Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 00 (2019) 000–000

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

47th SME North American Manufacturing Research Conference, NAMRC 47, Pennsylvania, USA

A neural network approach for chatter prediction in turning

Harish Cherukuri[a], E. Perez-Bernabeu[b], M.A. Selles[b], Tony L. Schmitz[a]*

[a]University of North Carolina at Charlotte, Mechanical Engineering and Engineering Science, Charlotte, NC, 28223, USA
[b]Universitat Politècnica de València, 46022 València, Spain

*Corresponding author. Tel.: +1-704-687-5086; fax: +1-704-687-8345. *E-mail address:* tony.schmitz@uncc.edu

Abstract

Machining processes, including turning, are a critical capability for discrete part production. One limitation to high material removal rates and reduced cost in these processes is chatter, or unstable spindle speed-chip width combinations that exhibit self-excited vibration. In this paper, an artificial neural network (ANN) is applied to model turning stability. The analytical stability limit is used to generate a data set that trains the ANN. It is observed that the number and distribution of training points influences the ability of the ANN model to capture the smaller, more closely spaced lobes that occur at lower spindle speeds. Overall, the ANN is successful (>90% accuracy) at predicting the stability behavior after appropriate training.

Keywords: turning, machine learning, neural network, stability, chatter

1. Introduction

Material removal processes, including turning and milling, are widely applied in industry. While significant advances have been achieved in the last decades, one limitation to high material removal rates is chatter, or unstable cutting conditions. The result is poor surface quality and potential damage to the tool, workpiece, and machine.

The control of stability in turning operations is crucial in industry and ongoing research efforts address chatter avoidance. Siddhpura and Paurobally [1] completed a literature review of chatter prediction in turning. They classified the techniques for chatter stability prediction as stability lobe diagrams, Nyquist plots, and finite element analyses. Their study also discussed the experimental techniques for chatter stability prediction and detection and separated them into three main groups: signal acquisition and processing techniques; chip analysis; and artificial intelligence techniques, including neural networks, hidden Markov models, and fuzzy logic. The authors noted that the number of publications featuring artificial intelligence techniques was low at the time of publication (only 11 from 1978 to 2012), although additional work has been done since then.

Chanda and Dwivedy [2] developed the governing nonlinear equations of motion for turning, considering both the workpiece and the tool to be flexible. The regenerative effect due to inherent time delay was considered. Copenhaver *et al.* [3] described a periodic sampling-based method for identifying the stability of modulated tool path turning (MTP). They compared a periodic sampling metric with the traditional frequency-domain approach, where the frequency spectrum is analyzed to identify the turning stability.

Filippov *et al.* [4] studied the transition from the stable turning to chatter using acoustic emission signals. High-frequency peaks appeared in transition to the chatter mode. A mathematical model of turning was presented by Gerasimenko *et al.* [5]. The stability limit for turning a thin-walled cylindrical part was defined. Gouskova *et al.* [6] determined the stability of a continuous cutting process for an arbitrary arrangement of two cutters. Gyebroszki *et al.* [7] combined the surface regeneration model of the turning process with the mathematical modeling of chip formation. It was shown that the time scale of chip formation was much smaller than the time scale of turning vibrations.

Hajdu *et al.* [8] incorporated noise and uncertainties in chatter predictions. They described a frequency-domain method to the robust stability analysis of machining operations, which uses the measured frequency response functions (FRFs) without filtering or modal parameter identification. Application of the method to a single-degree-of-freedom model for orthogonal cutting showed that the robust stability boundaries can be significantly smaller than the stability boundaries corresponding to the averaged FRF. Huang *et al.* [9] also considered uncertainties in their analysis. They used a probabilistic method (Monte Carlo simulation) and found that, in comparison with the traditional cutting force prediction method, Monte Carlo simulation provided better results, taking into account the influence of random parameters.

Mousavi *et al.* [10] presented a numerical model to predict the dynamic behavior of a robotic manipulator in a machining operation. They established theoretical stability limits taking into account the variability of the robot dynamics within its workspace. This enabled the cutting parameters and the robot configurations to be adapted along a machining trajectory. A stability diagram based on regenerative chatter in milling operations as a function of the kinematic redundancy variable was established. The theory was validated with experimental robotic machining trials. Liu *et al.* [11] investigated the probability of stability for turning. The authors defined and represented a reliability lobe diagram to identify stable and unstable zoned, rather than the traditional stability lobe diagram (SLD). The reliability was calculated using the FOSM (first-order second moment) and Monte Carlo methods and was compared to the traditional stability lobe diagram.

Khasawneh and Munch [12] proposed a new approach for determining the stability of stochastic dynamical systems by examining their time series using topological data analysis. Two statistical approaches (three sigma edit rule analysis and principal component analysis) were used by Jiménez *et al.* [13] to predict chatter, obtaining accuracy rates over 75%.

The receptance coupling method was used by Jasiewicz and Powalka [14] for determining the lathe-workpiece dynamics and inverse receptance coupling was proposed for the spindle. Lu *et al.* [15] proposed a predictive chatter model of a tailstock-supported flexible rod in straight turning including the effect of the traveling tool position along the longitudinal direction of the workpiece. The difference between the predicted tool locations and the experimental results was within 9%.

