Application of radial basis function neural networks in optimization of hard turning of AISI D2 cold-worked tool steel with a ceramic tool

S Basak[1], **U S Dixit**[1]*, and **J P Davim**[2]
[1]Department of Mechanical Engineering, Indian Institute of Technology Guwahati, Guwahati, India
[2]Department of Mechanical Engineering, University of Aveiro, Aveiro, Portugal

The manuscript was received on 10 September 2006 and was accepted after revision for publication on 5 March 2007.

DOI: 10.1243/09544054JEM737

Abstract: In this work, the optimization of a finish hard turning process for the machining of D2 steel with ceramic tools is carried out. With the help of replicate experimental data at 27 different cutting conditions, radial basis function neural network models are fitted for predicting the surface roughness and tool wear as functions of cutting speed, feed, and machining time. A novel method for neural network training is proposed. The trained neural network models are used as a black box in the optimization routine. Two types of optimization goal are considered in this work: minimization of production time and minimization of the cost of machining. One novel feature of this work is that the surface roughness is considered in the tool life instead of as a constraint. This is possible owing to the availability of the relationship of surface roughness with time in the neural network model. The results of optimization will be dependent on the tool change time and the ratio of operating cost to tool change cost. The results have been presented for the possible ranges of these parameters. This will help to choose the appropriate process parameters for different situations, and a sensitivity analysis can be easily carried out.

Keywords: hard turning, ceramic tools, surface roughness, tool wear, optimization, neural networks

1 INTRODUCTION

Hard turning is the process of turning materials with a hardness above HRC 45. Ceramic and cubic boron nitride (CBN) tools are widely employed for this purpose. A number of experimental studies have been carried out on the hard turning process using ceramic and CBN tools [**1**–**7**]. From these studies it is inferred that the machining performance in hard turning is highly dependent on the process parameters. With proper optimization of the process parameters, hard turning can compete with the grinding process. However, owing to the rapid development of different tool materials and the need to machine the newer hard materials, the task of optimization becomes challenging on account of lack of information about the cutting performance. A mix of judicious experimental study and optimization technique application is needed for obtaining optimized values of the process parameters.

Considering the difficulties in the development of empirical relations for each and every tool and work–material combination, neural networks have started to gain popularity for machining performance prediction [**8**–**16**]. Ezugwu *et al.* [**8**] used a black oxide ceramic cutting tool for the turning of a grey cast iron. They studied the tool life and failure modes and trained an artificial neural network, which could be used for predicting tool lives and failure modes for different cutting conditions. The accuracy of prediction by the trained network was not very good, and the authors felt the need to carry out more experiments for training the network. Kohli and Dixit [**11**] have proposed a procedure of neural network modelling for surface roughness prediction in turning that requires less experiment data.

**Corresponding author: Mechanical Engineering Department, Indian Institute of Technology Guwahati, Guwahati, Assam 781 039, India. email: uday@iitg.ernet.in*

This procedure is very important as the generation of experimental data is a costly affair. Design of experiments is part of the methodology, and new experimental data are generated in an adaptive manner until a neural network providing satisfactory prediction is obtained. The methodology is less effective when *a priori* experimental data are available with no scope to generate fresh data in an adaptive manner. Ojha and Dixit [**12**] employed neural networks for the prediction of tool life in turning. Özel and Karpat [**13**] utilized neural networks for the prediction of tool wear and surface roughness in hard turning with cubic boron nitride (CBN) tools and compared their performance with multiple-regression models. The neural networks were found to provide better prediction than multiple-regression models. The neural network models used in references [**8**] to [**13**] are multilayer perceptron (MLP) neural networks trained with a backpropagation algorithm. These networks require a large amount of time for training and cannot guarantee to provide global optima in terms of training error minimization. A relatively newer type of neural network, the radial basis function (RBF) neural network, is much faster to train and always provides global minima for minimizing training error for a given architecture. There have been a few applications of RBF neural networks in the prediction of machining performance [**14**–**16**].

Although optimization of the turning process is one of the most widely investigated research topics in machining [**17**, **18**], most of the optimization methods rely on Taylor's tool life equation. Tool life is often determined by measuring the flank wear. Usually, the allowable maximum surface roughness is put as a constraint either using the formula for ideal surface roughness or some empirical expression. The recent trend is to use neural network prediction models in the optimization of machining processes [**19**].

In the present work, radial basis function (RBF) neural networks are employed to predict the surface roughness and flank wear in the finish hard turning of AISI D2 cold-worked tool steel with a mixed ceramic turning tool. The networks are trained on the basis of 54 experiments over wide ranges of cutting speed and feed at a constant depth of cut of 0.2 mm. In order to avoid the problem of overfitting, a novel procedure of network training is used here. The fitted neural network models are used as a black box to predict the surface roughness and flank wear as a function of cutting speed v, feed f, and cutting time t, and are utilized in an optimization routine. In the present work, a tool is considered as failed when either the maximum flank wear exceeds a prescribed value or the centre-line average (CLA) surface roughness R_a of the machined surface exceeds the design value. Trained neural networks being available for predicting the flank wear at the nose (VC) and the surface roughness, the tool life as a function of feed and cutting speed can be computed. Two types of optimization goal are considered in this work: minimization of production time and minimization of the cost of machining. The optimization is carried out by writing a code in a MATLAB environment. The results of optimization are dependent on tool change time and the ratio of operating cost to tool change cost. Hence, the results have been presented for the possible ranges of these parameters.

2 NEURAL NETWORK MODELLING PROCEDURE

In finish hard turning, the depth of cut is kept very small and chosen according to the required final dimension of the job. The cutting speed and feed are two independent variables on which the machining performance depends. The machining performance will also be a function of the cutting time. Therefore, three input neurons of a neural network are cutting speed v, feed f, and cutting time t. Two commonly employed neural networks are the multilayer perceptron (MLP) network and the radial basis function (RBF) network [**20**]. The RBF network requires less time in training, is not dependent on the initial guess for the network weights, and always provides global minima when the minimization of error is carried out using network weights as the design variables. Therefore, this type of network was employed in the present work. An RBF neural network consists of three layers: an input layer, a hidden layer of RBF processing neurons, and an output layer consisting of pure linear processing neurons. The architecture of a typical neural network is shown in Fig. 1. Only one output neuron is shown in the output layer. It is possible to design network architecture with a greater number of neurons in the output layer. However, in this work, two different networks

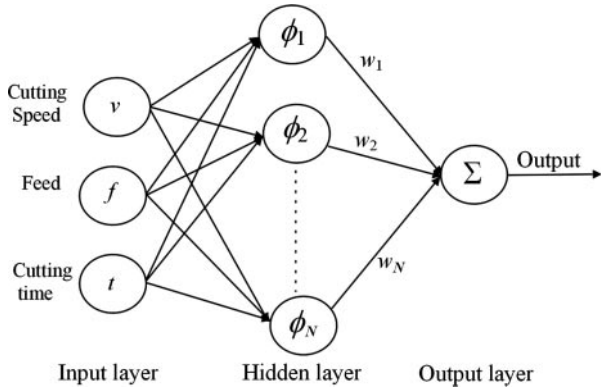

Fig. 1 Typical RBF neural network architecture

with one neuron in the output layer are fitted for predicting output variables VC and R_a. This offers the flexibility to have different types of radial basis processing function in the hidden layer for different networks.

