Int J Adv Manuf Technol (2001) 18:693–700

The International Journal of
Advanced
Manufacturing
Technology

A Novel Artificial Neural Networks Force Model for End Milling

V. Tandon and H. El-Mounayri

Department of Mechanical Engineering, Purdue School of Engineering and Technology, IUPUI, Indianapolis, IN, USA

The physical process of multipoint metal cutting depends on a large number of parameters that are strongly interlinked. A number of empirical and semimechanistic models are described in the literature. This paper uses the artificial neural networks (ANNs) approach to evolve a comprehensive model for critical process parameters, such as cutting force, based on a set of input machining conditions. A set of eight input variables is chosen to represent the machining conditions, and process parameters (such as maximum force and mean force) are predicted. Exhaustive experimentation is conducted to develop the model and to validate it. The model is tested for a typical machining scenario found in industry, namely pocket-milling. Excellent agreement between the simulated and experimental forces is found.

Keywords: End milling; Force modelling; Neural networks

1. Literature Review

Liu and Wang [1] aptly sum up the scenario. As the machining process is nonlinear and time-dependent, it is difficult for the traditional identification methods to provide an accurate model. Compared to traditional computing methods, the artificial neural networks (ANNs) are robust and global. ANNs have the characteristics of universal approximation, parallel distributed processing, hardware implementation, learning and adaptation, and multivariable systems. Because of this, ANNs are widely used for system modelling, function optimising, image processing, and intelligent control. ANNs give an implicit relationship between the input(s) and output(s) by learning from a data set that represents the behaviour of a system [2].

Lazaro et al. [3], in their ANN implementations, evolve knowledge of the machining environment by training these networks on run-time data. This is done every time a new part is encountered. A tape-tuning module has been presented which, for obvious reasons, finds use in the optimisation of mass-produced machined parts. It is, however, the efficient use of the limited available knowledge of the process that makes this approach interesting [3–7]. The weights are calibrated by training the algorithm on cutting-test results, which represent turning a work material with specified cutting tools. Tool wear is measured for each case, and weights are identified using the iterative learning process of back-propagation (BP). This is the most widely used optimisation procedure based on gradient descent that adjusts the weights to reduce the system error or cost function which is estimated by the total error for all patterns. This mean square error cost function is defined as E,

$$E = \frac{1}{p}\sum_{p=1}^{p} E^p \tag{1}$$

where E^p = squared error function for each pattern p.

Liu and Wang [1] also propose a modified BP ANN which adjusts its learning rate and adds a dynamic factor in the learning process for the on-line modelling of the milling system. Further, another modified (Levenberg–Marquardt, ALM) neural network is proposed for the real-time optimal control of the milling process. However, this study has severe limitations, the most important being the use of only one machining parameter as the input. All other parameters affecting the process have apparently been kept constant. This leads to the conjecture that the model evolved by their ANN is not generic enough to represent the milling process plant function. In this work, we propose to develop a generic end-milling process model using ALM neural networks. A much larger set of input machining parameters is considered than in other work reported so far.

2. Experimental Data

As a first step, a typical range of machining parameters is selected and experimental data over this whole range is conducted and identified as training and testing data sets for the neural network. The three components of the cutting force are measured by a Kistler 9257B dynamometer. These were sampled at 2500 Hz for 10 s each, and have been stored in files in spreadsheet format. The machine tool used for all the experiments in this work is a TRIAC® 3-axis CNC milling machine by Denford®. It is equipped with a proprietary controller PNC3®. Travel limits in the various axes are: X-travel = 290 mm, Y-travel = 170 mm, Z-travel = 200 mm.

Correspondence and offprint requests to: Dr H. El-Mounayri, Department of Mechanical Engineering, Purdue School of Engineering, Indianapolis, IN, USA. E-mail: hel-moun@engr.iupui.edu

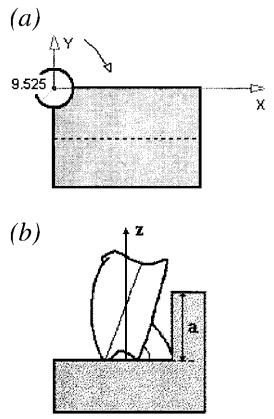


Fig. 1. (*a*) Radial depth of cut. (*b*) Axial depth of cut.

