ORIGINAL ARTICLE

Mill-cut: a neural network system for the prediction of thermo-mechanical loads induced in end-milling operations

Jitender K. Rai · Louis Villedieu · Paul Xirouchakis

Abstract This paper presents the design and implementation issues of a generalized system called mill-cut, developed for the prediction of cutting forces and temperature in end-milling operations. Based on an ANN approach, mill-cut predicts all the three components of cutting forces and average shear plane temperature for a given set of machining parameters broadly categorized into three groups viz. (i) cutting tool geometrical parameters (ii) cutting parameters and (iii) workpiece material properties. In the present work, for representing overall machining condition, 15 machining parameters having major impact on the cutting forces and cutting temperature were chosen. The feed-forward back-propagated ANN architecture has been incorporated, which was initially trained with analytical data before incorporating it as part of an integrated system. Results obtained from the proposed model show good agreement with the experimental/numerical (FEM based) results available in the literature.

Keywords Artificial neural networks · End-milling · Cutting force · Cutting temperature

Nomenclature

D	Cutter diameter (mm)
N_t	Number of flutes
β	Helix angle (degrees)
α	Rake angle (degrees)
C_s	Side cutting edge angle (degrees)
d_a	Axial depth of cut (mm)
d_r	Radial depth of cut (mm)
n	Spindle speed (rps)
f_t	Feed-rate per tooth (mm/tooth)
K_c	Specific cutting pressure (N/mm^2)
m_c	Rise in specific cutting force (constant)
ρ	Workpiece material density (kg/m^3)
K	Workpiece thermal conductivity (W/m-K)
C_p	Workpiece specific heat (J/kg-K)
μ	Coefficient of friction
F_x	Cutting force along feed direction (N)
F_y	Cutting force along radial direction (N)
F_z	Cutting force along axial direction (N)
T_{avg}	Average shear plane temperature (°C)
H_n	Number of hidden layers
N_h	Number of neurons in hidden layers
LR	Learning rate
E_n	Number of epochs
E_j	Mean absolute error

J. K. Rai (✉) · L. Villedieu · P. Xirouchakis
Swiss Federal Institute of Technology STI-IPR-LICP,
Station 9,
Lausanne 1015, Switzerland
e-mail: jitenderkumar.rai@epfl.ch

1 Introduction

Today high speed computer numerical control (CNC) machining due to its high productivity and throughput is becoming an indispensable process for manufacturing thin wall large size parts of aerospace/automotive industries. In high speed CNC machining high material removal rates (MRR) are mainly achieved by increasing cutting parameters, namely, cutting speed, feed rate, depth of cut and width of cut. In order to do this in an economical way (both in terms of quality and cost), it is necessary to increase the power and accuracy of the machine tools. The increase in power to remove more material in a shorter time increases the magnitude of both cutting forces and temperature. This in

turn creates problems in holding accuracy, increases tool wear etc., thus affecting overall part quality. Therefore, in order to assure better machining results, the prediction of cutting forces and temperature acting during the milling operation is essential to investigate the effect of the machining loads (both mechanical and thermal) on part quality. A priori knowledge of the magnitude of cutting force components and temperature generated during machining operation for a given set of cutter and machining parameters will allow manufacturing engineers to select appropriate machining strategies, comprising of fixture layout, clamping forces and machining parameters; thus, resulting in better machined part quality (i.e., minimal elastic/plastic deformations and stresses) with minimal tool wear and few tool breakages.

In the past, milling force prediction has been partially achieved through three main approaches, namely (i) mechanistic approach [1–4] (ii) finite element methods [5, 7] and (iii) artificial intelligence (AI) techniques like artificial neural networks (ANN) [8–11]. In spite of better prediction accuracy FEM-based solutions failed to gain popularity due to their high computational time and software price. Whereas analytical approaches become, in essence, inefficient and intractable when dealing with nonlinear problems involving large number of interrelated process variables. On the other hand, neural networks with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract pattern and detect trends that are too complex to be noticed by either humans or other computer techniques [14]. A trained neural network can be thought of as an "expert" in the category of information, it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Other advantages of ANN include the following: (i)adaptive learning: an ability to learn how to do tasks based on the data given for training or initial experience,(ii) self-organisation: an ANN can create its own organization or representation of the information it receives during learning time, and (iii) real time operation: ANN computations may be carried out in parallel [12, 13].