Tyler *et al.* [16] proposed an analytical stability model for turning which considered the process damping force that is dependent on the surface normal velocity, chip width, cutting speed, and a process damping coefficient. The analytical model was validated using time domain simulation and experiments.

Chatter prediction using machine learning techniques has also been studied, including neural networks, support vector machines, and others. Ahmad *et al.* [17] developed two different models of extreme learning techniques using random weights and hidden nodes. Lamraoui *et al.* [18] applied a neural network (NN) and the input data based on signal analysis to predict milling stability. Gupta *et al.* [19] used artificial intelligence techniques including support vector regression (SVR) and artificial neural networks (ANN) integrated with genetic algorithms (GA). The model was trained using the turning parameters as the input and corresponding surface roughness, tool wear, and power required as the output.

Jurkovic *et al.* [20] compared the performance of three machine learning methods for the prediction of operating parameters in high-speed turning. Observed parameters were the surface roughness (Ra), cutting force (F_c), and tool life (*T*). Polynomial (quadratic) regression, SVR, and ANN were used. Polynomial regression demonstrated the best performance in for F_c and Ra prediction, while the ANN showed the best performance for *T* prediction.

Khasawneh *et al.* [21] combined several deterministic and stochastic models to create persistence diagrams of turning stability. The approach was intended for chatter classification using signals produced by complicated and noisy manufacturing systems. Yao *et al.* [22] constructed a two-dimensional feature vector for chatter detection based on the standard deviation of the wavelet transform and the wavelet packet energy ratio in the emerging chatter frequency band. A support vector machine (SVM) was designed for pattern classification based on the feature vector. The machining states were classified into three categories: stable, transition, and chatter. Zagorski *et al.* [23] employed two NNs for chatter detection in milling: RBF (radial basis function) and MLP (multi-layered perceptron). Kumar and Singh [24] used an ANN based on feed forward back propagation for predicting the stable cutting zone and metal removal rate in turning. The tangent sigmoid activation function was applied.

2. Background

During turning, a sharp cutting edge is used to remove material in the form of a chip. Figure 1 shows an orthogonal cutting operation, where only the normal, F_n, and tangential, F_t, components of the resultant force, F, are considered. In general, the cutting force vector includes the third component along the workpiece rotation axis, but the orthogonal (planar) treatment is sufficient to describe the process dynamics. The figure also identifies: 1) the mean chip thickness, h_m, or commanded feed per revolution for the facing operation pictured; and 2) the force angle, β, between F and F_n. The side view of this operation (inset in Fig. 1) identifies the chip width, b. Together, the chip thickness and chip width define the area of material to be removed, $A = bh_m$.

The cutting force can be approximated as the product of the chip area and the process dependent specific force coefficient, K_s [25]. It depends on the workpiece material, tool geometry, and, to a lesser extent, the cutting speed (peripheral velocity of the rotating workpiece) and chip thickness.

$$F = K_s A = K_s b h_m \tag{1}$$

The normal and tangential components, F_n and F_t, can be expressed using F and the force angle:

$$F_n = \cos\beta\, F = \cos\beta\, K_s b h_m = k_n b h_m \text{ and} \quad (2)$$

$$F_t = \sin\beta\, F = \sin\beta\, K_s b h_m = k_t b h_m, \quad (3)$$

where the cutting force coefficients, k_n and k_t, are introduced which incorporate both K_s and β. A common approach used to characterize these process dependent values is to prescribe known cutting conditions and measure the force components directly.

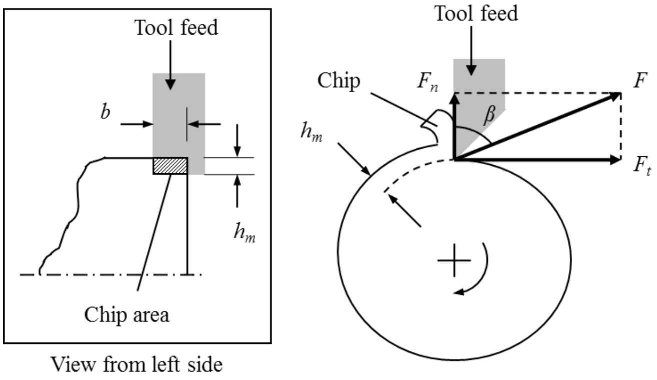

Fig. 1. Orthogonal cutting operation showing the cutting force with its normal and tangential components.

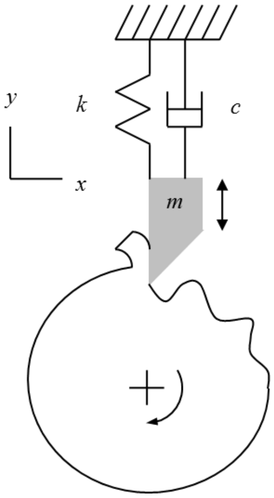

Fig. 2. Description of regenerative chatter in turning. Initial tool deflections are copied onto the workpiece surface and are encountered in subsequent revolutions. This varies the chip thickness and cutting force which, in turn, affects the resulting tool deflections.

