimensional error is induced mainly by the deflection of the work piece during milling, which does not remove the material as planned [3].

Several researches on machining of thin walled parts have been done [4]. None of the paper prescribed the optimum cutting parameters for minimum surface dimensional error. The goal of this paper is to achieve the minimum surface dimensional error for thin rib machining using ANN models and PSO Algorithms.

II. Problem Formulation

The machining sketch of the typical thin-walled work piece (Fig. 1) illustrates the deformation of the thin-walled structure in the machining process. The material ABDC needs to be cut away ideally. However, under the acting of milling force, Point C moves to Point C’ and the Point A moves to Point A’. Therefore, only material A’BDC is cut away in the practical machining process due to the deformation. After the miller moves away from the milling surface, the wall recovers elastically, and material CDC’ that should have been cut away remains unremoved. This causes the shape of the wall to be thicker in its higher part and thinner in its lower part. This is the most common problem in the machining of thin-walled components.

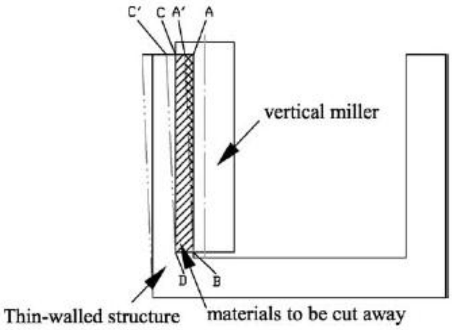

Fig. 1. Machining Sketch Of The Typical Thin-Walled Structure.

III. Cutting Force Modeling Using Ann

The three components of cutting forces are found out experimentally and using ANN models. The force components (Fig. 2) illustrate radial tangential and axial forces. The ANN model is used for analysis and prediction of the relationship between the cutting forces and machining parameters. Input parameters are axial and radial depth of cuts, feed and cutting speed. The output parameters of the model are cutting forces.

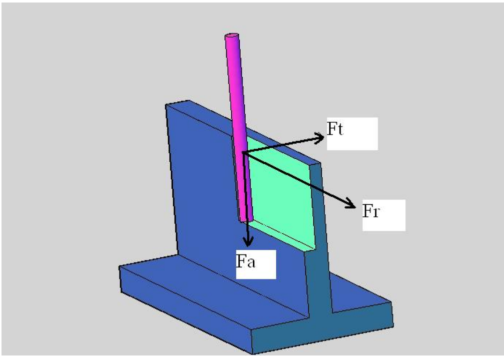

Fig 2. Cutting Forces components in milling Process

3.1 MODEL DESCRIPTION

The model consists of three - layered feed – forward back – propagation neural network. The network is trained with pairs of inputs/outputs datasets generated when high speed milling of Aluminum Alloy. Fig 3. represents a three layered ANN model for cutting forces.

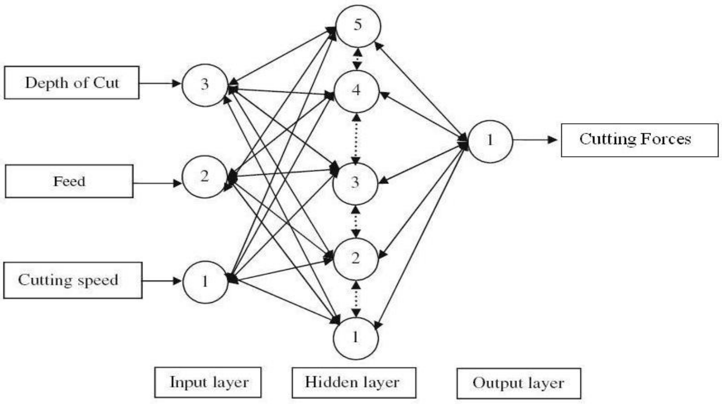

Fig 3. ANN Model for Cutting Forces

3.2 EXPERIMENTATION OF INPUT/OUTPUT DATASET

A series of milling tests were conducted in 3 – Axis vertical CNC milling machine fitted with 5kW high speed spindle, which has a maximum speed of 40,000 rpm. The work piece used was Aluminum Alloy of 6210. The diameter of the end mill, helix angle and radial rake angle are 6mm, 30° and 2° respectively. Before conducting the cutting trials, the work piece have been rough machined.

A Kistler piezo-electric 3– component dynamometer was used for measuring the forces. Teknonix make oscilloscope was used for recording these forces in terms of voltage. According to the center composite design method, Cutting conditions employed in the high speed milling trials are shown in Table 1.

The forces correspond to a cutting velocity of 453 m/min (spindle speed of 24000 rpm using a 6 mm cutter) and for feed rates varying from 1000 mm/min to 2500 mm/min.