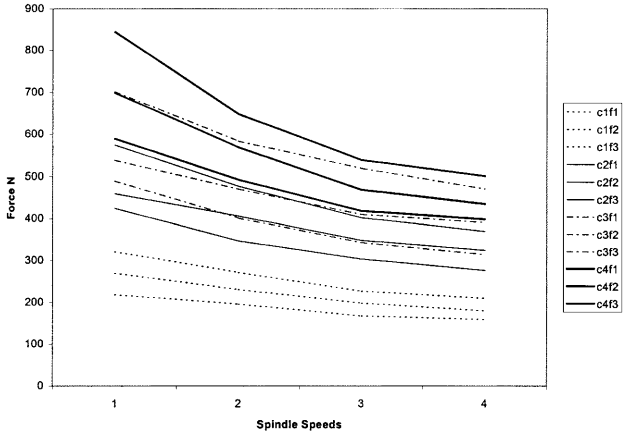


Fig. 2. Dependence of cutting force on spindle speed.

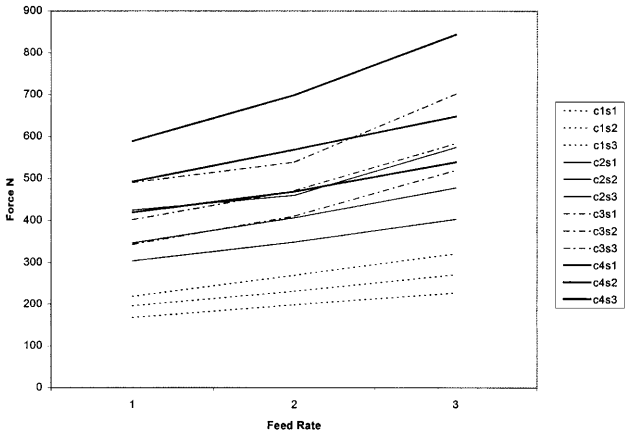


Fig. 3. Dependence of cutting force on feedrate.

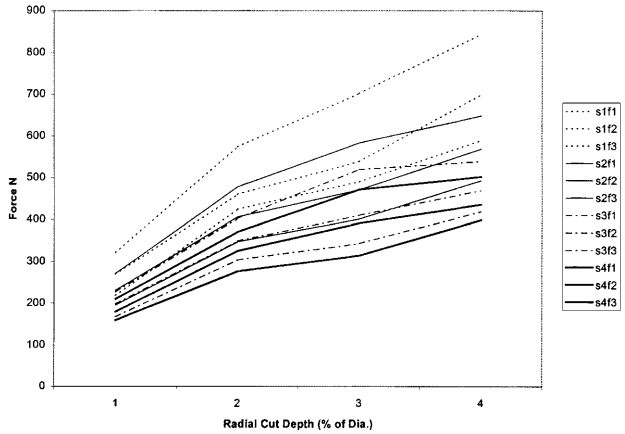


Fig. 4. Dependence of cutting force on radial depth of cut.

The spindle adapter has an R8 taper, which accommodates tools only with a $\frac{3}{8}$" shank. The spindle is powered by a 1 hp variable speed d.c. motor, programmable for 150–2500 r.p.m. Positioning resolution on the machine is 0.005 mm.
The experiments were carried out for all combinations of the chosen parameters, which are radial depth of cut, feedrate, and spindle speed.

2.1 Radial Depth of Cut

This parameter signifies the immersion (as a percentage of cutter diameter) of the milling tool in the workpiece, and is illustrated in Fig. 1(*a*) (Fig. 1(*b*) shows the definition of the axial depth of cut). The shaded rectangle is the workpiece as seen in plan view. The circle is the milling cutter, while the overlap along the *Y*-axis is the radial depth of cut (which can vary from 0% to 100% of the cutter diameter).

Four values over the possible range have been selected for use in the experiments:

c_1 = 25% of the cutter diameter
c_2 = 50% of the cutter diameter
c_3 = 75% of the cutter diameter
c_4 = 100% of the cutter diameter

2.2 Feedrate

This parameter shows the rate at which material is being removed in the process. Three values over the permissible range have been selected:

$f_1 = 100 \text{ mm min}^{-1}$
$f_2 = 120 \text{ mm min}^{-1}$
$f_3 = 150 \text{ mm min}^{-1}$

2.3 Spindle Speed

This parameter signifies the spindle rotation speed. It may also be presented as the cutting velocity, υ_c,

$$\upsilon_c = \pi DN \tag{2}$$

where D = cutter diameter (mm) and N = spindle speed (r.p.m.).
Five values over the possible range have been selected:

s_1 = 450r.p.m. (corresponds to $\upsilon_c = 8.98 \text{ m min}^{-1}$)
s_2 = 600r.p.m. (corresponds to $\upsilon_c = 11.97 \text{ m min}^{-1}$)
s_3 = 750r.p.m. (corresponds to $\upsilon_c = 14.96 \text{ m min}^{-1}$)
s_4 = 900r.p.m. (corresponds to $\upsilon_c = 17.95 \text{ m min}^{-1}$)