The cutting force causes deflections of the cutting tool. Because the tool has stiffness and mass, it can vibrate. If the tool is vibrating as it removes material, these vibrations are imprinted on the workpiece surface as a wavy profile. Figure 2 shows an exaggerated view, where the initial impact with the workpiece surface causes the tool to begin vibrating and the oscillations in the normal direction to be copied onto the workpiece. When the workpiece begins its second revolution, the vibrating tool encounters the wavy surface produced during the first revolution. Therefore, the chip thickness at any instant depends both on the tool deflection at that time and the workpiece surface from the previous revolution(s). Vibration of the tool therefore leads to a variable chip thickness which, according to Eq. 1, yields a variable cutting force since the force is proportional to the chip thickness. The cutting force governs the current tool deflection and, subsequently, the system exhibits feedback.

From a modeling standpoint, this “regeneration of waviness” appears as a time-delayed term in the chip thickness equation. Figure 3 shows an unwrapped view of the turning operation, where the surface on the left was produced in the previous revolution and the surface to the right of the tool (offset by the mean feed per revolution) was just cut away by the oscillating tool. Only the vibrations in the normal direction, y (positive direction out of the cut), are considered here because they have the most direct influence on the chip thickness.

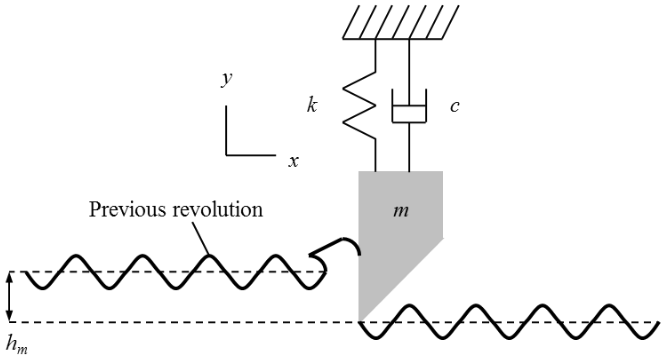

Fig. 3. Depiction of turning where the surface from the previous revolution, shown to the left of the tool, is removed by the vibrating cutter to produce a new wavy surface to the right of the tool.

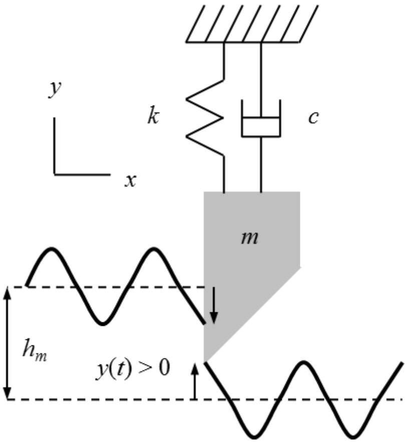

Fig. 4. The figure demonstrates the instantaneous chip thickness calculation. It depends on the mean feed per revolution, the current deflection, and the vibration during the previous revolution of the workpiece (to the left of the tool).

The time dependent, instantaneous chip thickness, $h(t)$, is determined using Eq. 4. It is seen that larger positive vibration during the previous revolution, $y(t-\tau)$, where τ is the time for one rotation, gives an increased chip thickness (i.e., less material was removed so the current chip is thicker). Larger positive current vibration, $y(t)$, on the other hand, yields a thinner chip; see Fig. 4.

$$h(t) = h_m + y(t-\tau) - y(t) \quad (4)$$

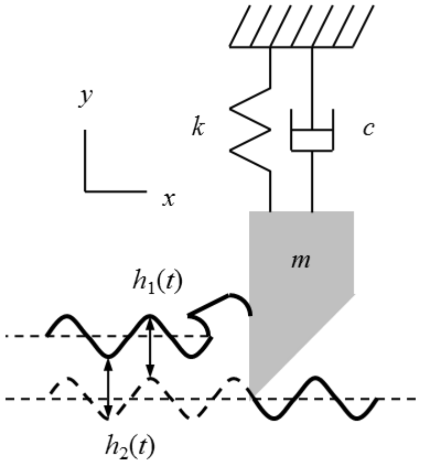

Fig. 5. The surface waviness between revolutions is in phase. Negligible chip thickness variation is obtained.

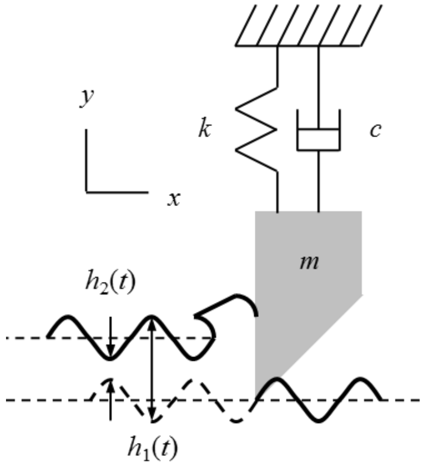

Fig. 6. Less favorable phase relationship between revolutions yields significant chip thickness variation.

The relative phasing between the surface waviness from one pass to the next determines the level of force variation and whether the operation is stable or unstable (chatter occurs). Figures 5 and 6 show two possibilities. In Fig. 5, the wavy surfaces between two revolutions are in phase. Therefore, even though vibration is present during material removal, the chip thickness variation (vertical distance between the two curves) is negligible and there is no appreciable force variation. This enables stable cutting at larger chip widths. Considering that the tool tends to vibrate at its natural frequency, it is intuitive that matching the workpiece rotating frequency (spindle speed) to the tool's natural frequency will lead to this preferred "in phase" situation. However, this is counter-intuitive based on a traditional understanding of resonance, where driving the system at its natural frequency is typically avoided. Figure 6 shows a less favorable phase relationship where there is significant variation in the chip thickness. This leads to unstable cutting at smaller chip widths than the previous case due to the force variations and subsequent tool deflections.