During the last decade, ANN has been successfully applied to a certain extent in modelling various manufacturing processes, like sheet metal forming, pressure die casting and metal cutting. In metal cutting process researchers have mainly focused on the development of cutting force predictive model for the optimal selection of cutting parameters. In the recent work of Zuprel et al. [9] a five-layered feed-forward back propogated (BP) neural network architecture was proposed for the prediction of ball-end milling forces. This model ignores the influence of tool geometrical parameters on machining forces. Tansel et al. [8] proposed a combined ANN and genetic algorithm (GA)-based model for predicting optimal machining

parameters. This model predicts the feed direction cutting force from three input parameters namely depth of cut, feed rate and cutter diameter. In the work of Tandon et al. [11] a similar approach was used for predicting the maximum, minimum and mean force value from six cutting parameter inputs. Most of these reported works mainly deal with the mechanical aspect of the cutting process for specific tool-workpiece configuration. Despite a significant amount of published research work, there is no cutting process model reported which can predict cutting forces and temperature from a single ANN configuration considering the effects of overall machining process parameters.

This paper reports the design and implementation issues of a generic ANN based machining load predictive system called mill-cut, developed for the estimation of cutting forces and average shear plane temperature in end-milling. The main advantages of the proposed system over previous works are the accurate and faster prediction of cutting force components, namely F_x(feed force), F_y (tangential force), F_z(axial force) as shown in Fig. 1 and average shear plane temperature (T_{avg}) for given sets of machining variables (as listed in Table 1) from a single configuration. Consideration of overall machining parameters helps cutting tool designers and machining process planners in analyzing and selecting appropriate machining parameters in a user friendly environment. Moreover the system can be easily integrated with commercial CAM packages in order to simulate the dynamics of machining processes.

2 Variables influencing machining loads

Machining loads induced during an end-milling operation strongly depend on tool geometry, cutting conditions and workpiece material properties.
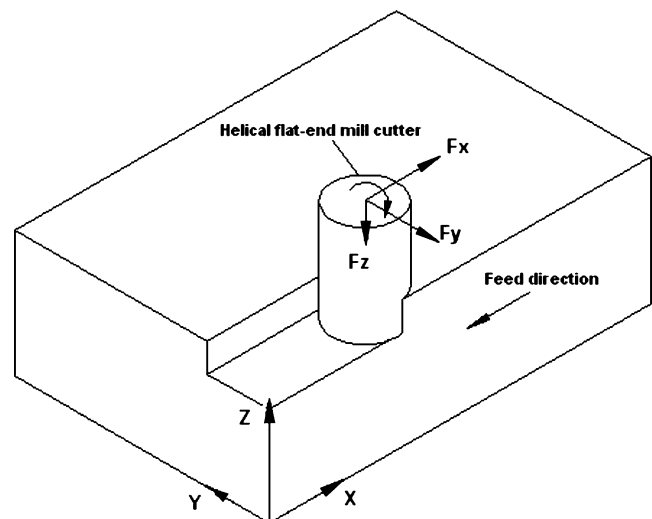

Fig. 1 Cutting forces in end milling operations

Table 1 Selected machining parameters

Cutter geometrical parameters				Cutting parameters				Workpiece material properties			
S.No	Parameter(A)	min$A$	max$A$	S.No	Parameter	min$A$	max$A$	S.No	Parameter	min$A$	max$A$
1.	D	6	22	1.	da	1	8	1.	Kc	550	2100
2.	Nt	2	6	2.	dr	4	20	2.	mc	0.21	0.25
3.	β	28°	32°	3.	n	10	40	3.	ρ	2700	7800
4.	α	5°	10°	4.	ft	0.1	0.3	4.	K	20	210
5.	Cs	40°	50°					5.	Cp	450	900
								6.	μ	0.25	0.35

As depicted in the literature, the magnitude of cutting forces depend most strongly on the feed-rate and axial depth of cut values [3, 4], whereas cutting temperature is highly influenced by the cutting speed and feed rate parameters [6, 7]. Tool geometrical parameters such as rake angle, tool nose radius and wedge angle are also having influence on both cutting forces and cutting temperature magnitudes [5, 6]. Workpiece material mechanical properties such as specific cutting force or specific cutting energy strongly influence both cutting forces and cutting temperatures whereas thermal properties such as thermal conductivity and specific heat strongly affect the thermal aspects of cutting [6, 7]. Therefore, for analyzing the effects of various machining parameters on thermo-mechanical aspects of cutting in real time environment, fifteen parameters broadly categorized into three groups viz. (i) Cutting tool geometrical parameters; helix angle(β),

rake angle(α), side cutting edge angle(C_s), cutter diameter(D) and number of flutes(N_t) (ii) cutting parameters; axial depth of cut(d_a), radial width of cut(d_r), feed rate(f_t) and cutting speed(n) (iii) the workpiece material properties of specific cutting pressure, rise in specific cutting pressure, density, specific heat, thermal conductivity and coefficient of friction were selected as input to the proposed ANN model as shown in Fig. 2.