In the present work, the form of the RBF was chosen as Gaussian with the expression

$$\phi(x) = \exp\left(\frac{-x^2}{\sigma^2}\right) \tag{1}$$

where parameter σ controls the zone of influence of the RBF and is commonly referred to as the spread parameter. The output o of a network consisting of N neurons in the hidden layer is calculated according to

$$o = \sum_{i=1}^{N} w_i \phi_i(\|\boldsymbol{x} - c_i\|_2) \tag{2}$$

where $\boldsymbol{x} = \{v,\ f,\ t\}^{\mathrm{T}}$ is the input vector of process variables, $\phi_i(\cdot)$ represents the processing functions of the ith neuron in the hidden layer, w_i is the corresponding weight, c_i is the corresponding RBF centre chosen from the input vector space, and $\|\cdot\|_2$ denotes the Euclidean norm. After fixing the spread parameter and centres for P number of input–output training pairs, the following linear system of equations is obtained

$$o_p = \sum_{i=1}^{N} w_i \phi_i\left(\left\|\boldsymbol{x}_p - c_i\right\|_2\right), \qquad p = 1, 2, \ldots, P \tag{3}$$

The above system of equation can be solved by the least-squares method, which in essence is the following optimization problem with weights as the design variables

$$\text{Minimize } E = \sum_{p=1}^{P}\left[o_p - \sum_{i=1}^{N} w_i \phi_i\left(\left\|\mathbf{x}_p - c_i\right\|_2\right)\right]^2 \tag{4}$$

In this problem, the objective function is convex and therefore the solution is always a global minimum.

Usually, the RBF neural network requires a large number of training pairs. However, often a limited number of training data are available. In this work, the experimental data are taken from the paper by Davim and Figueira [**7**] for the turning of high-chromium cold-worked tool steel D2 (AISI) with mixed alumina ceramic inserts. The workpiece had the following chemical composition (wt %): 1.55C–0.30Si–0.40Mn–11.80Cr–0.80Mo–0.80V. The hardness of the workpiece was about HRC 60. Only 27 experimental data replicated once were available, as shown in Table 1. Training with these limited data may result in a non-smooth and physically unrealistic function. To alleviate this problem, the ranges of cutting speed, feed, and cutting time were divided into two parts. Thus, the entire domain was subdivided into eight cubic cells of equal sizes, as shown in Fig. 2. For each cubic cell, the data at eight corners were available. Then, an interval regression model [**21**] was fitted for each cell, based on the corner data. The input vector $\boldsymbol{x} = \{v,\ f,\ t\}^{\mathrm{T}}$ is mapped to the lower estimate $Y_l(\boldsymbol{x})$ and upper estimate $Y_u(\boldsymbol{x})$ of a dependent variable according to

$$\begin{aligned} Y_l(\boldsymbol{x}) = {} & A_0^l + A_1^l v + A_2^l f + A_3^l t + A_4^l vf \\ & + A_5^l ft + A_6^l tv + A_7^l vft \end{aligned} \tag{5}$$

and

$$\begin{aligned} Y_u(\boldsymbol{x}) = {} & A_0^u + A_1^u v + A_2^u f + A_3^u t + A_4^u vf \\ & + A_5^u ft + A_6^u tv + A_7^u vft \end{aligned} \tag{6}$$

where A_i^l are the coefficients for predicting the lower estimate and A_i^u are the coefficients for predicting the upper estimate of the dependent variable Y. For each cell at each corner, two replicates were available. Thus, a total of 16 data were used for fitting of the interval regression model. The coefficients in equations (5) and (6) can be obtained by solving the following linear programming (LP) problem

$$\begin{aligned} \text{Minimize } \sum_{p=1}^{16} & \left(A_0^u - A_0^l\right) + \left(A_1^u - A_1^l\right) v_p + \left(A_2^u - A_2^l\right) f_p \\ & + \left(A_3^u - A_3^l\right) t_p + \left(A_4^u - A_4^l\right) v_p f_p + \left(A_5^u - A_5^l\right) f_p t_p \\ & + \left(A_6^u - A_6^l\right) t_p v_p + \left(A_7^u - A_7^l\right) v_p f_p t_p \end{aligned} \tag{7}$$

subject to

$$\left.\begin{aligned} & A_0^l + A_1^l v_p + A_2^l f_p + A_3^l t_p + A_4^l v_p f_p \\ & \quad + A_5^l f_p t_p + A_6^l t_p v_p + A_7^l v_p f_p t_p \leqslant Y_p \\ & A_0^u + A_1^u v_p + A_2^u f_p + A_3^u t_p + A_4^u v_p f_p \\ & \quad + A_5^u f_p t_p + A_6^u t_p v_p + A_7^u v_p f_p t_p \geqslant Y_p \end{aligned}\right\}, \quad p = 1, 2, \ldots, 16 \tag{8}$$

The objective of equation (7) is to minimize the widths of the interval outputs. The constraint condition (equation (8)) means that the interval output must include the given output.

For example, consider the cell with the ranges of cutting speed 80–150 m/min, feed 0.1–0.15 mm/rev, and cutting time 5–10 min. Corresponding to this cell, eight different cutting conditions are obtained from the experimental data. Output variable R_a is selected for creating the model. As each cutting condition has two replicates of output variables, 16 sets of input–output datasets can be obtained from Table 1. Thus, for the above LP problem, 16 patterns of input variables $\{v_p,\ f_p,\ t_p\}$ and output variable Y_p are obtained. Solving the above problem yields the coefficients of the equations (5) and (6) corresponding to the particular cell and output variable. This procedure is repeated for each of the eight cells and for each of the output variables. Thus, separate linear regression models are obtained for each cell and each output.

Table 1 Comparison of CLA surface roughness and flank wear predictions by neural networks with the experimental data obtained from reference [7]

Cutting parameters			Flank wear values VC (mm)			Surface roughness values R_a (μm)		
v (m/min)	f (mm/rev)	t (min)	Replicate 1	Replicate 2	NN predicted values	Replicate 1	Replicate 2	NN predicted values
80	0.05	5	0.05	0.06	0.05	0.45	0.40	0.42
80	0.05	10	0.08	0.08	0.08	0.52	0.47	0.51
80	0.05	15	0.11	0.10	0.11	0.57	0.55	0.55
80	0.1	5	0.05	0.05	0.06	0.82	0.62	0.70
80	0.1	10	0.08	0.08	0.08	1.02	0.92	0.91
80	0.1	15	0.09	0.09	0.09	1.14	0.95	1.01
80	0.15	5	0.04	0.03	0.04	0.54	0.55	0.52
80	0.15	10	0.08	0.08	0.08	0.60	0.80	0.71
80	0.15	15	0.10	0.09	0.10	0.74	0.90	0.81
150	0.05	5	0.09	0.11	0.12	0.41	0.36	0.39
150	0.05	10	0.14	0.14	0.17	0.58	0.42	0.50
150	0.05	15	0.16	0.15	0.18	0.62	0.58	0.66
150	0.1	5	0.09	0.08	0.09	0.77	0.80	0.74
150	0.1	10	0.15	0.13	0.15	0.87	0.91	0.91
150	0.1	15	0.26	0.24	0.23	1.15	0.95	1.06
150	0.15	5	0.10	0.09	0.09	0.79	0.75	0.78
150	0.15	10	0.17	0.18	0.16	1.06	1.07	1.04
150	0.15	15	0.24	0.25	0.26	1.27	1.37	1.31
220	0.05	5	0.34	0.33	0.33	0.25	0.26	0.27
220	0.05	10	0.60	0.59	0.58	0.45	0.46	0.52
220	0.05	15	0.65	0.64	0.64	1.31	1.49	1.40
220	0.1	5	0.19	0.21	0.22	0.62	0.55	0.60
220	0.1	10	0.29	0.31	0.31	1.09	0.86	0.97
220	0.1	15	0.31	0.33	0.36	1.33	1.42	1.41
220	0.15	5	0.18	0.20	0.19	0.91	0.85	0.87
220	0.15	10	0.28	0.31	0.33	1.35	1.26	1.29
220	0.15	15	0.75	0.80	0.79	1.46	1.49	1.46