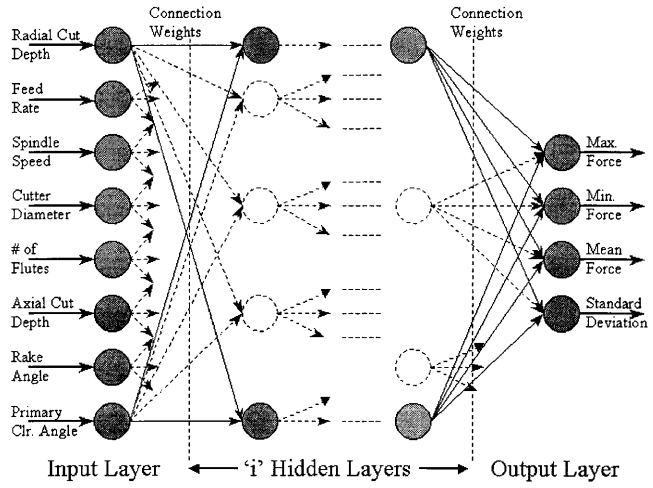


Fig. 5. Predictive force model network topology.

s_5 = 1000r.p.m. (corresponds to v_c = 19.95 m min^{-1})

Note that another measure often quoted is the "feed per tooth", this is indicative of both the feedrate and the spindle speed.

$$\text{Feed/tooth} = \frac{F}{N \times Z} \tag{3}$$

The first set of 48 experiments is conducted using a 2-flute, HSS, Do-All® end mill. The tool geometry parameters are rake angle = 14°, primary clearance angle = 16°, and the helix angle = 37.5°. This is a tool designed specifically for non-ferrous metals such as aluminium and has a higher rake. In this first set of experiments, the values vary from 0.05 mm tooth^{-1} (for F = 100 mm min^{-1} and N = 1000 r.p.m.) to 0.125 mm tooth^{-1} (for F = 150 mm min^{-1} and N = 600 r.p.m.). A second set of experiments is conducted using a 4-flute, HSS, Do-All® end mill. The tool geometry parameters are rake angle = 10°, primary clearance angle = 12°, and the helix angle = 30°. A third set of experiments is conducted using a 2-flute, HSS, Do-All® end mill. The tool geometry parameters are rake angle = 10°, primary clearance angle = 12°, and the helix angle = 30°.

Plots of the maximum resultant force versus machining parameters observed in the first set of experiments (with tool 1) are presented in Figs 2, 3, and 4 (c_i is the ith value of the radial cut depth, f_i is the ith value of the feedrate, s_i is the ith value of the spindle speed used), and demonstrate the general trend of dependence on each of these parameters separately. This summarises all we know about the machining process.

We can safely infer from those graphs that the cutting forces are inversely proportional to the spindle speed and directly proportional to the feedrate and radial depth of cut. This information will be needed when evaluating the simulation and optimisation empirically.

3. Current Approach

A feed forward neural network is designed with one to two hidden layers and sigmoid activation functions. To predict the maximum force, mean force, and other relevant conditions, it is important to describe the ALM method before proceeding to the details of the development.

Back propagation (BP) is the training method of choice. Standard BP is a gradient descent algorithm and the name BP refers to the manner in which the gradient is computed for a nonlinear multilayer ANN. One of the main attractions of using BP networks is that they tend to give reasonable results when presented with inputs that they have never seen. Typically, a new input will lead to an output similar to the correct output for input vectors used in the training that are similar to the new input being presented. This generalisation property makes it possible to train a network on a representative set of input/target pairs and obtain good results without training the network on all possible input/output pairs, which is generally impossible in situations such as ours. Also, with this method, the order in which the patterns are presented to the network does not influence the training. This is also because adaptation is done only at the end of each epoch.

Table 1. Pre-processing of experimental data.

Experiment conditions 1		F_x	F_y	F_z	Resultant
2-flute, HSS	Maximum	0.0440	0.0732	0.0928	2.1841
Rake = 14	Minimum	−1.6309	−1.4600	−0.5713	0.0000
Clr. angle = 16	Mean	−0.7501	−0.6012	−0.2231	1.0016
Radial = 25%	Skew	0.1558	−0.0176	−0.0235	−0.0906
Feed = 100 mm min^{-1}	Standard	0.5641	0.4827	0.1884	0.7467
Spindle speed = 600 r.p.m.	deviation				
Experiment conditions 2					
2-flute, HSS	Maximum	0.0440	0.0928	0.2100	1.9576
Rake = 14	Minimum	−1.4941	−1.2500	−0.4981	0.0000
Clr. angle = 16	Mean	−0.6974	−0.5018	−0.1919	0.8958
Radial = 25%	Skew	0.1782	−0.0375	−0.0041	−0.1094
Feed = 100 mm min^{-1}	Standard	0.5172	0.4008	0.1584	0.6525
Spindle speed = 750 r.p.m.	deviation				