3 Mill-cut design and implementation issues

In the present work a feed-forward multi layer supervised neural network architecture was developed for predicting four outputs namely three cutting force components i.e., feed force(F_x), radial force(F_y), axial force(F_z) and the average shear plane temperature (T_{avg}) from a given set of

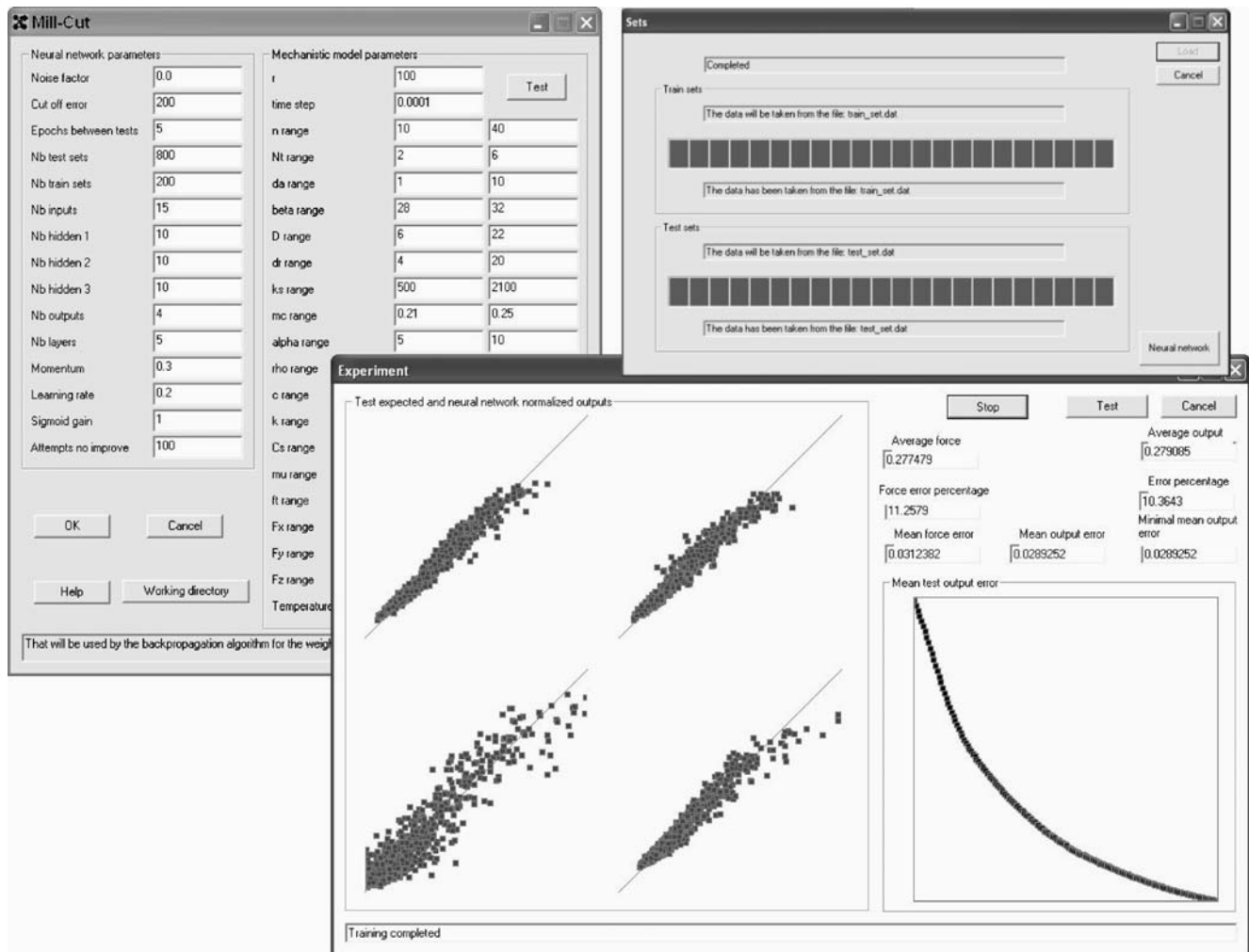
Fig. 2 Mill-cut ANN architecture
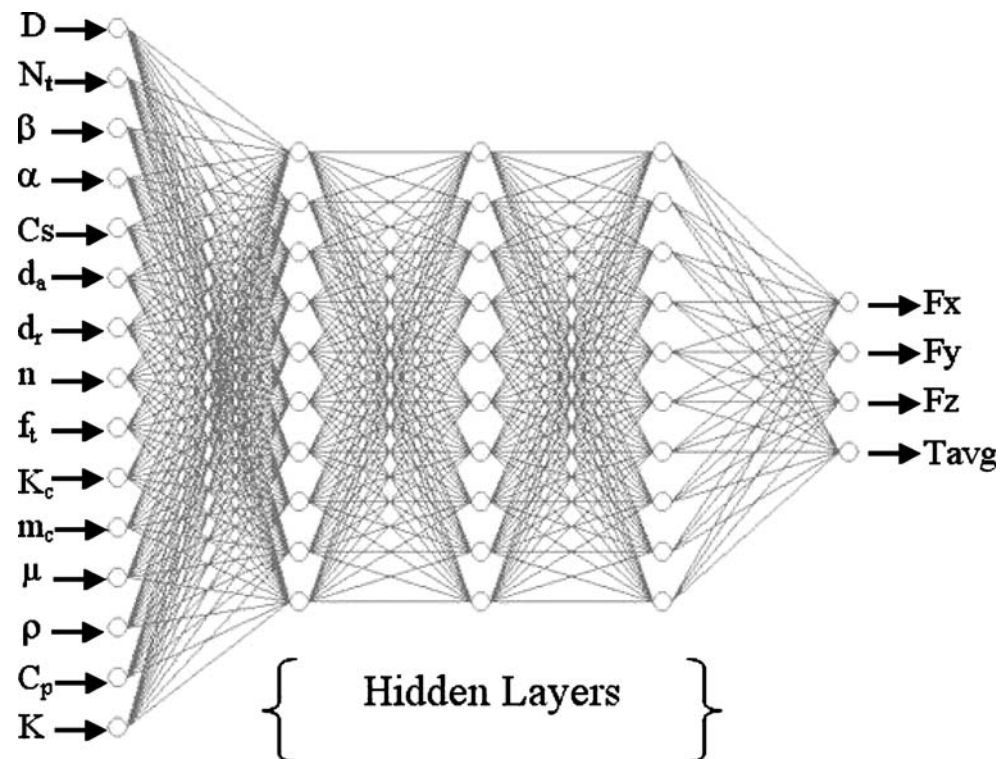
Fig. 3 Mill-cut graphical user interface (GUI)