Depending on the feedback system "gain", or chip width *b*, and spindle speed, Ω, the turning operation will either be stable or exhibit chatter, which causes large vibrations and forces and leads to poor surface finish and, potentially, tool/workpiece damage. In stable machining, the vibrations diminish from revolution to revolution. In unstable machining, the vibrations grow from revolution to revolution until limited in some way. Surprisingly, the vibrations may become large enough that the tool jumps out of the cut, losing contact with the workpiece. The vibrations in unstable cutting may be at least as large as the chip thickness and it is not surprising that these large vibrations may result in damage to the machine, tool, and workpiece. The governing relationships for this behavior are provided in Eqs. 5 through 7 [26]. In these equations, b_{lim} is the limiting chip width to avoid chatter, f_c is the chatter frequency (should it occur), *FRF* is the frequency response function that describes the tool's dynamic response, N is the integer number of waves of vibration imprinted on the workpiece surface in one revolution, and $\frac{\varepsilon}{2\pi}$ is any additional fraction of a wave, where ε is the phase (in rad) between current and previous tool vibrations.

$$b_{lim} = \frac{-1}{2K_s \cos\beta Re[FRF]} \quad (5)$$

$$\frac{f_c}{\Omega} = N + \frac{\varepsilon}{2\pi} \quad (6)$$

$$\varepsilon = 2\pi - 2\tan^{-1}\left(\frac{Re[FRF]}{Im[FRF]}\right) \quad (7)$$

3. Artificial neural networks

The machine learning approach applied here for chatter prediction follows the supervised learning model, where the learning algorithm uses known input-output pairs for training. Once trained, the model can be used to predict outputs for new input data. When the output (typically discrete values) is used to create categories or classes, the problem is called a classification problem. When the output is a real, continuous value (or values), it is a regression problem. Since chatter prediction involves predicting whether a given set of input variables (spindle speed, Ω, and limiting chip width, b_{lim}) leads to chatter or not, a binary classification problem is to be solved. It is also supervised since the prediction is based on pairs of values (Ω, b_{lim}) for which the stability is known *a priori*. Furthermore, the model developed in this paper applies an ANN. An overview of ANNs is presented in the following paragraphs.

ANNs consist of a collection of basic units called neurons arranged in layers (Fig.7). The first (left) layer is the input layer and the last (right) layer is the output layer. The layers in between are hidden layers. A neural network can consist of no hidden layers or one or more hidden layers. In a feedforward neural network, the neurons in one layer are connected to the neurons in the next layer and the information flows forward from the input to the output through the hidden layers. When there are many hidden layers, the network is called a deep neural network (DNN). The connections between the neurons are called the synapses. A neuron, the basic building block of ANNs, consists of a set of input values

$x_i, i = 1 \ldots n$, a set of weights $w_i, i = 1 \ldots n$, and a transfer (or activation) function, f (see Fig. 8). A linear transformation consisting of the weighted sum of all the inputs $\sum w_i x_i$ and a bias, b, is calculated as:

$$z = b + \sum_{i=1}^{n} w_i x_i \tag{8}$$

for each neuron. The output, h, is calculated from this neuron through the (usually) nonlinear transfer function, $f(z)$. Typically, each neuron in a given layer has the same transfer function and for each neuron, i, in that layer, the output is calculated as $h_i = f(z_i)$, where z_i calculated using Eq. 8. The outputs serve as the inputs for each of the neurons in the next layer, which can use a different or the same transfer function. This process is continued until the output layer is reached where the neurons compute the output variables $y_i, i = 1 \ldots p$, (p is the number of outputs). For a binary classifier, there is usually only one neuron in the output layer and therefore p is taken to be unity. However, two neurons can also be used for binary classification.

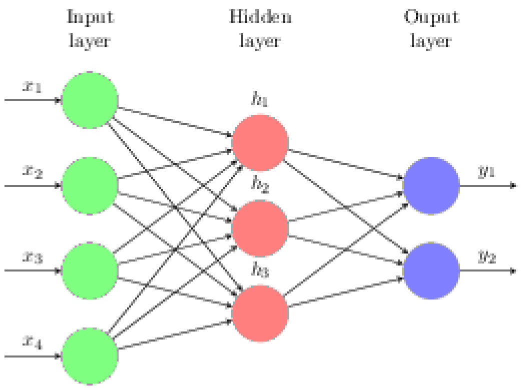

Fig. 7. An example of an artificial neural network (ANN). This ANN has four inputs (features), one hidden layer with three neurons, and two outputs.

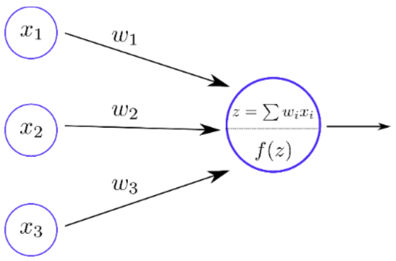

Fig. 8. A single neuron consists of the inputs x_i, the weights w_i, and the transfer function $f(z)$, which produces the scalar value h. Note that z is defined with the bias b absorbed into the summation by setting $x_0 = 1$ and $w_0 = b$.