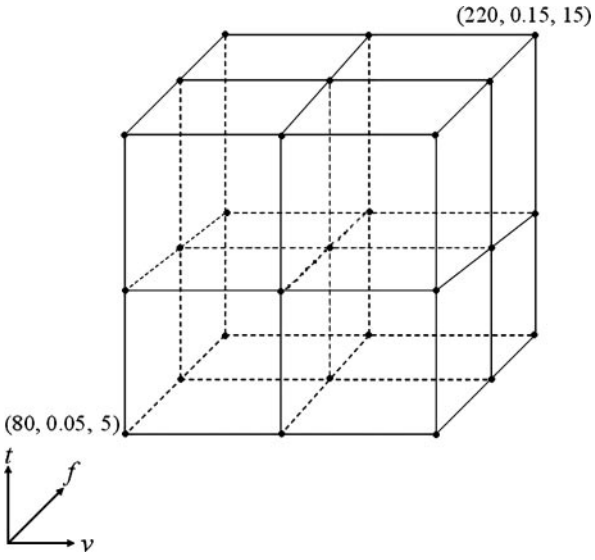

Fig. 2 Division of the input domain into eight cubic cells for fitting of linear regression models

The piecewise linear regression equations for flank wear and surface roughness are provided in Tables 2 and 3 respectively. A subroutine has been written to predict the upper and lower estimates of each of the output variables throughout the domain (i.e. the total range of the process parameters) by unifying all the interval regression models.

The RBF neural network as shown in Fig. 1 is trained with the help of 125 uniformly distributed cutting conditions throughout the range of process parameters. These 125 data points are obtained by choosing five uniformly distributed data points along the range of each process parameter. The interval regression models are used to predict the lower and upper estimates of the dependent variables corresponding to these 125 data points. The mean value of these lower and upper estimates along with the corresponding process parameter values forms the training set for the RBF neural network.

Each of the output variables was predicted though a separate network. Therefore, two neural networks were fitted to predict the two output variables, i.e. surface roughness and flank wear. Apart from being trained by a training dataset, the neural network also requires an appropriate spread parameter to have a proper generalization of the neural model in predicting the values throughout the range. An inappropriate spread parameter may lead to a non-smooth function. For example, Fig. 3 shows the variations in surface roughness R_a with cutting speed for a constant feed of 0.1 mm/rev and a cutting time of 10 min. The three curves correspond to three networks trained with different spread parameters. It is seen that in the speed range 80–220 m/min the spread parameter 1.92 provides a unimodal curve,

Table 2 Regression equations for the eight cubic cells for the prediction of flank wear

Cell	Equation for lower and upper estimates of VC (mm)
$v = 80$–150 m/min, $f = 0.05$–0.1 mm/rev, $t = 5$–10 min	$VC^{l} = -0.014 + 0.094v + 0.034f + 0.021t - 0.094fv + 0.189vt$ $VC^{u} = -0.02 + 0.22v + 0.043f + 0.064t - 0.283fv - 0.116ft - 0.094vt + 0.566vft$
$v = 80$–150 m/min, $f = 0.05$–0.1 mm/rev, $t = 10$–15 min	$VC^{l} = -0.334 + 0.974v + 0.849f + 0.501t - 2.169fv - 1.221ft - 1.131vt + 3.111vft$ $VC^{u} = -0.351 + 0.911v + 0.84f + 0.561t - 1.98fv - 1.311ft - 1.131vt + 3.111vft$
$v = 80$–150 m/min, $f = 0.1$–0.15 mm/rev, $t = 5$–10 min	$VC^{l} = 0.134 - 0.094v - 0.189f - 0.03t + 0.189fv + 0.077ft + 0.283vft$ $VC^{u} = 0.071 - 0.031v - 0.094f - 0.004t + 0.094fv - 0.013ft + 0.094vt + 0.283vft$
$v = 80$–150 m/min, $f = 0.1$–0.15 mm/rev, $t = 10$–15 min	$VC^{l} = 0.505 - 1.226v - 0.411f - 0.587t + 1.131fv + 0.411ft + 1.697vt - 1.131vft$ $VC^{u} = 0.546 - 1.226v - 0.505f - 0.715t + 1.226fv + 0.604ft + 1.886vt - 1.414vft$
$v = 150$–220 m/min, $f = 0.05$–0.1 mm/rev, $t = 5$–10 min	$VC^{l} = 0.05 - 0.223f - 2.229t + 0.283fv + 3.086ft + 3.489vt - 4.526vft$ $VC^{u} = 0.301 - 0.251v - 0.664f - 2.7t + 0.754fv + 3.934ft + 3.96vt - 5.374vft$
$v = 150$–220 m/min, $f = 0.05$–0.1 mm/rev, $t = 10$–15 min	$VC^{l} = -0.527 + 1.257v - 0.437f - 1.363t - 0.283fv + 3.407ft + 1.603vt - 3.677vft$ $VC^{u} = -0.716 + 1.446v - 0.124f - 1.174t - 0.566fv + 3.407ft + 1.603vt - 3.677vft$
$v = 150$–220 m/min, $f = 0.1$–0.15 mm/rev, $t = 5$–10 min	$VC^{l} = -0.016 + 0.126v - 0.124f - 0.737t + 0.094fv + 0.849ft + 1.037vt - 0.849vft$ $VC^{u} = -0.164 + 0.314v + 0.034f - 0.326t - 0.094fv + 0.373ft + 0.566vt - 0.283vft$
$v = 150$–220 m/min, $f = 0.1$–0.15 mm/rev, $t = 10$–15 min	$VC^{l} = -5.473 + 7.543v + 6.981f + 7.449t - 9.711fv - 9.81ft - 10.089vt + 13.86vft$ $VC^{u} = -5.519 + 7.669v + 7.08f + 7.706t - 9.9fv - 10.196ft - 10.466vt + 14.426vft$

Table 3 Regression equations for the eight cubic cells for the prediction of CLA surface roughness