fifteen input variables as listed in Table 1. The developed ANN architecture comprises of five layers viz. the input layer, three hidden layers and the output layer. As shown in Fig. 2 the input layer comprises of fifteen neurons representing each of the selected machining variables. Whereas the number of hidden layer(s)/neurons is selected using the hit and trial approach i.e., various combinations of hidden layer(s)/neurons were extensively experimented upon before the final configuration, which yields better accuracy was selected. The activation function used in the present work is the sigmoidal function defined by the logistic function:

$$f(x) = \frac{1}{1 + e^{-x\sigma}} \tag{1}$$

where x is the summed *weighted outputs* to the neurons and σ is the sigmoid gain(Threshold).

A Windows based neural-network toolbox with user friendly graphical interface (Fig. 3) developed during the course of this work using C++ programming language [14] was used to train the machining load predictive network.

The proposed system architecture comprises of two main modules: (i) Training and test data post-processing module, and (ii) The ANN module. The post-processing module mainly deals with the normalization and formatting of the training and test data sets which are then input to the ANN module to train the network. One can functionally activate the proposed system by simply executing the following tasks:

Step 1: First, specify the maximum-minimum range values for the selected input and output variables

Table 2 Selected helical flat-end mill cutters

Cutter No.	D	N_t	Cutter No.	D	N_t
1.	6	3	6.	18	2
2.	8	2	7.	20	3
3.	12	6	8.	20	4
4.	16	3	9.	22	3
5.	16	6	10.	22	4

Table 3 Simulated cutting condition data(normalized) for training and testing of mill-cut

| S.No. | ANN input parameters | | | | | | | | | | | | | | | | ANN Outputs | | |
	D	N_t	β	α	C_s	d_a	d_r	n	f_t	K_c	m_c	μ	ρ	C_p	K	F_x	F_y	F_z	T_{avg}
1.	0.350	0.585	0.001	0.564	0.174	0.193	0.896	0.809	0.480	0.823	0.747	0.859	0.711	0.514	0.304	0.215	0.231	0.128	0.208
2.	0.446	0.166	0.015	0.091	0.378	0.364	0.119	0.147	0.989	0.005	0.009	0.532	0.571	0.602	0.607	0.247	0.272	0.096	0.180
3.	0.783	0.057	0.166	0.663	0.876	0.451	0.803	0.352	0.608	0.520	0.302	0.727	0.956	0.926	0.539	0.275	0.314	0.090	0.126
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
267.	0.276	0.009	0.376	0.093	0.838	0.677	0.273	0.056	0.919	0.588	0.691	0.726	0.485	0.205	0.744	0.530	0.513	0.215	0.297
268.	0.385	0.108	0.468	0.458	0.361	0.949	0.735	0.744	0.599	0.609	0.572	0.152	0.225	0.425	0.803	0.523	0.595	0.356	0.346
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
998.	0.877	0.683	0.303	0.427	0.178	0.070	0.822	0.967	0.153	0.582	0.191	0.817	0.475	0.156	0.504	0.109	0.108	0.026	0.323
999.	0.722	0.682	0.732	0.406	0.835	0.280	0.475	0.569	0.756	0.123	0.368	0.035	0.517	0.663	0.426	0.251	0.209	0.207	0.204
1000.	0.375	0.346	0.105	0.949	0.272	0.921	0.847	0.550	0.472	0.317	0.456	0.983	0.298	0.739	0.567	0.456	0.498	0.163	0.217