In supervised learning, the training data (input data and the corresponding output data) is used to train the ANN model. The training starts with an initial assumption on the weights w_i. The input data is processed by the ANN and output is predicted. The error between the predicted outputs and the known outputs is calculated using a cost (or error) function which can be the sum of the squares of the errors between predicted and observed outputs, for example. Since the predicted values depend on the weights and biases, it is clear that the error function, E, is also a function of the weights and biases for a given set of training data, i.e., $E = E(w_i, b)$. By absorbing the bias b into the weights as an additional parameter, E can be assumed to be a function of only the weights w_i. If the error is not acceptable, the weights are updated through various methods. One approach is the gradient descent method, where the weight updates are computed using the derivatives of the error function with respect to the weights:

$$w_i = w_i - \eta \frac{\partial E}{\partial w_i}. \tag{9}$$

In Eq. 9, η is the learning rate that is used to control the magnitudes of the corrections applied to w_i. Too large a value of η will lead to convergence issues and too small a value increases the computational time and cost. The updated weights are again used for predictions and calculating the error in predictions. This process is repeated until the error is less than a preselected value or a maximum number of iterations has been reached. Although Eq. 9 captures the essence of weight updates, in a typical ANN with multiple hidden layers, the gradient calculation is quite complicated and involved. The back propagation algorithm may be used to compute these gradients. In the standard back propagation algorithm, the learning rate is kept constant. A modification of this algorithm, called the resilient back propagation algorithm, uses separate learning rates for each weight and, in addition, these rates can be altered during the training process to accelerate the convergence. Furthermore, the adjustments to weights do not include the partial derivatives of the error function with respect to weights. Instead, only the signs of the derivatives are used in place of the derivatives [27]. In the present work, the resilient backpropagation algorithm was used for updating the weights.

When the training is complete, the performance of the model is evaluated using test data with known outputs. Additional cross-validation methods are also used to further evaluate the model performance. If the predictions from the test data and cross-validations are satisfactory, the ANN model is used for predictive purposes on new sets of input data.

4. ANN model for chatter prediction

In this work the R neural net package, neuralnet [27], was used to build the ANN model. The input parameters for the neural network were the spindle speed and limiting chip width; see Fig. 9. The data set for training and testing was obtained from the stability algorithm described in section 2. The data set was generated by considering random values for the pairs (Ω, b_{lim}) in the range of 1000 rpm to 4000 rpm and 0 mm to 2.5 mm. For each set of values, the cut was labeled as stable or unstable (chatter) using the stability limit. A total of 201 points were generated this way. The distribution of these points is shown in Fig. 10. For training the ANN model, the data was first rescaled using the min-max method. The values, x, for each of the inputs was mapped to the range [0, 1] through the transformation $(x - x_{min})/(x_{max} - x_{min})$. Since there were no outliers in the input data, this rescaling method was acceptable. The normalized data was separated (randomly) into training and test data sets using an 80:20 ratio. This was determined to be the most reasonable split after attempting different proportions of training and test data. Thus, the training set consisted of 160 points, while the test set contained 41 points.

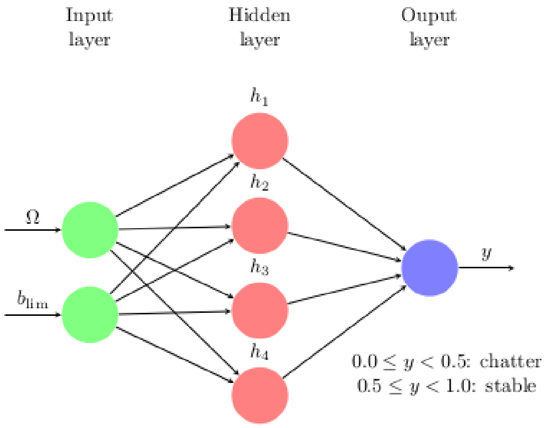

Fig. 9. The artificial neural network architecture proposed for the chatter stability problem.

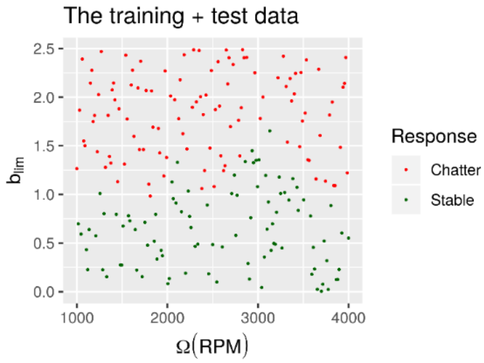

Fig. 10. The data used for training and testing the neural network model.