Table 1: Cutting Conditions

Factors and levels selected					
Factors	**Levels**				
	-2	**-1**	**0**	**1**	**2**
Vc (m/min)	119.65	150	200	250	280.36
Fz (mm)	0.01	0.02	0.03	0.04	0.05
Ad (mm)	4.8	4.9	5	5.1	5.2
Rd (mm)	0.3	0.4	0.5	0.6	0.7

3.3 PRE-PROCESSING OF INPUT/OUTPUT DATASET

The generalization capability of the neural network is essentially dependent on the selection of the appropriate input/output parameters of the systems, the distribution of the dataset and the format of the presentation of the dataset to the network. Thus before the ANN can be trained and the mapping learnt, it is important to process the experimental data into patterns.

For this model, the input parameters used are the four main cutting parameters including cutting speed, axial depth of cut, radial depth of cut and feed per tooth. The output dataset is cutting force.

In total 32 machining tests were conducted and a total of 32 input/output dataset pairs were collected during machining tests. Prior to the use of the datasets, principal component analysis was performed using the MATLAB subroutine prepca to test the correlation between the input and output dataset. Result shows that the four selected cutting parameters (input dataset) accounts for more than 97% variability in Cutting forces (output dataset).

Before training the network, the input/output datasets were normalized within the range of ±1, using the Matlab subroutine premnmx. Since only a limited number of experiments are representative of the feasible parameter space, it is important that the ANN realizes each set fully. This is achieved by normalizing the data as

$$X = (X_R - X_{\min})\frac{X_{N\max} - X_{N\min}}{X_{\max} - X_{\min}}$$

Where X_R is the real value of the variable before normalization. X_{min} and X_{max} are the minimum and maximum values of the variable X. These are normalized to values X_{Nmin}, X_{Nmax} such that $0 < X_{Nmin} < X_{Nmax} <1$. For example, cutting speed is varied from 119.8 to 280.4 m/min in our experiments. Thus, X_{min} =119.8 and X_{max} =280.4. Then we choose X_{Nmin} to be 0 and X_{Nmax} to be 1. In this way the range 119.8 to 280.4 is now mapped to 0 ~ 1. So when X_R = 200 m/min, X =0.4994.

3.4 NEURAL NETWORK DESIGN AND TRAINING

The network architecture such as number of neurons and layers are very important factors that determine the functionality and generalization capability of the network. For this model, standard multiplayer feed forward back propagation hierarchical neural networks were designed with MATLAB 7.5.0 Neural Network Toolbox. The networks consist of three layers: input layer, hidden layer and output layer. In order to determine the optimal architecture, four different networks with different number of layers and neurons in the hidden layer were designed and tested.

In general, The networks have four neurons in the input layer, corresponding to each of the four cutting parameters and one neuron in the output layer, corresponding to surface roughness. Networks with one or two layers and with 9 or 12 in the hidden layer were used. For all networks linear transfer function 'purelin' and tangent sigmoid transfer function 'tansig' were used in the output and hidden layer, respectively. The networks were trained with Levenberg-Marquardt algorithm. This training algorithm was chosen due to its high accuracy in similar function approximation. In order to improve the generalization of the network, the automatic Bayesian regularization scheme was used in conjunction with the Levenberg-Marquardt algorithm.

For training with the Levenberg-Marquardt combined with Bayesian regularization, the input/output dataset was divided randomly into two categories: training dataset, consisting of two-thirds of the input/output dataset and test dataset, which consists of one third of the data.

IV. Particle Swarm Optimization (PSO)

Particle swarm optimization is a population-based stochastic optimization technique developed by Eberhart and Kennedy in 1995, inspired by the social behavior of bird flocking or fish schooling.

4.1 BASIC OF PARTICLE SWARM OPTIMIZATION

The particle swarm concept originated as a simulation of a simplified social system. The original intent was to graphically simulate the choreography of a bird flock or fish school. However, the particle swarm model can be used as an optimizer. As stated previously, PSO simulates the behaviors of bird flocking. Suppose a group of birds are randomly searching for a food in area. Only one piece of food exists in the area being searched. All the birds do not know where the food is, but they know how far the food is in iteration. So what is best strategy to find the food? The most effective one is to follow the bird nearest to the food.