as shown in Fig. 3. for normalizing the training and test data sets in the range 0–1.

Step 2:   Set the neural network parameters (shown in Fig. 3) appropriately and upload the training and test data sets to the ANN module.

Step 3:   Execute the network training.

The graphical display of the mean output error and the desired and predicted output comparative graphs (as shown in Fig. 3) plotted during the training phase allows users to interpret the network present learning status. Once training process is over (i.e., one of the predefined convergence criteria meet) by simply executing the test button (see Fig. 3) and defining a new set of input machining variables, machining loads can be determined interactively.

4 Gathering and postprocessing of training data

Any neural network will not be able to identify the relation between the given input-output parameters unless it is properly trained using accurate data, hence sufficient care must be taken in the preparation of training data. Few prerequisites for the preparation of training data are: for modelling any non linear problem a minimum of three levels for each input parameter must be chosen, the training data must cover the wide variety of possible ranges and values of each input-output parameter must be normalized appropriately. This training data could either be collected by conducting practical experiments or from FEM simulations using commercial packages like DEFORM3D™ or AdvantEdge™. The reported FEM-based software can only simulate single point orthogonal cutting operation and moreover, generating a large number of data sets (as in
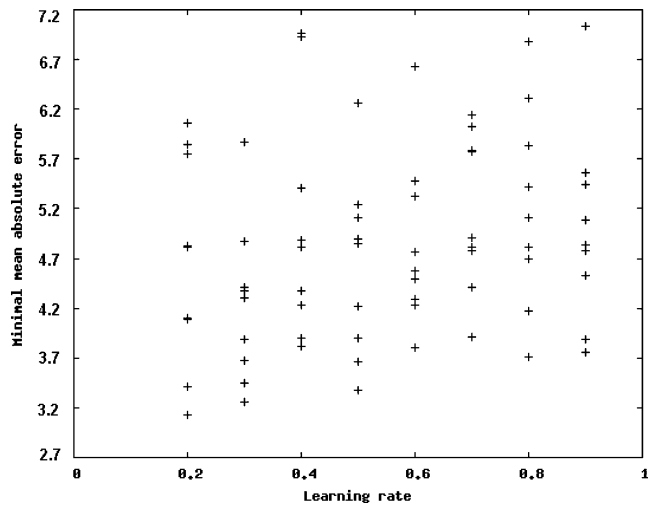

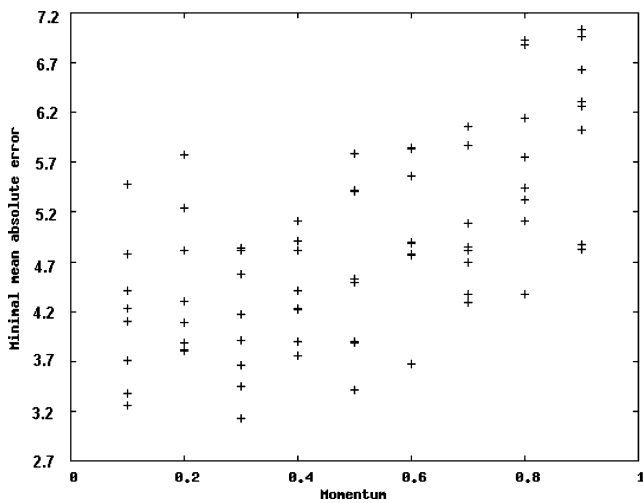**Fig. 4** Influence of learning rate on the prediction accuracy

Fig. 5 Influence of momentum on the prediction accuracy

Fig. 6 Influence of sigmoid gain on the prediction accuracy

the present case) for modelling the nonlinear relation between a large set of input-output parameters, use of any of these techniques will be very expensive both in terms of time and money. Therefore, in order to first test the feasibility of the proposed ANN architecture in solving the presented problem, initially it was trained using data sets generated by Li et al. [4]; Stephenson [6] mechanistic models. For this, ten different helical flat-end mill cutters (as listed in Table 2) and three different workpiece materials, namely aluminium, titanium and steel were considered. After defining maximum(maxA) and minimum(minA) range values for the selected input parameters using Sandvik tool catalogue [15], a total of 1,000 data series were generated, where each series comprise of the selected 15 input process variables and the four outputs. Feasibility of input parameters combinations is tested in terms of available machine power and allowable cutting parameters limits for different tool-workpiece material combinations as given in [15].