Cell	Equation for lower and upper estimates of R_a (μm)
$v = 80$–150 m/min, $f = 0.05$–0.1 mm/rev, $t = 5$–10 min	$R_a^{l} = 0.809 - 1.288v - 1.332f - 1.097t + 3.582fv + 4.024ft + 1.697vt - 5.374vft$ $R_a^{u} = 0.54 - 1.1v - 1.174t + 1.98fv + 3.124ft + 2.734vt - 5.374vft$
$v = 80$–150 m/min, $f = 0.05$–0.1 mm/rev, $t = 10$–15 min	$R_a^{l} = 0.069 - 0.849v + 1.444f + 0.013t + 0.566fv - 0.141ft + 1.037vt - 0.849vft$ $R_a^{u} = -0.523 + 1.603v + 2.554f + 0.42t - 4.054fv - 0.707ft - 1.32vt + 3.677vft$
$v = 80$–150 m/min, $f = 0.1$–0.15 mm/rev, $t = 5$–10 min	$R_a^{l} = -1.291 + 3.551v + 1.817f + 6.111t - 3.677fv - 6.789ft - 10.371vt + 12.728vft$ $R_a^{u} = 1.5 - 0.66v - 1.44f + 1.431t + 1.32fv - 0.784ft - 3.111vt + 3.394vft$
$v = 80$–150 m/min, $f = 0.1$–0.15 mm/rev, $t = 10$–15 min	$R_a^{l} = 3.186 - 3.426v - 3.231f - 0.604t + 4.431fv + 0.784ft + 0.094vt + 0.566vft$ $R_a^{u} = 2.043 - 2.483v - 1.294f + 0.617t + 2.074fv - 1.003ft - 0.377vt + 2.263vft$
$v = 150$–220 m/min, $f = 0.05$–0.1 mm/rev, $t = 5$–10 min	$R_a^{l} = 0.08 - 0.22v + 2.267f - 0.39t - 1.697fv - 0.99ft + 0.66vt + 1.98vft$ $R_a^{u} = -0.596 + 0.566v + 3.664f + 2.619t - 3.394fv - 6.904ft - 2.829vt + 9.334vft$
$v = 150$–220 m/min, $f = 0.05$–0.1 mm/rev, $t = 10$–15 min	$R_a^{l} = 3.669 - 6.129v - 1.899f - 5.773t + 5.469fv + 5.259ft + 9.523vt - 8.769vft$ $R_a^{u} = 9.57 - 13.2v - 13.71f - 12.63t + 19.8fv + 19.157ft + 17.82vt - 25.457vft$
$v = 150$–220 m/min, $f = 0.1$–0.15 mm/rev, $t = 5$–10 min	$R_a^{l} = 3.894 - 4.054v - 3.454f - 3.724t + 4.054fv + 4.011ft + 4.054vt - 3.111vft$ $R_a^{u} = 4.35 - 4.84v - 3.754f - 5.575t + 4.714fv + 5.387ft + 7.166vt - 5.657vft$
$v = 150$–220 m/min, $f = 0.1$–0.15 mm/rev, $t = 10$–15 min	$R_a^{l} = 6.871 - 8.831v - 6.703f - 8.19t + 9.523fv + 8.884ft + 11.22vt - 11.314vft$ $R_a^{u} = 2.921 - 3.771v - 3.737f - 3.433t + 5.657fv + 5.361ft + 5.563vt - 7.071vft$

whereas other spread parameters provide multimodal curves, which seems to be physically unrealistic. The neural network is seen to overestimate surface roughness at low spread parameters and underestimate the surface roughness at high spread parameters. Therefore, a proper spread parameter should be chosen to have the best-fit neural network providing the least r.m.s. percentage error for the testing and training data. A total of uniformly distributed 316 data points were chosen as the testing data. The MATLAB function NEWRB was used for training of the network. The function uses an algorithm that starts with zero neurons in the hidden layer and keeps on adding the input vector with the greatest error as the centre in successive runs until the error goal is achieved. The following modifications were carried out to this function. First, the maximum number of neurons in the hidden layer was limited to 32 in order to limit the number of computations and the memory requirement for the fitted function. Second, the architecture providing the minimum effective r.m.s. percentage error (the minimum training and testing error) was chosen among 32 different architectures corresponding to 1–32 centres (neurons in the hidden layer), instead of prescribing any error goal.

The appropriate spread parameter was found for each of the networks (corresponding to each output variable) by minimizing the effective error (the maximum of training and testing error) subject to the constraint that all the predictions lie within the lower and upper estimates of the regression model. For each prediction lying outside the upper and lower bound, a penalty (a high value, 10 in this case) is added to the effective error. The sum of penalties and effective error is referred to as the penalized effective error e in subsequent discussion. The penalized effective error is minimized by

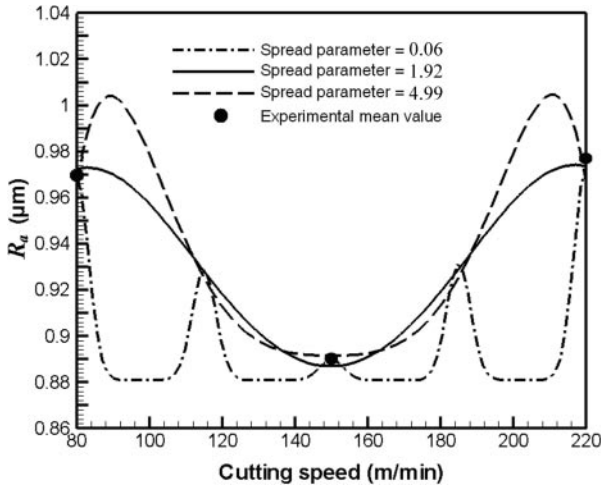

Fig. 3 Effect of the spread parameter on the model

optimizing the spread parameter. One-dimensional optimization for the spread parameter was carried out using a one-dimensional search algorithm called the golden section search [**22**]. The algorithm works as follows.

Step 1. Obtain the penalized effective errors for the networks trained with the spread parameters $\sigma_1 = 0.01$ and $\sigma_2 = 5$. Denote the respective penalized effective errors as e_1 and e_2. The values of σ_1 and σ_2 are chosen on the basis that the spread parameter providing the minimum effective error lies in between these values.

Step 2. Obtain $\sigma_3 = \tau\sigma_1 + (1-\tau)\sigma_2$ and $\sigma_4 = \tau\sigma_2 + (1-\tau)\sigma_1$, where τ is called the golden number and is equal to 0.618. Obtain the penalized effective errors e_3 and e_4 corresponding to these parameters. In the first iteration, both e_3 and e_4 are to be computed. In the subsequent iterations, one value will always be available and only one value needs to be computed.

Step 3. If $e_3 < e_4$, then $\sigma_1 = \sigma_4$ and $e_1 = e_4$. Else if $e_4 < e_3$, then $\sigma_2 = \sigma_3$ and $e_2 = e_3$.
Else if $e_3 = e_4$, then $\sigma_1 = \sigma_4$, $e_1 = e_4$, $\sigma_2 = \sigma_3$, and $e_2 = e_3$.

Step 4. Is $|\sigma_1 - \sigma_2| < 0.005$? If no, go to step 2. Else stop the program and take the optimum spread parameter as $(\sigma_1 + \sigma_2)/2$.

In this algorithm the interval $|\sigma_1 - \sigma_2|$ is reduced $(0.618)^{n-1}$ times after training the networks with n different spread parameters. Thus, the convergence is achieved in 24 iterations for one network architecture specified by the number of neurons in the hidden layer. The total CPU time required for the network training in a Pentium IV computer equipped with MATLAB is about 400 s. The results of the neural network modelling are discussed in the next section.

3 RESULTS OF NEURAL NETWORK MODELLING

The optimum architecture of the neural networks for the prediction of surface roughness and tool wear has 29 neurons. The optimum spread parameter for the network predicting the flank wear is 1.46, and for the network predicting the surface roughness it is 1.92. The r.m.s. percentage error for the 125 training data is 15.18 and 6.63 per cent for the prediction of flank wear and surface roughness respectively. The corresponding errors for 316 testing data are 15.38 and 6.31 per cent respectively. These errors are calculated on the basis of the mean values of the experimental results as well as the piecewise linear regression model. The experimental results have scatter. Therefore, these errors are reasonable.