3.1 Back Propagation Algorithm

The simplest implementation of BP updates the network weights and biases in the direction in which the performance function decreases most rapidly (i.e. the negative of the gradient). These corrections may be made incrementally (after each pattern presentation) or in batch mode. In the latter case, the weights are updated only after the entire training pattern set has been applied to the network. Before the update can be evaluated it is important to define the error signal on each of the PEs [8,9].

$$\delta_{kj} = z_{kj}(1 - z_{kj})(b_{kj} - z_{kj}) \tag{4}$$

where,

δ_{kj} = error signal for the jth PE on the kth pattern
z_{kj} = response on the jth PE (activation function)
z_{kj} $(1- z_{kj})$ = first derivative of the activation function
b_{kj} = target value for the jth PE

The gradients calculated at each pattern step are now added together to determine the change. Further, a momentum factor is built into this evaluation to avoid the problem of being caught in local optima.

$$w_{ji}^{new} = w_{ji}^{old} + \eta \sum_{k} \delta_{kj} y_{ki} + \alpha \Delta w_{ji}^{old} \tag{5}$$

where,

η = learning coefficient
α = momentum coefficient
y_{ki} = activation function for the ith PE
Δw_{ji}^{old} = previous weight update for the ith PE
w_{ji}^{new} = corrected weight for the ith PE, from the jth PE in the previous layer
w_{ji}^{old} = previous weight for the ith PE from the jth PE in the previous layer

Note that the above methods are essentially gradient descent methods [10]. Use of second-order Newton methods have also been reported and the update rules applicable are as follows,

$$\mathbf{w}^{new} = \mathbf{w}^{old} - \mathbf{H}(b_{ki}, z_{ki})^{-1} \mathbf{g}_{ki} \tag{6}$$

where,

$\mathbf{H}$ = Hessian matrix of the mean square error (performance function)
$\mathbf{g}_{ki}$ = gradient with respect to current weights and biases

A variation of the standard BP technique for adjusting the network weights is the Levenberg–Marquardt technique [11]. This algorithm which is similar to quasi-Newton methods, approaches second-order training speed without much added computational expense. It works on the premise that since most of the performance functions have the form of the sum of squares, the Hessian matrix (second derivatives), $\mathbf{H}$, can be approximated as

$$\mathbf{H} = \mathbf{J}^T \mathbf{J} \tag{7}$$

hence, the gradient may be computed as

$$\mathbf{g} = \mathbf{J}^T \boldsymbol{\delta} \tag{8}$$

where $\mathbf{J}$ is the Jacobian matrix containing the first derivatives of the network errors with respect to network weights and δ is the error for ith pattern.

The update rule is a Newton-like expression,

$$\mathbf{w}_i^{new} = \mathbf{w}^{old} - [\mathbf{J}^T \cdot \mathbf{J} + \mu \cdot \mathbf{I}]^{-1} \mathbf{J}^T \cdot \boldsymbol{\delta} \tag{9}$$

Note that when the scalar μ is zero, this is the second-order Newton's method, which is faster and more accurate near the error minimum, whereas when μ is large, it becomes a gradient descent with a small step size [11].

3.2 Predictive Force Modelling

A feed forward ALM network is designed and studied for effectiveness in learning the nonlinear map between the input machining parameters and the output conditions. The ANN designed for this application is presented in Fig. 5.

3.3 Data Pre-Processing

Before the ANN can be trained and the mapping learnt, it is important to process the experimental data into patterns. Training/testing pattern vectors are formed. Each pattern is formed with an input condition vector, P_i,

$$P_i = \begin{bmatrix} RadialCutDepth \\ FeedRate \\ SpindleSpeed \\ CutterDiameter \\ \#offlutes \\ AxialCutDepth \\ RakeAngle \\ ClearanceAngle \end{bmatrix}$$

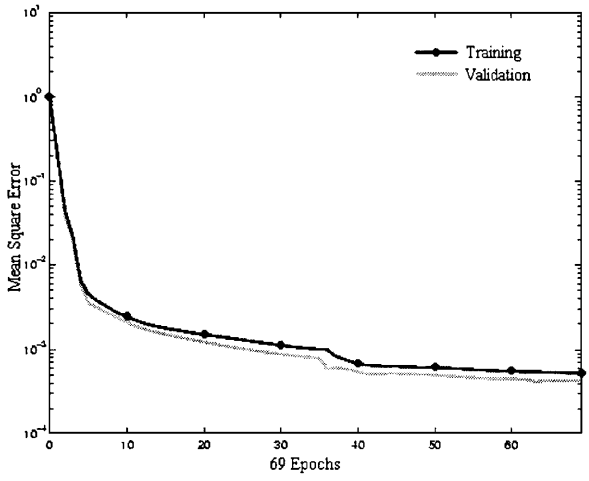


Fig. 6. Development of error performance.