As in real case measured cutting force data contain noise, to meet this requirement 5–10% noise had been introduced randomly in the generated data sets.

Finally, the data set of 1,000 data series was normalized in the range 0–1 using Eq. 2 and divided into two representative subsets of 800 and 200 data series. The first subset was used for ANN training and the second for checking the trained ANN's prediction accuracy using unknown inputs.

$$v' = \frac{v - \min A}{\max A - \min A} \qquad (2)$$

where A is a parameter, having maximum and minimum values of $maxA$ and $minA$ as listed in Table 1; max-min normalization maps v of A to v' in the range 0–1. Table 3 presents a sample of normalized data sets used to train the proposed neural network.

5 Training and testing of mill-cut

The proposed ANN model is trained using the back propogation (BP) alogorithm [14]. The back propagation (BP) algorithm is the most widely used method for determining the error derivative of the weight (error surface gradient). The BP algorithm progresses iteratively, through a number of epochs (during iterative training of a neural network, an epoch is a single pass through the entire training set). On each epoch, the training cases are submitted to the network, and the target and actual output values are compared and the error is calculated. This error, together with the error surface gradient, is used to adjust the weights, and then the process is repeated. An overall performance

Table 4 Prediction accuracy for differnt network topology

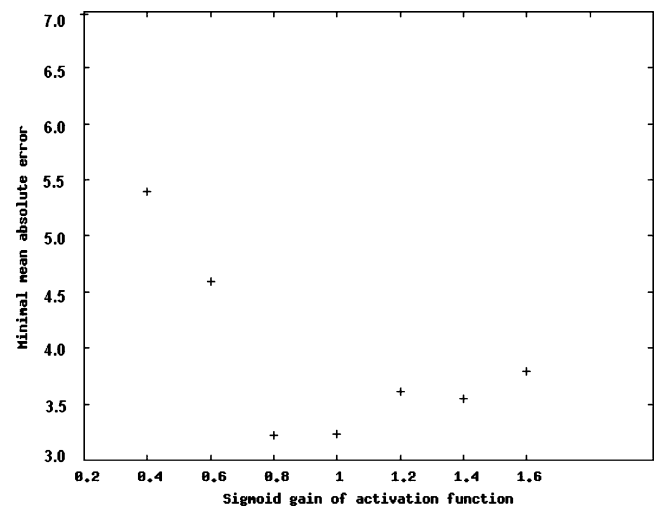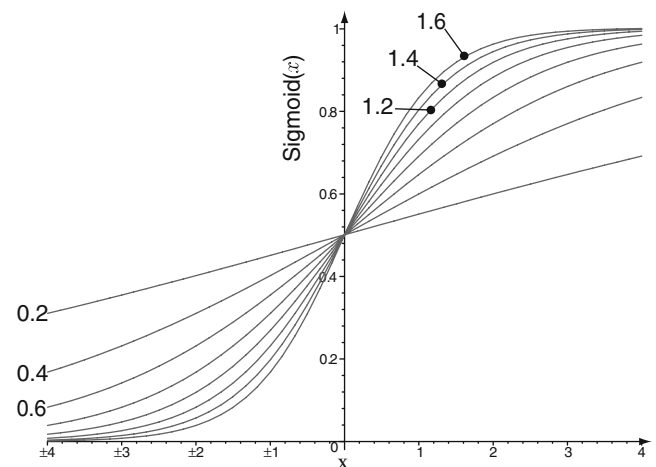| H_n	N_h	LR	Momentum	E_n	E_j (%)
One	10	0.2	0.1	5	5.913
Two	10	0.2	0.2	5	4.018
Three	10	0.2	0.3	5	3.147
Four	10	0.2	0.3	5	3.247