PSO learns from this scenario and uses it to solve optimization problems. In PSO, each single solution is a “bird” in the search space (we call it a “particle”). All particles have fitness values that are evaluated by the fitness function to be optimized and have velocities that direct the “flying” of the particles. The particles fly through the problem space by following the current optimum particles. PSO initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two “best” values. The first one is the best solution (fitness) it has achieved so far. The fitness value is also stored; this value is called *pbest.* Another “best” value tracked by the particle swarm optimizer is the best value obtained so far by any particle in the population. This best value is a global best and is called *gbest.* When a particle takes part of the population as its topological neighbors, the best value is a local best and is calles *lbest.*

After finding the two best values, the particle updates its velocity and positions. The particle updates its velocity and positions with the following.

$$v[\,] = w\,v[\,] + C_1\,rand(\,)\times(pbest[\,] - present[\,]) + C_2\,rand(\,)\times(gbest[\,] - present[\,]) \quad \text{-----}(i)$$

$$present[\,] = present[\,] + v[\,] \quad \text{-----}(ii)$$

where

v[ ] = particle velocity
present[ ] = current particle (solution)
pbest[ ] = best solution among each particle
gbest[ ] = best among defined as stated before
rand( ) = random numbers between (0,1)
w = inertia weights, usually 0.8 or 0.9
C_1, C_2 are learning factors. Usually, $C_1 = C_2 = 2$.

4.2 GENERAL FLOW CHART OF PSO

The general flow chart of PSO can be described as follows:

Step. 1: Generation of initial condition of each agent. Initial searching points (si °) and velocities (vi °) of each agent are usually generated randomly within the allowable range.
Step. 2: Evaluation of searching point of each agent. The objective function value is calculated for each agent. If the value is better than the current pbest of the agent, the pbest value is replaced by the current value.
Step. 3: Modification of each searching point.
Step. 4: Checking the exit condition. Otherwise, go to step 2.
Fig. 4. shows the general flow chart of PSO strategy.

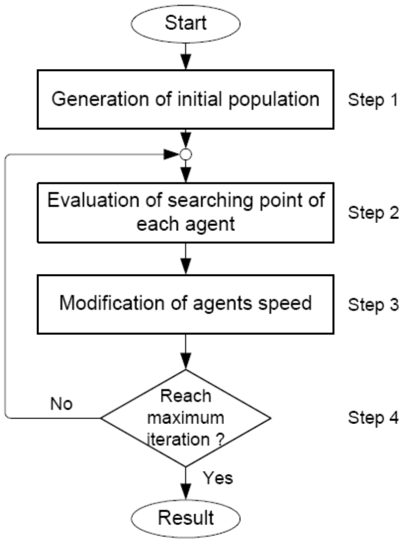

Fig 4. A general PSO Algorithm

4.3 ADAPTATION OF PSO TECHNIQUE TO MILLING OPTIMIZATION PROBLEM

In order to search for optimal process parameters, neural network model of cutting force was integrated with particle swarm optimizer. The architecture of system is shown in Fig. 5. The optimization process executes in two phases. In first phase, the neural prediction model on the basis of recommended cutting conditions generates 3D surface of cutting forces, which represent the feasible solution space for the PSO algorithm.

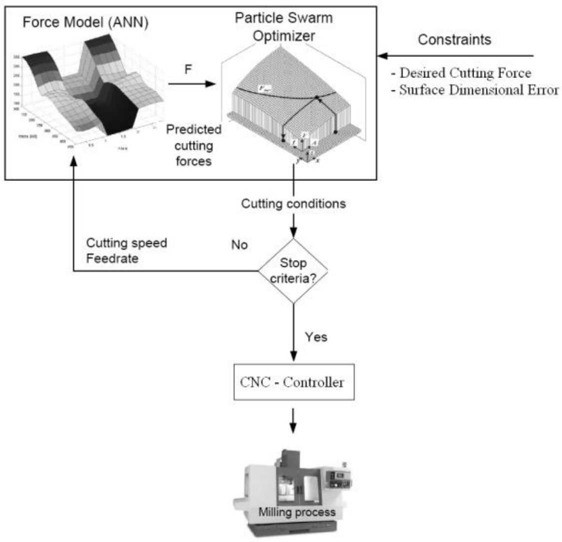

Fig. 5 PSO based neural network optimization scheme

The optimization process executes in two phases. In first phase, the neural prediction model on the basis of recommended cutting conditions generates 3D surface of cutting forces, which represent the feasible solution space for the PSO algorithm.

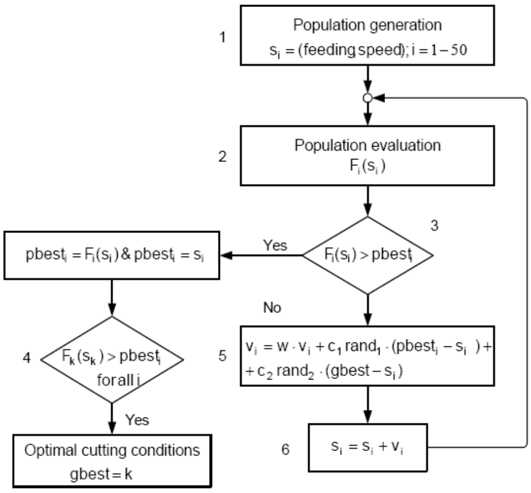

Fig. 6 PSO algorithm for optimization of cutting conditions

PSO algorithm generates a swarm of particles on the cutting force surface during the second phase. Swarm of particles flys over the cutting force surface and search for maximal cutting force. The coordinates of a particle which has found the maximal (but still allowable) cutting force represent the optimal cutting conditions. Fig.6 shows the PSO flowchart of optimization of milling process.