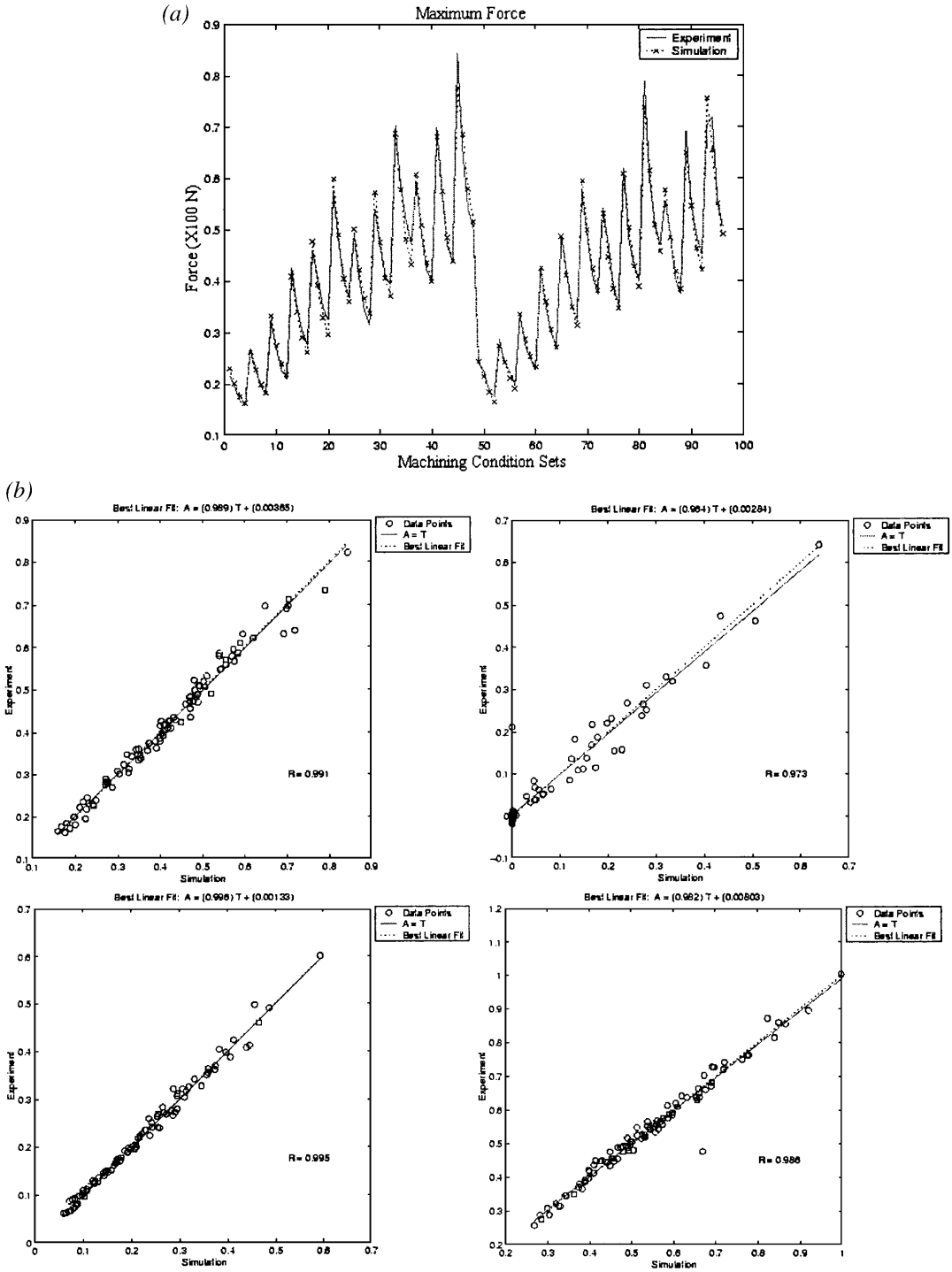


Fig. 7. (*a*) Simulation versus experimental data. (*b*) Regression analysis of network response.

and the corresponding target vector,

$$T_i = \begin{bmatrix} Max.Force \\ Min.Force \\ MeanForce \\ Std.Deviation \end{bmatrix}$$

Samples of pre-processed data are shown in Table 1. The force here refers to the resultant cutting force, $F = \sqrt{(F_x^2 + F_y^2 + F_z^2)}$.