Fig. 7 Sigmoid shape of sigmoid gain values ranging from 0.2–1.6

Table 5 Validation test sets cutting conditions for ANN

ANN input parameters

S.No.	D	N_t	β	α	C_s	d_a	d_r	n	f_t	K_c	m_c	μ	ρ	C_p	K
1.	20	3	30	6.0	45	4	16	20	0.125	600	0.25	.26	2700	210	900
2.	20	4	30	7.0	45	1	8	20	0.125	600	0.25	.32	2700	210	900
3.	22	4	28	7.0	42	2	8	10	0.200	1700	0.25	.32	7500	50	460
4.	16	3	28	8.5	47	2	14	10	0.225	2100	0.23	.34	7800	50	460
5.	18	2	28	9.0	47	4	10	25	0.300	1700	0.25	.26	7800	50	460
6.	16	6	30	9.0	45	4	10	25	0.300	600	0.25	.30	2700	210	900
7.	16	6	32	10.0	45	6	14	40	0.100	1300	0.21	.30	4500	22	540
8.	20	3	32	6.0	50	6	18	40	0.125	1300	0.21	.30	4500	22	540
9.	22	3	30	10.0	50	8	22	35	0.125	1400	0.25	.26	4500	22	540
10.	8	2	30	10.0	45	8	8	20	0.225	700	0.25	.32	2700	210	900

measurement of the neural network is also done by calculating the mean absolute error(E_j, obtained from Eq. 3) for each data series of the test set after a certain number of training cycle/epochs. The learning process terminates once the predefined convergence criteria i.e., the number of attempts (set to 100 in the present case) without any improvement in E_{mean} (obtained from Eq. 4) is met.

$$E_j = \frac{\sum_{i=1}^{n_o} \left(\frac{O_{a_i} - O_{n_i}}{O_{a_i}} * 100 \right)}{n_o} \qquad (3)$$

$$E_{mean} = \frac{\sum_{j=1}^{n_t} E_j}{n_t} \qquad (4)$$

where, E_j is the mean absolute error for each jth test set, O_a and O_n are the actual and the neural network predicted ith

output, n_o is the total number of outputs and E_{mean} is the mean absolute error for the entire test sets (n_t).

In the current work for analyzing the topological effect on neural network performance different cases were considered. Topologies were changed by varying the number of hidden layers and corresponding number of neurons. Also for analyzing the effect of individual training parameters such as learning rate, momentum, number of epochs and sigmoid gain on the performance of neural network various tests were conducted.

6 Results and discussions

Training and testing of the machining load predictive model with varying topologies, viz., configuration with one hidden layer *to* four hidden layers were first carried out. Figures 4 and 5 shows the influence of learning rate and momentum on the prediction accuracy of a network with three hidden layers. These graphs shows minimal mean absolute error for 72 combinations of nine momentum

Table 6 Validation results from ANN and estimated prediction accuracy

ANN outputs				Desired outputs				Absolute percentage error				E_j (%)
F_x	F_y	F_z	T_{avg}	F_x	F_y	F_z	T_{avg}	F_x	F_y	F_z	T_{avg}	
403	572	98	240	415	589	101	250	2.99	2.83	2.87	4.17	3.215
93	110	16	252	101	113	16	255	7.43	2.4	1.77	1.13	3.183
772	887	129	691	810	854	133	707	4.72	3.91	3.33	2.2	3.540
1093	1443	129	770	1068	1543	138	779	2.34	6.46	6.17	1.13	4.025
2001	2761	534	792	2090	2870	550	767	4.26	3.79	2.94	3.32	3.578
903	1223	277	296	959	1268	289	304	5.81	3.53	4.02	2.62	3.995
1225	1993	546	811	1245	2119	520	825	1.6	5.94	5.06	1.69	3.573
1209	1761	184	871	1248	1723	191	913	3.14	2.18	3.91	4.56	3.448
1868	2783	531	918	1952	2858	548	896	4.33	2.63	3.19	2.44	3.147
1388	1983	334	217	1336	2058	343	226	3.92	3.63	2.56	4.18	3.573

values ranging from 0.1–0.9 and eight learning rates between 0.2–0.9 with a step size of 0.1. Referring to Figs. 4 and 5, it is evident that the the best possible combination of learning rate and momentum for training the considered network is 0.2 and 0.3, respectively. Similar tests were also performed for configurations with one, two and four hidden layers and the obtained best combination of learning rate and momentum values are presented along with the corresponding minimal mean absolute error in Table 4. The network with three hidden layers (ten neurons in each) was found to be the optimal configuration, both in terms of prediction accuracy (mean absolute error within 6.5% for all the cases tested) and computational time (approximately 15 minutes to train and test the network).