The ANN consisted of one hidden layer with four neurons (nodes); see Fig. 9. The input layer consisted of two input neurons while the output layer consisted of one output neuron. The output was the probability of stability, $p(x)$, defined by $p(x) = \{Y = 1|X\}$, where $X = \{\Omega, b_{lim}\}$ and $Y = 1$ corresponded to a stable cut. If p was less than 0.5, the cut was considered unstable and, when $p \geq 0.5$, the cut was considered stable. The activation function for both the hidden layer and output layer was the logistic function. The error function was the sum of squared errors, E, given by:

$$E = \sum_{i=1}^{n} (\bar{y}_i - y_i)^2 , \tag{10}$$

where $\bar{y}_i$ are the target outputs, y_i are the predictions, and n is the total number of inputs. The other option available in the neuralnet package for the error function is the cross-entropy function. Here, the synapse weights are updated using the default resilient back propagation algorithm. As mentioned previously, this algorithm uses a separate learning rate for each of the weights and the rates are automatically adjusted by the algorithm depending on the behavior of the error function. Since the weights are initially randomly assigned, the ANN model can be trained repetitively in order to minimize E. In the present work, the model was trained five times and the model parameters with the least error were selected. The training process was assumed complete when the absolute partial derivatives of the function with respect to the weights were less than 0.05 (threshold value). The ANN model with the least error is shown in Fig. 11. The figure also displays the synapse weights. The weights range from approximately -15 to 67.

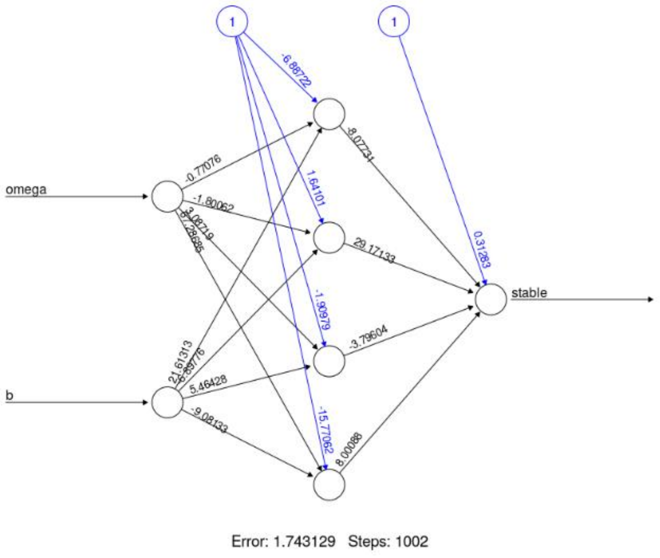

Fig. 11. ANN model based on the training dataset.

		Prediction		
		stable	chatter	total
Actual	stable	18	0	18
	chatter	3	20	23
	total	21	20	

Fig. 12. Confusion matrix for the ANN model predictions on the test data.

The performance of the ANN model was evaluated using the test data set. The confusion matrix for the predictions for the test data set is shown in Fig. 12. The ANN model predicts 38 (17 true positives and 21 true negatives) out of 41 observations correctly with an accuracy rate of 92.6%.

The ANN decision boundary and analytical stability limit obtained from the section 2 analysis are displayed in Fig. 13. The comparison demonstrates that the smaller lobes are not accurately modeled by the ANN decision boundary. However, the largest lobe is captured very well. This can be explained by noting that the complete data set used for the neural network model has more points distributed around the largest lobe and few points near the smallest lobes. In fact, as can be seen in Fig. 13, there are almost no points near the first two lobes. In spite of this, the ANN decision boundary captures the smaller lobes in an average sense. A refined distribution of the data around the lobes would provide improved results.

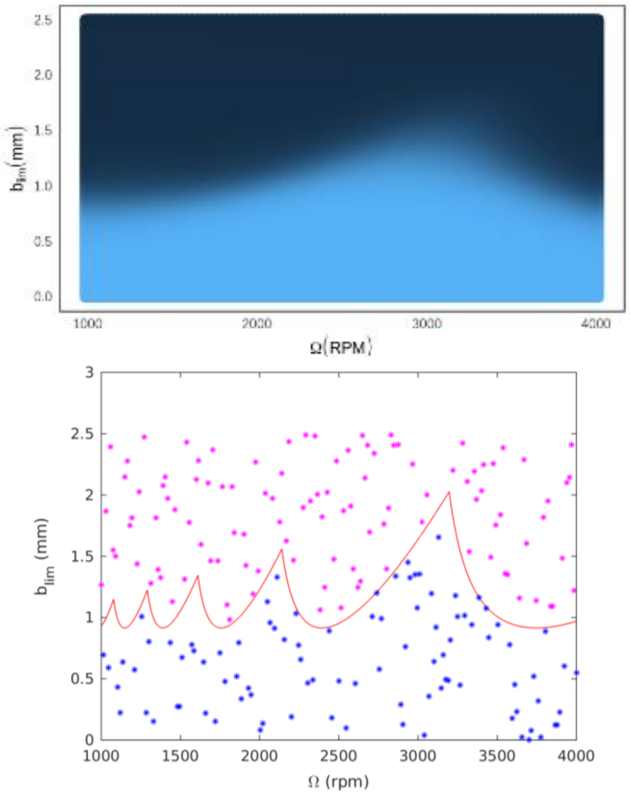

Fig. 13. (Top) ANN decision boundary. (Bottom) Stability limit from the section 2 analysis.