An overall regression model was fitted on the complete domain of process parameters, based on the experimental results, using the following set of equations

$$\mathrm{VC} = C_1 v^{a_1} f^{b_1} t^{d_1} \tag{9}$$

$$R_\mathrm{a} = C_2 v^{a_2} f^{b_2} t^{d_2} \tag{10}$$

These equations can be transformed into the following linear form

$$\ln \mathrm{VC} = \ln C_1 + a_1 \ln v + b_1 \ln f + d_1 \ln t \tag{11}$$

$$\ln R_\mathrm{a} = \ln C_2 + a_2 \ln v + b_2 \ln f + d_2 \ln t \tag{12}$$

A total of 27 sets of experimental data were available. The mean of two experimental replicates was employed for finding the unknowns in equations (11) and (12) using a multiple linear regression model, which yielded the following equations

$$\mathrm{VC} = 0.4340 v^{1.5668} f^{-0.1226} t^{0.7307} \tag{13}$$

$$R_\mathrm{a} = 1.4816 v^{0.2275} f^{0.5922} t^{0.5377} \tag{14}$$

The results of the predictions made by the overall regression model and neural network model are shown in Figs 4 and 5 for flank wear (VC) and surface roughness, R_a, respectively, for 27 experimental data. Figure 4 shows the lower and upper experimental values of flank wear, flank wear predicted by the overall regression model, and neural network predicted flank wear for 27 experimental data. In most of the cases, the neural network values lie within the lower and upper experimental values. The lower and upper experimental values of surface roughness, surface roughness predicted by the overall regression model, and neural network predicted surface roughness are shown in Fig. 5. In this case also, most of the predictions by neural network lie in between the two replicate values of the experimental data. The neural network predictions are

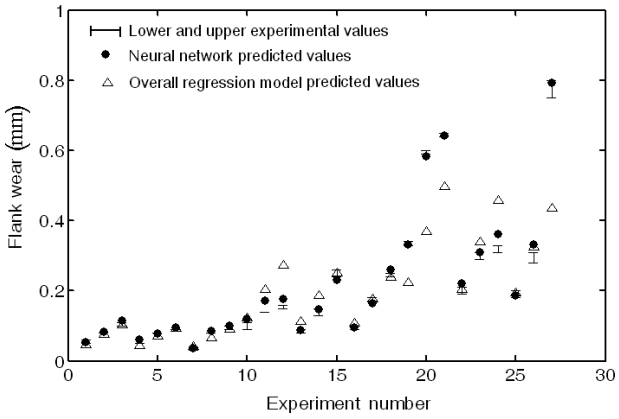

Fig. 4 Comparison of neural network and overall regression model predicted values for the tool flank wear with the experimental values

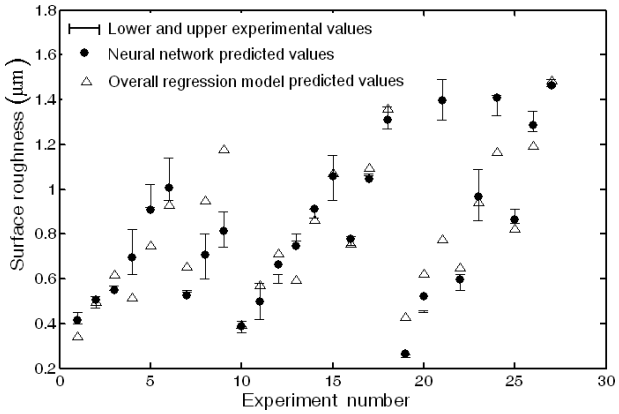

Fig. 5 Comparison of neural network and overall regression model predicted values for the surface roughness with the experimental values

seen to have smaller deviation outside the lower and upper experimental values compared with the overall regression model.

The r.m.s. percentage errors from the mean of experimental replicates have been calculated. The r.m.s. percentage errors for the neural network predicted flank wear and surface roughness from the mean of experimental values are 9.3 and 4.21 per cent respectively. The same errors for the predictions by the overall regression model for flank wear and surface roughness are 26.48 and 23.66 per cent respectively. The deviation of the neural network predicted flank wear values outside the lower and upper experimental values is 7.23 per cent, and the same deviation for surface roughness is 3.01 per cent. The deviations for predictions by the overall regression model outside the lower and upper experimental values are 23.93 per cent for flank wear and 19.85 per cent for surface roughness. The coefficients of determination for flank wear and surface roughness predicted by the overall regression model are 73.63 and 70.81 per cent respectively, while for the predictions by the neural network model the coefficients of determination are 99.28 and 99.35 per cent for flank wear and surface roughness respectively.

With this discussion, it is clear that the neural network predictive model has smaller deviation from experimental values in comparison with the overall regression model. Although piecewise regression models in combination provide similar results, they do not yield a continuous function and require more memory and computation time for the prediction compared with the neural network model. Therefore, the neural network model is used as a black box for optimization in this work. Contours of different outputs with varying cutting speed and feed in the experimental range at a particular cutting time are studied with the help of fitted neural network predictive models.

Figure 6 shows the contours of flank wear after cutting times of 5, 10, and 15 min. It is observed that at high speed, after a cutting time of 5 and 10 min, the wear is greater with a lower feed than with a higher feed. The reason for this is that, at lower feed, chips do not form properly and the cutting action is mainly rubbing. Once the tool has been sufficiently worn, the excessive heat generation at a combination of high feed and high speed dominates, leading to high wear at higher feeds. This is clearly seen from the contour after a cutting time of 15 min.

The contours of surface roughness after different cutting times are shown in Fig. 7. It can be seen that the surface roughness increases with time. In the beginning, the surface roughness is lowest in the low-feed and high-speed zone. However, after a sufficient cutting time has elapsed, the lowest surface roughness is obtained in the low-feed and moderate-speed zone. This is due to a low wear rate in the moderate speed range.

4 OPTIMIZATION PROCEDURE

In this work, two objectives are considered: minimization of the production time and minimization of cost. In finish turning, the time of producing a workpiece is given by

$$T_{\mathrm{p}} = \frac{\pi LD}{fv}\left(1 + \frac{t_{\mathrm{c}}}{T}\right) \tag{15}$$

where L is the machining length, D is the diameter of the job, t_{c} is the tool change time, and T is the tool life. The cost of machining a job is given by

$$C = C_0 \frac{\pi LD}{fv}\left(1 + \frac{t_{\mathrm{c}}}{T}\right) + C_{\mathrm{t}} \frac{\pi LD}{fvT} \tag{16}$$

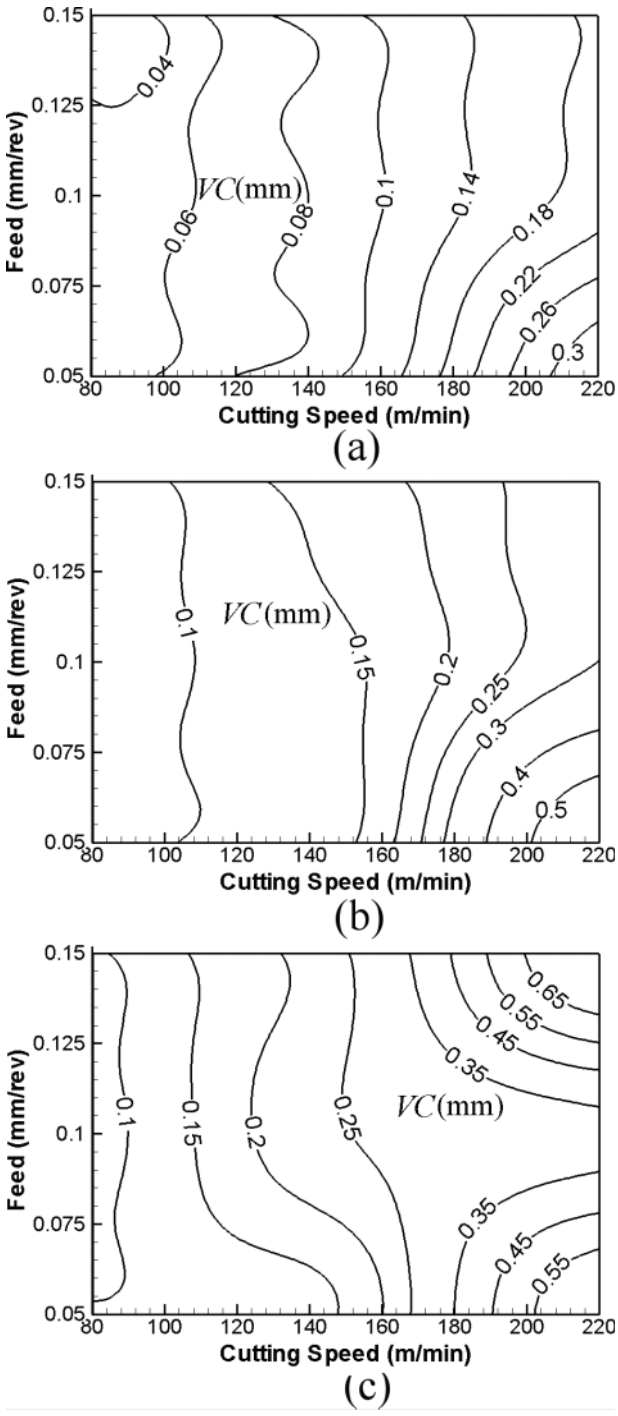

Fig. 6 Contours of tool flank wear after a machining time of (a) 5 min, (b) 10 min, and (c) 15 min