Since only a limited number of experiments are representative of the feasible parameter space, it is important that the ANN realises each set fully [2]. This is achieved by normalising

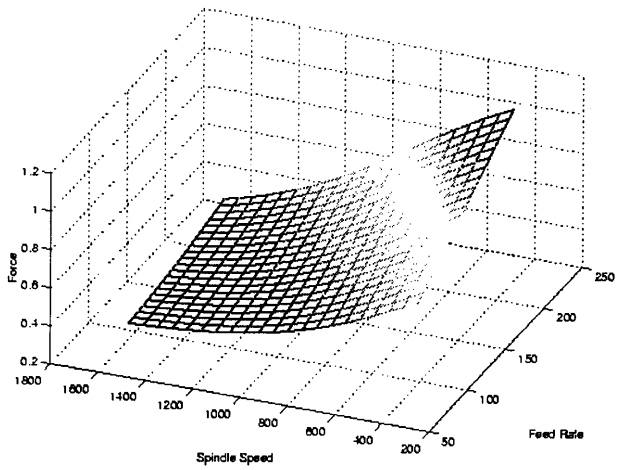


Fig. 8. Force map over two-machining condition space.

the data as follows,

$$X = (X_R - X_{min})\frac{X_{N\max} - X_{N\min}}{X_{\max} - X_{\min}} + X_{N\min} \quad (10)$$

here X_R is the real value of the variable before normalisation. X_{min} and X_{max} are the minimum and maximum values of the variable X. These are normalised to values X_{Nmin}, X_{Nmax} such that $0 < X_{N\,min} < X_{N\,max} < 1$. For example, spindle speed is varied from 450 to 1000 r.p.m. in our experiments. Thus, $X_{min} = 450$ and $X_{max} = 1000$. Then we choose X_{Nmin} to be 0.1 and X_{Nmax} to be 1. In this way the range 450–1000 is now mapped to 0.1–1. So when $X_R = 600$ r.p.m., $X = 0.345$.

3.4 Network Topology

As a first step, only one hidden layer is designed and the number of PEs is varied. According to White's theorem (1989) [8], one layer with nonlinear activation functions is sufficient to map any nonlinear functional relationship with a reasonable level of accuracy. The configuration of the first layer is chosen to be 6. This is a reasonable choice, as the error beyond that number does not reduce drastically. Further, we do not want to have a larger than required network because:

1. The generalisation characteristics of the network suffer owing to hard coding of the training data in the increased number of connections.
2. Computational expense increases with increase in size.

Again, a similar exercise is conducted with a second hidden layer. Here, the number of PEs is varied between 1 and 6. As the first hidden layer is successful in reducing/compressing the dimensionality of the data, it is customary to have fewer PEs in subsequent layers.

3.5 Training and Validation

As noted earlier, each experiment conducted amounts to one pattern vector. The complete set was divided in the ratio 3:1 uniformly to constitute two data sets. Whereas the larger set (75% of available data) was used for learning the mapping, the remainder was reserved for use in testing and validation of the functional relationship produced by the ANN.

Network weights were initialised one layer at a time using the Nguyen–Widrow algorithm [12]. This reduces training time by setting the initial weights of the hidden PEs so that each is assigned its own interval over the input space. Thus, the active regions of the layer's neurons will be distributed roughly evenly over the input space. The advantage of this over completely random initialisation is that once the training starts, the weight movements are smaller and settle quickly, since the majority of weight movements were eliminated by the method of initialisation [12].

Results of training and testing conducted on 96 experimental data sets are presented. Error in simulation recall is extremely small (~0.0003) for both interpolative and extrapolative recall (Fig. 6). Figure 7(*a*) shows the comparison of experimental and simulated data for the maximum force output component.