To further evaluate the performance of the ANN model, a 10-fold cross-validation analysis was performed on the complete data set. In this approach, the data was randomly divided into 10 subsets of approximately equal size. One of the 10 subsets was taken to be the test set, while the remaining nine sets were used for training. The accuracy of the trained model was predicted using the test set. This process was repeated 10 times by selecting a different subset as the test set and the remaining nine subsets as the training set each time. This ensured that each observation was part of at least one test set. As before, for each repetition, the training process was carried out five times with different initializations of the synapse weights and the model with the least error was chosen for predictions on the test set. The 10-fold validation test and the corresponding accuracy rates are shown in Fig. 14. It is observed that the accuracy rate is high for almost all the cases with the mean rate being 0.916 or 91.6%, which is comparable to the accuracy rate of the ANN model.

The generalized weights (GW) are plotted against the normalized values of b and Ω in Fig. 15. These quantities, defined by [28]:

$$\tilde{w}_i = \frac{\partial}{\partial x_i}\left[\frac{p(x)}{1-p(x)}\right], \tag{11}$$

characterize the sensitivity of the log odds (the term in the square brackets) to the input variables. The figures show these weights for all the input values in the training set. The variance of the weights in each case is quite high indicating that the two input variables have a strong, nonlinear effect on the ANN model. An interesting point to be made from these plots is that, overall, b has a negative effect on the model while Ω has a mostly positive effect.

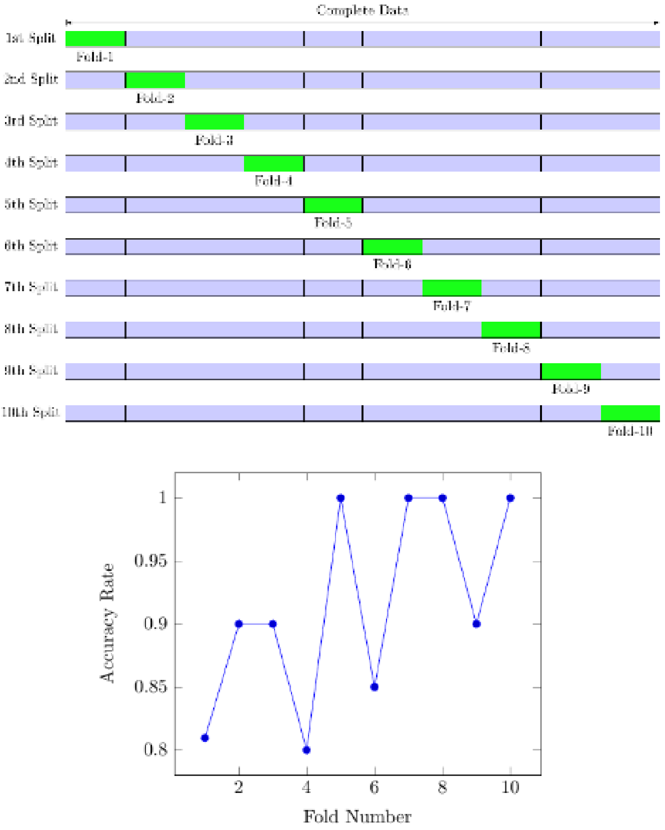

Fig. 14. (Top) Schematic of the 10-fold cross validation method. (Bottom) Accuracy rate for each fold number.

5. Conclusions

This paper provides an analysis of the application of the artificial neural network (ANN) to modeling stability behavior in turning. The analytical stability limit is used to generate a data set that trains the ANN. It is observed that the number and distribution of training points influences the ability of the ANN model to capture the smaller, more closely spaced lobes that occur at lower spindle speeds. Overall, the ANN is successful (>90% accuracy) at predicting the stability behavior after appropriate training.

Acknowledgements

The authors gratefully acknowledge financial support from the UNC ROI program. Elena Perez-Bernabeu and Miguel Selles also acknowledge support from Universitat Politènica de València (PAID-00-17). Additionally, some of the neural net figures and the 10-fold cross validation figures are based on the TikZ codes provided on StackExchange-TeX by various users. Harish Cherukuri would like to thank them for their valuable advice.

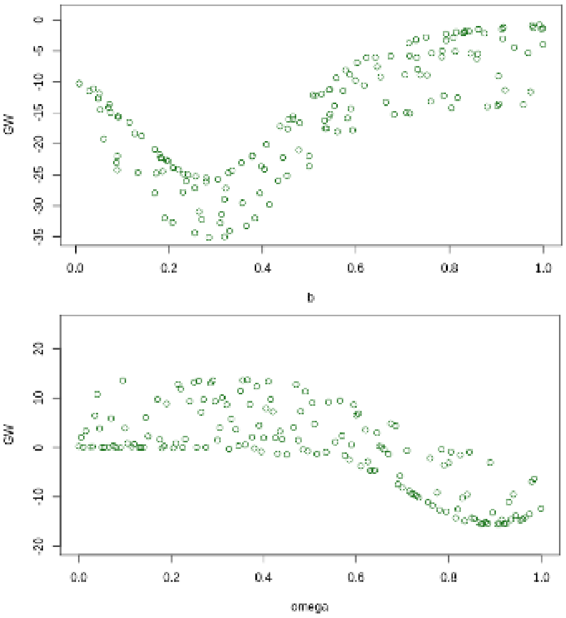

Fig. 15. Variation of the generalized weights with respect to the two input variables.