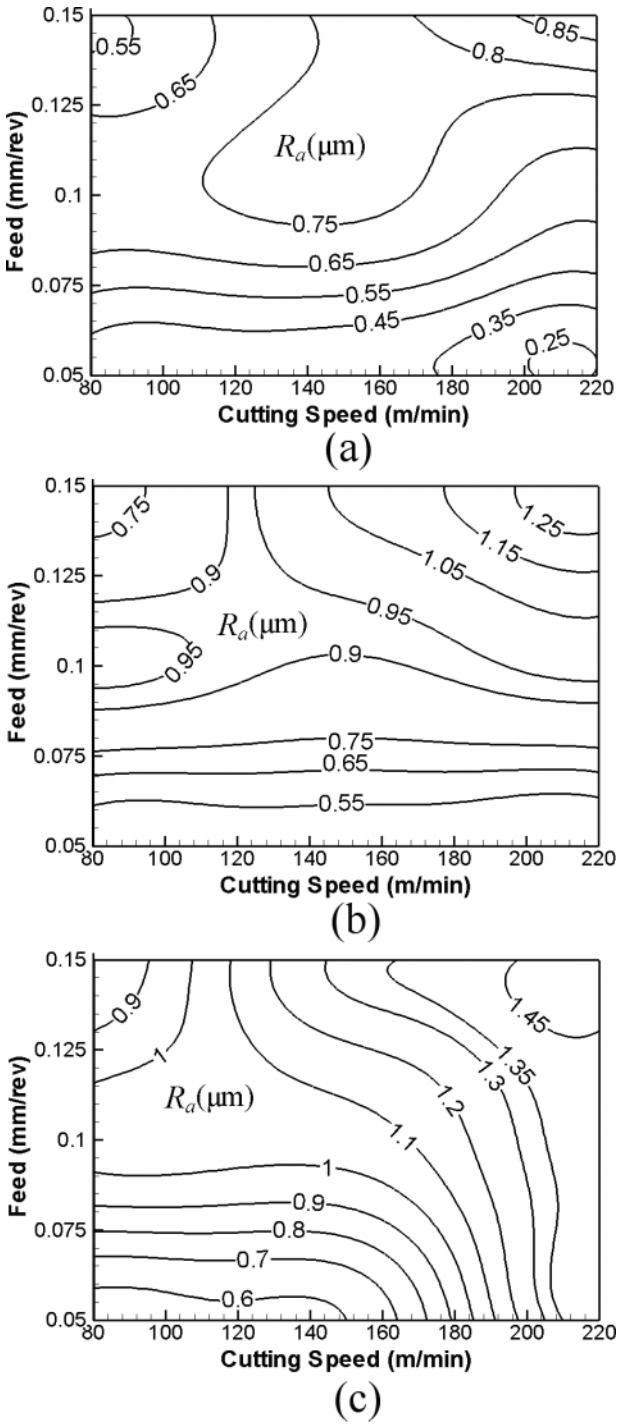

Fig. 7 Contours of surface roughness after a machining time of (a) 5 min, (b) 10 min, and (c) 15 min

where C_0 is the operating cost in \$/min and C_t is the tool change cost in \$.

Equation (16) can be written as

$$C = C_0 \frac{\pi LD}{fv}\left(1 + \frac{t_c + C_t/C_0}{T}\right) \tag{17}$$

The optimization problem can be the minimization of either T_p or C. As C_0, L, and D are constants, the objective function for both optimization problems can be expressed as [**23**]

$$\text{Minimize } F = \frac{1}{fv}\left(1 + \frac{t_c^*}{T}\right) \tag{18}$$

where

$$t_c^* = \begin{cases} t_c & \text{for minimum production time} \\ t_c + C_t/C_0 & \text{for minimum cost of machining} \end{cases} \tag{19}$$

Both optimization problems are subjected to the constraints of cutting forces and cutting power. However, considering that the machine is rigid enough to withstand the forces and power in finish turning, these constraints are not considered here. Only the bounds on f and v are considered. Equation (18) contains the addition of two terms $1/(fv)$ and $t_c^*/(fvT)$. In both terms, increasing fv reduces the objective function value. At a particular value of fv, tool life is dependent on the value of f or v. Therefore, the maximum possible tool life is obtained by golden section search [**22**], the algorithm for which is the same as that discussed in section 2. It is generally seen that an increase in fv reduces the maximum possible tool life T at that fv. If, as a result of an increase in fv, the term fvT increases, then the objective function will decrease irrespective of t_c^*. However, if an increase in fv decreases fvT, then the objective function may increase or decrease depending on the value of t_c^* .

Let the highest possible fv be denoted by x_1, and the corresponding maximum possible tool life be denoted by T_1. If any other fv is denoted by x_2, and the maximum possible tool life is denoted by T_2, then this condition will yield a smaller value of the objective function F than that at the highest fv provided that

$$\frac{1}{x_2}\left(1+\frac{t_c^*}{T_2}\right) < \frac{1}{x_1}\left(1+\frac{t_c^*}{T_1}\right) \tag{20}$$

This gives

$$t_c^* > \frac{T_1 T_2 (x_1 - x_2)}{T_2 x_2 - T_1 x_1} \tag{21}$$

The meaning of the above inequality is that, as long as the value of t_c^* is greater than the right-hand side term, reducing fv will reduce the production time. When this expression equals t_c^*, reduction of fv should be stopped. Therefore, for different fv, an expression can be calculated that can be considered to be the value of t_c^* for which this fv is optimum.

Following this procedure, a curve can be drawn between t_c^* and the optimum fv. At a fixed fv there will be one optimum value of v, and thus another curve can be plotted between fv and the optimum v. The first curve can be used to obtain the optimum fv for a given t_c^*, and the second can be used to obtain the corresponding v (and f).

In this work, the tool life is the minimum of two machining times, one producing a flank wear of 0.4 mm and the other a surface roughness greater than a prescribed value. The fitted neural network model can predict the machining time to cause a specified wear or surface roughness. However, neural networks are not good at extrapolation. Therefore, for a machining time shorter than 15 min the prediction was carried out by the trained neural network, and for a machining time longer than 15 min a linear extrapolation was used.

Figure 8 shows the plots of fv and maximum possible tool life T for surface roughness values R_a of 0.8 and 1.4 μm respectively. It can be seen that in general the tool life decreases with increasing fv. Tool life will be shorter when the surface finish requirement is stringent. However, the difference in tool lives for the two surface roughness requirements decreases with increasing fv, because at high fv the failure due to flank wear becomes the deciding factor.