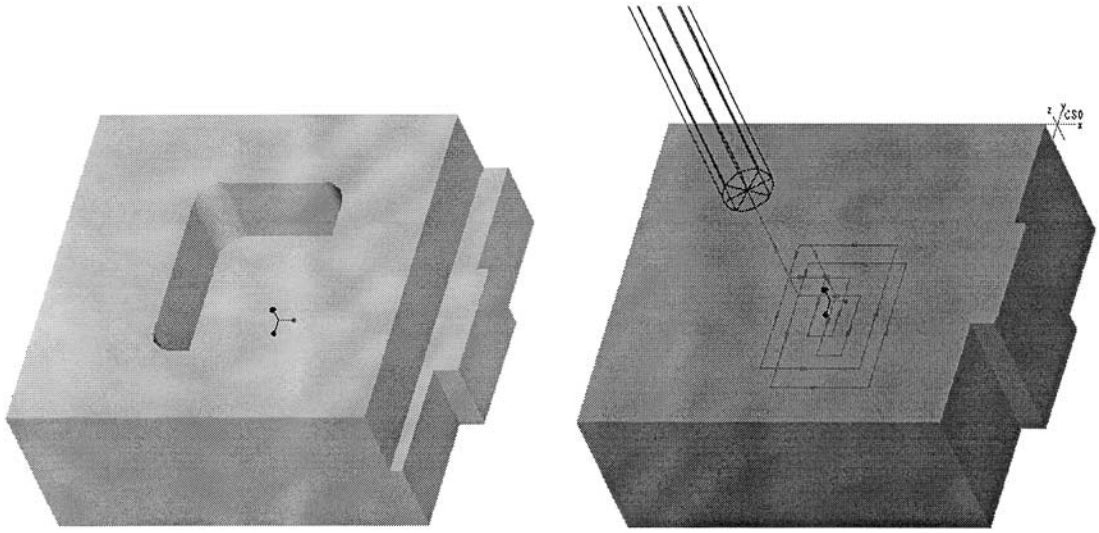
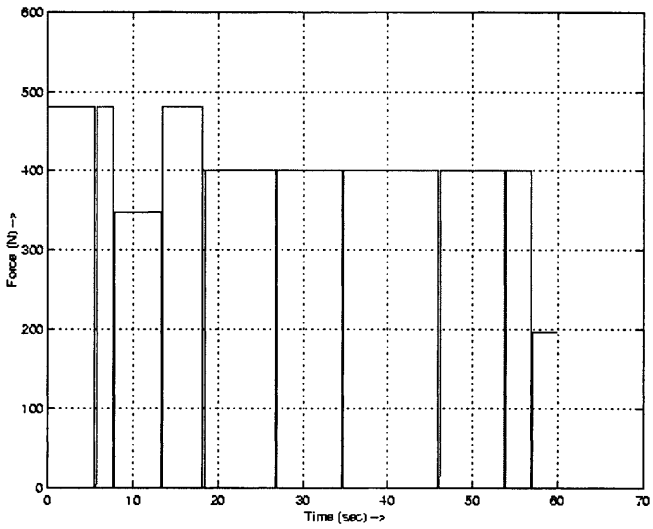


Fig. 9. Design model and tool path on workpiece.

(a)



(b)

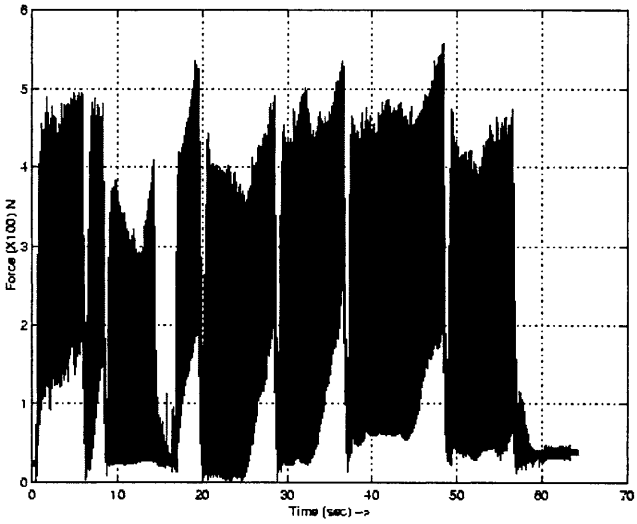


Fig. 10. (*a*) Simulated force variation (*b*) Experimentally acquired force variation.

Next, a linear regression between the network response and the target outputs is perfomed. The results are presented separately for the four output parameters (see Fig. 7(*b*)). The network is seen to map each of the four outputs very well. The correlation coefficients are also given.

Further, force values for a common range of values have been computed to enable us to understand better the dependence of cutting force on the machining conditions of feedrate and spindle speed. The force map evolved (over commonly used machining space) is presented in Fig. 8. This is a unique way of presenting the results, in the sense that, to our knowledge, no mathematical models reported so far have presented such a comprehensive map of the process.

Other machining parameters in the map above are cutter diameter = 0.25in, number of flutes = 2, radial cut depth = 0.75*D* (where *D* is the diameter of the cutter), axial depth of cut = 1.0*D*, rake angle = 10°, and primary clearance angle = 12°.

4. Test Case

We shall now demonstrate the application of our research and further validate it for a very common machining situation, namely, pocket milling. We must machine the workpiece to create the rectangular pocket on the top face of the component (see Fig. 9).