V. Deformation Analysis Using FEA Software

For the predicted cutting force values, the deformation is estimated using FEA Software ANSYS. The finite element model obtained for existing boundary conditions is shown in fig.7.

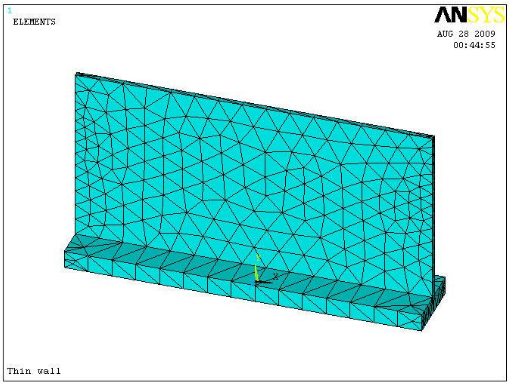

Fig 7. FE Model of work piece

As the work piece is clamped on the top face during machining, nodes located at the clamping area and the bottom surface are constrained in all degrees of freedom. Various rib structures are modeled by changing the geometrical parameters, namely, thickness, height, and length of the rib that are given below:

- Thickness of the rib (mm): 2, and 3.
- Length of the rib (mm): 100, and 150.
- Height of the rib (mm): 40 and 50.

VI. Conclusion

The objective function is determined by neural cutting force model (cutting force simulator). The goal of this case is to minimize the force function under given constraints. This problem is solved using the PSO algorithm. In PSO, 50 particles were used and search continues until error gradient is smaller than a specified value. Matlab code simulates the trained neural network to predict cutting forces at given cutting distances and these values are used to calculate the objective function which PSO algorithm attempts to minimize. The results are tabulated in Table 2. Each run corresponds to each time the program is run to find the optimum machining parameters. Table 2 shows optimal cutting conditions along with the number of generations it took to reach that optimum.

Table 2 Repeatability of results

Run	Speed (rpm)	Feed rate (mm/min)	F (N)	No of Iterations
1.	24100	1360	30	22
2.	24000	1380	30	25
3.	24000	1400	28	28
4.	24050	1390	28	32
5.	24050	1385	29	22

This optimization method has higher convergence, unlike traditional methods and it is always successful in finding the global optimum.

VII. Scope For Future Work

This study has presented multi - objective optimization of thin rib machining process by using neural network modeling and Particle swarm optimization. A neural network model was used to predict cutting forces during machining and particle swarm optimization was used to obtain optimum cutting speed and feed rate. This research can be extended to develop the error compensation strategy for thin-walled ribs machining.

References

[1] I. Nieminen, J. Paro and V. Kauppinen “High-Speed Milling Of Advanced Materials”, Joumal of Materials Processing Technology 56 (1996) 24-36.
[2] He Ning, Wang Zhigang, Jiang Chengyu, Zhang Bing, “Finite element method analysis and control stratagem for machining deformation of thin-walled components”, Journal of Materials Processing Technology 139 (2003) 332–336
[3] S. Ratchev, S. Liu , A.A. Becker, “Error compensation strategy in milling flexible thin-wall parts”, Journal of Materials Processing Technology 162–163 (2005) 673–681
[4] Tang Aijun, Liu Zhanqiang, “Deformations of thin-walled plate due to static end milling force”, journal of materials processing technology 206(2008) 345–351
[5] F. Cus , U. Zuperl , V. Gecevska, “High speed end-milling optimization using Particle Swarm Intelligence”, Journal of Achievements in Materials and Manufacturing Engineering, VOLUME 22, ISSUE 2, June 2007.
[6] Liu Gang, “Study on deformation of titanium thin-walled part in milling process”, journal of materials processing technology 209(2009), 2788–2793
[7] Tang Aijun, Liu Zhanqiang, “Deformations of thin-walled plate due to static end milling force”, journal of materials processing technology 206(2008), 345–351
[8] S. Ratchev, S. Liu , A.A. Becker, “Error compensation strategy in milling flexible thin-wall parts”, Journal of Materials Processing Technology 162–163 (2005) 673–681
[9] Zhixin Liu, Dawei Zhang, Houjun Qi, “Surface Roughness Modeling of High Speed Machining TC4 based on Artificial Neural Network Method”, IEEE Explore (2009).
[10] R. Saravanan, “Manufacturing Optimization through Intelligent Techniques”, First Ed, Taylor & Francis group, London, 2006.