Figure 9 shows the optimum fv versus the specified t_c^* for two different desired surface roughness values. For $R_a \leqslant 0.8\,\mu m$, the optimum fv continuously decreases with increasing t_c^*. As the value of t_c^* increases, the sensitivity of the solution with t_c^* decreases. This also implies that the solution for maximum production rate and minimum cost will not differ much at high t_c^*. The curve is discontinuous for $R_a \leqslant 1.4\,\mu m$. In this case, for low t_c^*, fv is equal to 33 000 mm^2/min and for high t_c^* it is 26 200 mm^2/min. For low as well as high t_c^*, the results are not sensitive to t_c^*. This is because at $fv = 33\,000$ mm^2/min, both f and v reach their respective upper limits of 0.15 mm/rev and 220 m/min and cannot be increased further with decreasing t_c^*. Similarly, for $fv = 26\,200$ mm^2/min, the optimum feed reaches the upper limit of 0.15 mm/rev.

For $R_a \leqslant 0.8\,\mu m$, the fv can be selected that corresponds to $t_c^* = t_c$ for minimum production time. Let this fv be called $(fv)_1$. If C_t/C_0 is relatively small in comparison with tool change time t_c, then $t_c^* = t_c + C_t/C_0 \approx t_c$. Thus, the production cost is also minimum at $(fv)_1$. If C_t/C_0 is comparable with t_c, then minimum production time corresponding to $t_c^* = t_c$ is obtained at $(fv)_1$, but minimum production cost corresponding to $t_c^* = t_c + C_t/C_0$ is obtained at a different fv. Let this fv be denoted by $(fv)_2$. Thus,

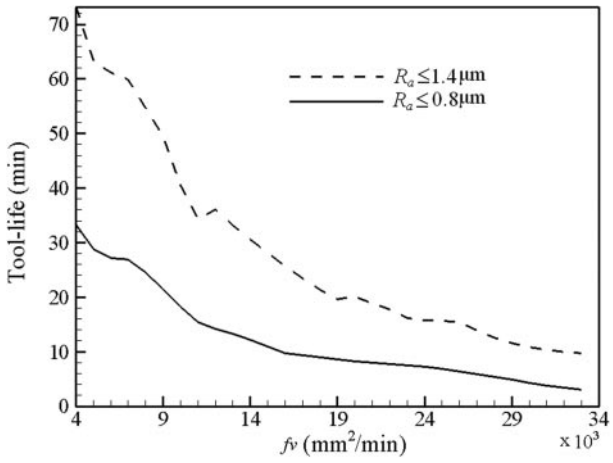

Fig. 8 Variation in tool life with fv

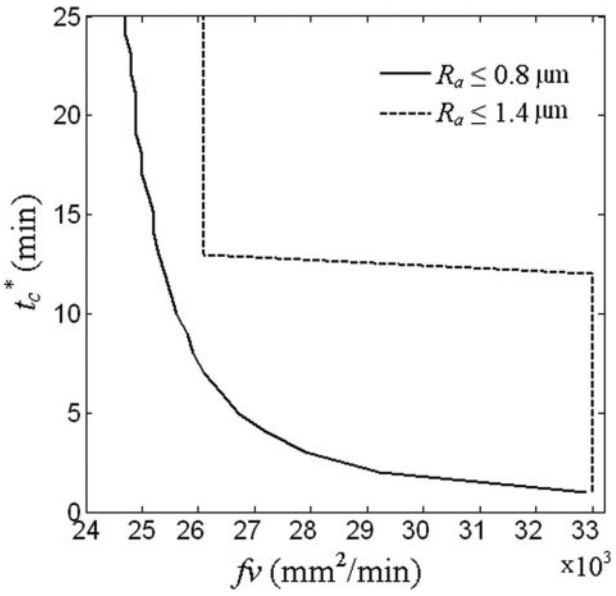

Fig. 9 fv versus t_c^*

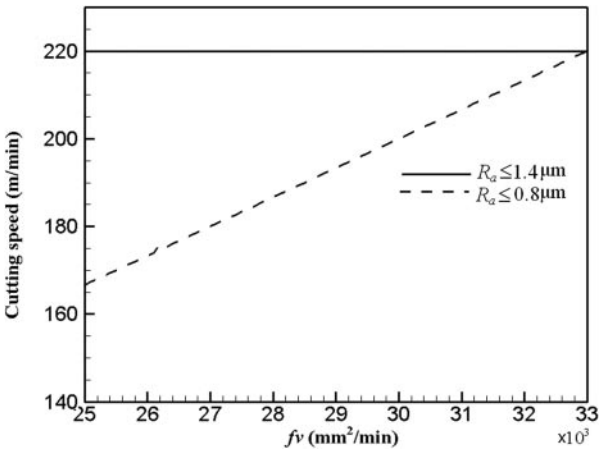

Fig. 10 fv versus cutting speed

while moving from $(fv)_1$ to $(fv)_2$, the production time increases from its minimum and the production cost decreases to its minimum. All the optimum solutions lying on the curve shown in Fig. 9 between $(fv)_1$ and $(fv)_2$ are Pareto optimal solutions, which form the set of non-dominated solutions [**22**]. Two solutions are called non-dominating or Pareto optimal if any one solution is not better than the other solution if all the optimization goals are considered. Here, between the solutions for minimum production time and minimum production cost, many solutions are obtained that are better than the other solutions from the point of view either of production time or of production cost. A higher level of decision is required to choose among these solutions. No such Pareto optimal solutions are obtained for $R_a \leqslant 1.4\,\mu$m.

Figure 10 shows the optimum cutting speed for each fv. Knowing v, it is possible to calculate f from this figure as well. For $R_a \leqslant 0.8\,\mu$m, the cutting speed is constant at 220 m/min, the highest possible in the range of experimental study. Depending on t_c^*, the optimum f is adjusted between 0.11 and 0.15 mm/rev. However, for $R_a \leqslant 1.4\,\mu$m, the reverse is the case. Here, the optimum f remains constant at 0.15 mm/rev, and the optimum cutting speed is adjusted between 167 and 220 m/min. It can be seen that in this problem the upper limits on feed and velocity become the deciding factor. Similar observations were made by Ojha and Dixit [**24**] in finish turning optimization using a genetic algorithm and a surface roughness prediction model, although the tool life aspect was not considered in their work.

Finally, it should be mentioned that, barring a few exceptions, such as reference [**24**], in most of the papers on optimization of the turning process, only a hypothetical example has been chosen for testing of the optimization algorithm. Also, the surface roughness has been incorporated as a time-independent constraint using the highly idealized formula

$$R_a = \frac{f^2}{32R} \tag{22}$$

where R is the nose radius of the tool. In the present work, the effect of time on surface roughness has been considered on the basis of shop-floor experiments. Moreover, the solutions are presented graphically for a range of t_c^*, by using which the user can obtain the appropriate solution for minimum production time as well as cost by supplying the data on tool change time and manufacturing costs.

5 CONCLUSIONS

In the present work, RBF neural network models have been fitted for machining performance prediction. A novel method for training the neural networks has been introduced. The trained neural networks are used to optimize the process parameters. The optimization problem has been discussed in a general way for any general tool change time and cost data. The results of the optimization have been presented graphically.

The optimization results show that, in order to obtain a surface roughness value of less than 0.8 μm, the machining has to be carried out at 220 m/min. The feed can range from 0.11 to 0.15 mm/min, depending on the tool change time for obtaining the maximum production rate. For obtaining the minimum cost of machining, the feed has to be adjusted depending on the sum of the ratio of operating cost

to tool change cost, C_t/C_0, and tool change time t_c. If C_t/C_0 is small in comparison with t_c, it is possible to optimize for maximum production rate alone. However, if this is not the case, then the solutions lying between t_c and $t_c + C_t/C_0$ form the Pareto optimal solutions and a higher-level decision is needed to choose one among these solutions.