References

[1] Siddhpura M, Paurobally R. A review of chatter vibration research in turning. International Journal of Machine Tools & Manufacture 2012; 61:27-47.

[2] Chanda A, Dwivedy SK. Nonlinear dynamic analysis of flexible workpiece and tool in turning operation with delay and internal resonance. Journal of Sound and Vibration 2018; 434:358-378.

[3] Copenhaver R, Schmitz T, Smith S. Stability analysis of modulated tool path turning. CIRP Annals - Manufacturing Technology 2018; 67:49-52.

[4] Filippov AV, Rubtsov VE, Tarasov SY, Podgornykh OA, Shamarin NN, Detecting transition to chatter mode in peakless tool turning by monitoring vibration and acoustic emission signals. Int J Adv Manuf Technol 2018; 95:157-169.

[5] Gerasimenko A, Guskov M, Gouskov A, Lorong P, Shokhin A. Analytical modeling of a thin-walled cylindrical workpiece during the turning process. Stability analysis of a cutting process. Jve International Ltd. Journal of Vibroengineering 2017;19(8):5825-5841.

[6] Gouskova AM, Guskov MA, Tung DD, Panovko GY. Modeling and investigation of the stability of a multicutter turning process by a rrace. Journal of Machinery Manufacture and Reliability 2018;47(4):317-323.

[7] Gyebroszki G, Bachrathy D, Cserna G, Stepan G. Stability of turning processes for periodic chip formation, 2018. Adv. Manuf.; 6:345-353.

[8] Hajdu D, Insperger T, Stepan G. Robust stability analysis of machining operations 2017. Int J Adv Manuf Technol; 88:45-54.

[9] Huang X, Hu M, Zhang Y, Lv C. Probabilistic analysis of chatter stability in turning. Int J Adv Manuf Technol 2016; 87:3225-3232.

[10] Mousavi S, Gagnol V, Bouzgarrou BC, Ray P. Dynamic modeling and stability prediction in robotic machining. Int J Adv Manuf Technol 2017;88:3053-3065.

[11] Liu Y, Li T, Liu K, Zhang Y. Chatter reliability prediction of turning process system with uncertainties. Mechanical Systems and Signal Processing 2016;66-67:232-247.

[12] Khasawneh FA, Munch E. Chatter detection in turning using persistent homology. Mechanical Systems and Signal Processing 2016;70-71:527-541.

[13] Jiménez Cortadi A, Irigoyen I, Boto F, Sierra B, Suárez A, Galar D. A statistical data-based approach to instability detection and wear prediction in radial turning processes. Eksploatacja i Niezawodnosc – Maintenance and Reliability 2018;20(3):405-412.

[14] Jasiewicz M, Powalka B. Prediction of turning stability using receptance coupling 2018. AIP Conference Proceedings 2018;1922,100005.

[15] Lu K, Lian Z, Gu F, Liu H. Model-based chatter stability prediction and detection for the turning of a flexible workpiece. Mechanical Systems and Signal Processing 2018;100:814-826.

[16] Tyler C, Troutman JR, Schmitz TL. A coupled dynamics, multiple degree of freedom process damping model, Part 1: Turning. Precision Engineering 2016;46:65-72.

[17] Ahmad N, Janahiraman TV, Tarlochan F. Modeling of surface roughness in turning operation using extreme learning machine. Arab J Sci Eng 2015;40:595-602.

[18] Lamraoui M, Barakat M, Thomas M, El Badaoui M. Chatter detection in milling machines by neural network classification and feature selection. Journal of Vibration and Control 2015;21(7):1251-1266.

[19] Gupta AK, Guntuku SC, Desu RK, Balu A. Optimisation of turning parameters by integrating genetic algorithm with support vector regression and artificial neural networks. Int J Adv Manuf Technol 2015;77:331-339.

[20] Jurkovic Z, Cukor G, Brezocnik M, Brajkovic T. A comparison of machine learning methods for cutting parameters prediction in high speed turning process. J Intell Manuf 2016. DOI 10.1007/s10845-016-1206-1.

[21] Khasawneh FA, Munch E, Perea JA. Chatter classification in turning using machine learning and topological data analysis. IFAC PapersOnLine 2018;51(14):195-200.

[22] Yao Z, Mei D, Chen Z. On-line chatter detection and identification based on wavelet and support vector machine. Journal of Materials Processing Technology 2010;210:713-719.

[23] Zagórski I, Kulisz M, Semeniuk A, Malec A. Artificial neural network modelling of vibration in the milling of AZ91D alloy. Advances in Science and Technology Research Journal 2017;11(3):261-269.

[24] Kumar S, Singh B. Ascertaining of chatter stability using wavelet denoising and artificial neural network. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 2018.

[25] Tlusty G. Manufacturing Equipment and Processes, Prentice-Hall, Upper Saddle River, NJ, 2000.

[26] Schmitz T, Smith, KS. Machining Dynamics: Frequency Response to Improved Productivity, Second Edition, Springer, New York, NY, 2019.

[27] Fritsch S, Guenther F, Guenther MF. Package ‘neuralnet’. The Comprehensive R Archive Network. 2016 Aug 5.

[28] Intrator O, Intrator N. Interpreting neural-network results: A simulation study. Computational Statistics & Data Analysis. 2001 Sep 28;37(3):373-93.