Four different immersion levels are found in the tool path. The conditions are analysed using the predictive model developed and the results are presented in Table 2.

Note that our machine tool has a small 0.2–0.4 s delay between different segments of the cutter path. This has been manually incorporated into the simulated force variation graph (Fig. 10(*a*)). The above machining operation was conducted, and the actual cutting force data acquired, using the set-up described above. The measured forces are shown in Fig. 10(*b*).

The simulated and experimental forces compare very well (within 5% difference). The difference of peaking can be attributed to:

1. Inaccuracies in the computation of the immersion geometry as it varies during cornering.
2. The wear condition of the tool which leads to an increase in the cutting force.

5. Conclusions and Future Work

Artificial neural networks have been used successfully for predictive force modelling of the flat-end milling process. An effort is made to include as many different machining conditions as possible that influence the cutting process. Exhaustive experimentation is conducted and presented here. This forms the basis of the model developed. Further, the model is tested on the commonly encountered operation of pocket milling, yielding very good results.

While the results of the current work are encouraging, it is important to note its limitations, and thus chart a future course for the research. In the current work, all simulation-related experimentation were conducted using flat-end milling cutters. This limits the application of the predictive force model to similar cutters. A more general treatment would require the use of a variety of tool shapes such as ball- and taper-end mills. Further, because of machine tool limitations, only a $\frac{1}{4}$ in

Table 2. Various immersions and associated machining parameters.

Immersion	Feed/speed conditions	Maximum force (N)
1	100/800	481
$\frac{3}{4}$	100/800	401
$\frac{1}{2}$	100/800	347
$\frac{1}{4}$	100/800	195.7

diameter end mill could be used. Whereas this is sufficient to demonstrate the potential of the concept, it is important to extend its capabilities to model meaningful industrial implementations. This would also require conducting more experiments, such as machining with different tool shapes and types. While the essential structure of the ANN model will remain the same, suitable parameters representing the immersion conditions may have to be formulated. For example, in the case of flat-end mills, radial and axial depths of cut are sufficient to describe the uncut chip thickness (load). However, for a more general tool shape, this uncut chip thickness may have to be represented using a different set of parameters. These will then replace the depths of cut in the ANN input vector.

This work is also limited in terms of the workpiece material (here aluminium) and the tool material (here HSS). Other workpiece/tool combinations must be considered in subsequent developments.

Also, future work should involve extending the predictive force model to include the dynamic effects of the process in order to predict other process parameters (such as surface finish and chatter). This would also help in identifying and avoiding machining conditions prone to chatter. This extension will also require exhaustive experimentation on the dynamics of the cutter–workpiece system.

References

1. Y. Liu and C. Wang, "Neural network based adaptive control and optimisation in the milling process", International Journal of Advanced Manufacturing Technology, 15(11), pp. 791–795, 1999.
2. Q. Liu and Y. Altintas, "On-line monitoring of flank wear in turning with multi-layered feed-forward neural network", International Journal of Machine Tools and Manufacture, 39, pp. 1945–1959, 1999.
3. S. A. Lazaro, J. Zhang and L. A. Kendall, "Knowledge-based approach for improvement of CNC part programs", Journal of Manufacturing Systems, 13(1), pp. 20–30, 1994.
4. S. B. Billatos and P. C. Tseng, "Knowledge-based optimisation for intelligent machining", Journal of Manufacturing Systems – SME, 10(6), pp. 464–475, 1991.
5. S. T. Chiang, D. I. Liu, A. C. Lee and W. H. Chieng, "Adaptive control optimization in end milling using neural networks", International Journal of Machine Tools and Manufacture, 34(5), pp. 637–660, 1995.
6. T. J. Ko and D. W. Cho, "Adaptive optimization of face milling operations using neural networks", Journal of Manufacturing Science and Engineering, 120, pp. 443–451, 1998.
7. T. W. Liao and L. J. Chen, "Manufacturing process modeling and optimization based on multi-layer perceptron network", Journal of Manufacturing Science and Engineering, 120, pp .109–119, 1998.
8. R. C. Eberhart, P. Simpson and R. Dobbins, Computational Intelligence PC Tools, AP Professional, ISBN 0–12–228630–8, 1996.
9. D. E. Rumelhart and J. L. McClelland, PDP Research Group, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press, 2 vols, ISBN 0262181207 and 0262132184, 1986.
10. E. K. P. Chong and S. H. Zak, An Introduction to Optimization, John Wiley, 1996.
11. H. Demuth and M. Beale, Neural Network Toolbox v3 User's Guide, MathWorks, 1999.
12. D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights", Proceedings of the International Joint Conference on Neural Networks, vol. 3, pp. 21–26, 1990.