For a surface roughness value of less than 1.4 μm, the optimum feed is 0.15 mm/rev and the speed is 220 m/min for low t_c^* and 167 m/min for high t_c^*. In this case, few Pareto-optimal solutions are obtained. Thus, except near the transition t_c^* from low to high (at about 12–13 min), the solution is not sensitive to t_c^*.

Knowing the zone of optimized solutions, a more refined model can be developed by concentrating on this zone. A greater number of experiments in this zone will yield a more accurate model. Thus, the procedure helps in achieving the goal of economic production.

ACKNOWLEDGEMENTS

The experimental work was carried out at University of Aveiro under the supervision of the third author. The authors acknowledge the help received from Paulo Matos and contributions made by L. Figueira. The authors also wish to acknowledge the anonymous reviewers for their valuable suggestions.

REFERENCES

1 **Diniz, E., Gomes, D. M.,** and **Braghini Jr, A.** Turning of hardened steel with interrupted and semi-interrupted cutting. *J. Mater. Processing Technol.*, 2005, **159**, 240–248.
2 **Poulachon, G., Albert, A., Schluraff, M.,** and **Jawahir, I. S.** An experimental investigation of work material microstructure effects on white layer formation in PCBN hard turning. *Int. J. Mach. Tools and Mf.*, 2005, **45**, 211–218.
3 **Grzesik, W.** and **Wanat, T.** Comparative assessment of surface roughness produced by hard machining with mixed ceramic tools including 2D and 3D analysis. *J. Mater. Processing Technol.*, 2005, **169**, 364–371.
4 **Lima, J. G., Avila, R. F., Abrao, A. M., Faustino, M.,** and **Davim, J. P.** Hard turning: AISI 4340 high strength low alloy steel and AISI D2 cold work tool steel. *J. Mater. Processing Technol.*, 2005, **169**, 388–395.
5 **Pavel, R., Marinescu, I., Deis, M.,** and **Pillar, J.** Effect of tool wear on surface finish for a case of continuous and interrupted hard turning. *J. Mater. Processing Technol.*, 2005, **170**, 341–349.
6 **Jiang, W., Malshe, A. P.,** and **Goforth, R. C.** Cubic boron nitride (CBN) based nanocomposite coatings on cutting inserts with chip breakers for hard turning applications. *Surf. and Coatings Technol.*, 2005, **200**, 1849–1854.
7 **Davim, J. P.** and **Figueira, L.** Machinability evaluation in hard turning of cold work tool steel (D2) with ceramic tools using statistical techniques. *Mater. and Des.*, 2006, **28**, 1186–1191.
8 **Ezugwu, E. O., Arthur, S. J.,** and **Hines, E. L.** Tool-wear prediction using artificial neural networks. *J. Mater. Processing Technol.*, 1995, **49**, 255–264.
9 **Azouzi, R.** and **Guillot, M.** On-line prediction of surface finish and dimensional deviation in turning using neural network based sensor fusion. *Int. J. Mach. Tools and Mf.*, 1997, **37**, 1201–1217.
10 **Risbood, K. A., Dixit, U. S.,** and **Sahasrabudhe, A. D.** Prediction of surface roughness and dimensional deviation by measuring cutting forces and vibrations in turning process. *J. Mater. Processing Technol.*, 2003, **132**, 203–214.
11 **Kohli, A.** and **Dixit, U. S.** A neural-network-based methodology for the prediction of surface roughness in a turning process. *Int. J. Advd Mfg Technol.*, 2004, **25**, 118–129.
12 **Ojha, D. K.** and **Dixit, U. S.** An economic and reliable tool life estimation procedure for turning. *Int. J. Advd Mfg Technol.*, 2005, **26**, 726–732.
13 **Özel, T.** and **Karpat, Y.** Predictive modeling of surface roughness and tool wear in hard turning using regression and neural networks. *Int. J. Mach. Tools and Mf.*, 2005, **45**, 467–479.
14 **Briceno, J. F., El-Mounayri, H.,** and **Mukhopadhyay, S.** Selecting an artificial neural network for efficient modeling and accurate simulation of the milling process. *Int. J. Mach. Tools and Mf.*, 2002, **42**, 663–674.
15 **Lin, J. T., Bhattacharyya, D.,** and **Kecman, V.** Multiple regression and neural network analyses in composite machining. *Composite Sci. and Technol.*, 2003, **63**, 539–548.
16 **Sonar, D. K., Dixit, U. S.,** and **Ojha, D. K.** The application of radial basis function neural network for predicting the surface roughness in a turning process. *Int. J. Advd Mfg Technol.*, 2006, **27**, 661–666.
17 **Satishumar, S., Asokan, P.,** and **Kumanan, S.** Optimization of depth of cut in multipass turning using nontraditional optimization techniques. *Int. J. Advd Mfg Technol.*, 2006, **29**, 230–238.
18 **Abburi, N. R.** and **Dixit, U. S.** Multi-objective optimization of multipass turning processes. *Int. J. Advd Mfg Technol.*, 2007, **32**, 902–910.
19 **Ozcelik, B., Oktem, H.,** and **Kurtaran, H.** Optimum surface roughness in end milling Inconel 718 by coupling neural network model and genetic algorithm. *Int. J. Advd Mfg Technol.*, 2005, **27**, 234–241.
20 **Ham, F. M.** and **Kostanic, I.** *Principles of neurocomputing for science and engineering,* 2001 (McGraw-Hill, New York).
21 **Tanaka, H.** Fuzzy data analysis by possibilistic linear models. *Fuzzy Sets Syst.*, 1987, **24**, 363–375.
22 **Rao, S. S.** *Engineering optimization – theory and practice,* 1996 (Wiley and New Age International (P), New Delhi).
23 **Kiliç, S. E.** Use of one-dimensional search method for the optimization of turning operations. *Model. Simul. Control B,* 1987, **14**, 39–63.

24 Dixit, U. S. and **Ojha, D. K.** Finish turning process optimization with genetic algorithm and a neural network based surface roughness prediction model. In Proceedings of 7th Japan–India Joint Seminar on *Advanced manufacturing systems*, Machida City, Tokyo, Japan, 16–21 February 2004.

APPENDIX

Notation

a_1, b_1, d_1	exponents of cutting speed, feed, and cutting time in the overall regression model predicting flank wear
a_2, b_2, d_2	exponents of cutting speed, feed, and cutting time in the overall regression model predicting surface roughness
A_i^{l}	constants for predicting the lower estimate of a dependent variable
A_i^{u}	constants for predicting the upper estimate of a dependent variable
c_i	RBF centre associated with the ith neuron
C	total cost of finish turning of the workpiece (\$)
C_{o}	operating cost of finish turning (\$/min)
C_{t}	tool cost (\$)
C_1	constant in the overall regression model predicting flank wear
C_2	constant in the overall regression model predicting surface roughness
D	diameter of the workpiece (mm)
e	penalized effective error
f	feed per revolution (mm/rev)
L	length of the workpiece (mm)
N	number of hidden neurons
o	output of the neural network
P	number of input–output training pairs
R	tool nose radius (mm)
R_{a}	centre–line average surface roughness (μm)
t	cutting time (min)
t_{c}	tool change time (min)
t_{c}^*	effective tool change time in equation (19)
T	tool life for finish turning (min)
T_{p}	time of finish turning of each workpiece (min)
v	cutting speed (m/min)
VC	tool flank wear (mm)
w_i	weights associated with the ith neuron
$\boldsymbol{x}$	input vector of process variables
Y	output variable
Y_{l}	lower estimate of the output variable
Y_{u}	upper estimate of the output variable
σ	spread parameter
τ	golden number
$\phi(\cdot)$	processing function in equation (1)