In this work, the impact of the variation of the sigmoid gain on neurons activation function was also studied using the configuration with three hidden layers (10 neurons in each), a learning rate of 0.2, a momentum of 0.3 and the number of epochs between tests was set to 5. A total of eight experiments were made using eight different values for the sigmoid gain between 0.2 and 1.6 with a step size of 0.2. In Fig. 6 one can see the influence of sigmoid gain on the prediction accuracy of the network, where worst results i.e., maximum error are found for sigmoid gain values ranging from 0.2–0.6. The best value of the sigmoid gain is 1 where the minimal mean output error is 3.147%. The explanation for the general decrease of the error with the increase of the sigmoid gain is that the search space and the weights initialization corresponds more in this case. Looking at Fig. 7 which shows the sigmoid shapes for different sigmoid gain values ranging from 0.2 to 1.6 with a step size of 0.2, it is evident that the optimal sigmoid gain is adapted to the weights initialization range (−1 to 1 in the present case) and to the search space. Although a sigmoid gain value of 1 is generally a good choice, this study shows that the sigmoid gain choice can lead to improvement in prediction accuracy if adapted to the weights initialization range and to the search space.

Table 5 presents input data sets for ten validation test cases used for checking the prediction accuracy of the developed ANN machining load predictive model. Results for the same are depicted in Table 6.

7 Conclusions

In the current work, a generalized feed-forward neural network architecture was developed for predicting machining loads induced in milling operations. Comparisons between the prediction accuracy of the proposed model with other similar works reported in the literature were made. The obtained minimal absolute error of 3.147% shows its better prediction accuracy over its other counterparts presented in [8, 9, 11] having minimal absolute error ranging from 4–5%

for a maximum number of twelve input parameters and three outputs (mainly three components of cutting forces). The obtained prediction accuracy has successfully demonstrated the capabilities of ANN for modeling nonlinear relationships between large numbers of input-ouput parameters as involved in the machining process.

A Windows based environment has been developed, using which the system can be easily configured to simulate different machining conditions depending upon the availability of training data (obtained either experimentally or numerically using FEM software). Its simple design allows even new users to fully configure the system and make efficient use of the same in an interactive environment.

The functionality of mill-cut can be further enhanced by the inclusion of optimization models such as genetic algorithms(GA) in order to improve cutting tools design (i.e., selection of optimal tool geometrical parameters) and selection of optimal cutting parameters; thus, it can help manufacturers in achieving better machined part quality and improved tool life. In future work an experimental validation of the developed system will be also carried out, where the model will be trained using real field data and its prediction accuracy will be validated against the practical results, so that the accuracy and reliability of the proposed machining load predictive model in solving real field problems can be evaluated.

References

1. Yoon MC, Kim YG (2004) Cutting dynamic force modeling of endmilling operation. J Mater Process Technol 155–156:1383–1389
2. Lazoglu I, Altintas y (2002) Prediction of tool an chip temperature in continuous and interrupted machining. Int J Mach Tools Manuf 42:1011–1022
3. Zheng HQ, Li XP, Wong YS, Nee AYC (1999) Theoretical modeling and simulation of cutting forces in face milling with cutter runout. Int J Mach Tools Manuf 39:2003–2018
4. Li HZ, Zhang WB, Li XP (2001) Modeling of cutting forces in helical end milling using a predictive machining theory. Int J Mech Sci 43:1711–1730
5. Fang G, Zeng P (2005) Three-dimensional thermo-elastic-plastic coupled FEM simulations for metal oblique cutting processes. J Mater Process Technol 168:42–48
6. Stephenson SA, Agapiou JS (2005) Metal cutting theory and practice, 2nd edn. Taylor & Francis, New York
7. Ozel T, Zeren E (2004) Determination of work material flow stress and friction for FEA of machining using orthogonal cutting tests. J Mater Process Technol 153–154:1019–1025
8. Tansel IN, Ozcelik B, Bao WY, Chen P, Rincon D, Yang SY, Yenilmez A (2006) Selection of optimal cutting conditions by using GONNS. Int J Mach Tools Manuf 46:26–35
9. Zuprel U, Cus F, Mursec B, Ploj T (2006) A generalized neural network model of ball-end milling force system. J Mater Process Technol 175:98–108
10. Zuprel U, Cus F (2004) Tool cutting force modeling in ball-end milling using multilevel perceptron. J Mater Process Technol 153–154:268–275

11. Tandon V, El-Mounayri H, Kishawy H (2002) NC end milling optimization using evolutionary computations. Int J Mach Tools Manuf 42:595–605

12. Suneel TS, Pande SS, Date PP. A technical not on integrated product quality model using artificial neural networks. J Mater Process Technol 121:77–86

13. Samuel HH, Zhang HC (1993) Neural networks in manufacturing: a survey. IEEE /CHMT Int. Electronics Mmanufacturing Technology Symposium, 1993, pp 177–190

14. Masters T (1993) Practical neural network recipes in C++. Acdemic Press, New York

15. Sandvik tool catalogue, AB Sandvik Coromant, Sweden, 2001