Springer

KSME

Journal of Mechanical Science and Technology 24 (5) (2010) 1041~1052
www.springerlink.com/content/1738-494x
DOI 10.1007/s12206-010-0308-5

Flank wear regulation using artificial neural networks †

Danko Brezak*, Dubravko Majetic, Toma Udiljak and Josip Kasac

Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Zagreb, Croatia

(Manuscript Received March 25, 2009; Revised December 3, 2009; Accepted January 29, 2010)

Abstract

Tool wear regulation highly influences product quality and the safety and productivity of machining processes. Hence, it is one of the most important elements in the supervisory control of machine tools. The development of this type of machine tool adaptive control is practically at its infancy because there are still no industrial solutions concerning robust, reliable, and highly precise continuous tool wear estimators. Therefore, this paper primarily aims at the determination of a tool wear regulation model that can ensure the maximum allowed amount of tool wear rate within a predefined machining time, while simultaneously maintaining a high level of process productivity. The proposed model is structured using Radial Basis Function Neural Network controller and Modified Dynamical Neural Network filter. It is analysed using an analytical tool wear model with experimentally adjusted parameters.

Keywords: Control; Machining; Neural network; Productivity maximisation; Tool wear regulation

1. Introduction

In order to achieve higher levels of machining process automation, different types of control strategies and techniques have to be developed and integrated into the machine tool control system. Major research efforts have been focused on the development of different force control models because of their high influence on productivity. In addition, problems concerning chatter suppression, elimination of burr formation, and achievement of chip discontinuous forms, among others, have also been considered. However, a less work has been done in the field of tool wear regulation (TWR). In their state-of-the-art paper on machining process monitoring and control, Liang et al. [1] acknowledged this problem. However, they did not offer possible solutions thereto. The development of tool wear control models is practically at its infancy because there are still no industrial solutions for robust, reliable, and highly precise on-line tool wear estimators, despite extensive research efforts in the past years. Koren [2] emphasised that the absence of industrially acceptable on-line direct tool wear measurement systems and the disadvantages of indirect measurement techniques represent the primary obstacles in developing commercially adaptive control systems with optimisation for milling, turning, and drilling.

According to Liang et al. [1] and Landers et al. [3], the purpose of tool wear regulation system is to adjust process parameters so that the tool life can either be maximised in a job shop production environment or be attuned to a scheduled tool change period in a mass production environment. Another issue is related to tool breakage avoidance in situations of highly intensive unexpected wear when there is a need and a possibility to finish a machining operation or to provide a safe tool exit.

In this paper, a tool wear regulation model, based on two types of static and dynamic neural networks, is presented. At times, the estimation of tool wear parameters can be highly imprecise; hence, estimated values have to be filtrated in order to reduce the negative influence of estimation errors on the overall control process. A filter, structured in the form of a recurrent neural network, is therefore proposed. For the activation function, the Radial Basis Function is chosen. Moreover, neural network parameters are adjusted using a variant of the Resilient Back–Propagation learning algorithm (RPROP). The control algorithm is realised using the Radial Basis Function Neural Network (RBFNN) because of its good approximation capabilities and the possibility of a relatively simple and fast structural configuration.

With experimentally defined parameters, adaptation of the controller and filter structure, as well as analyses of their characteristics in this type of machining process control, is conducted using the Koren-Lenz analytical flank wear model. This model is chosen because flank wear width (VB) is the most dominant tool wear parameter. Additionally, machining process disturbances and tool wear estimator errors are also

† This paper was recommended for publication in revised form by Associate Editor In-Ha Sung

*Corresponding author. Tel.: +385 1 61 68 357, Fax.: +385 1 61 56 940
E-mail address: danko.brezak@fsb.hr

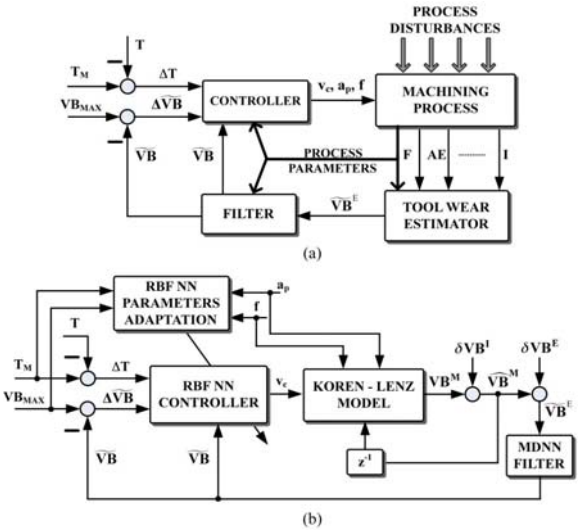

Fig. 1. Tool wear regulation model in real (a) and simulated (b) form.

added to the simulated TWR model (Fig. 1).

2. TWR objectives

Following the aforementioned reasons for TWR implementation, the primary control objective of the proposed TWR model is focused on constraining the amount of tool wear parameter, – the flank wear width (*VB*), – to some defined maximal value (VB_{MAX}) within a predefined machining time (T_M) which represents the tool life. This condition is additionally expanded with the high process productivity criterion quantified by the material removal rate (*MRR*) – the volume of the workpiece material removed in the time unit.

$$VB = VB_{MAX} \text{ for } T = T_M \text{ and } MRR \rightarrow max. \quad (1)$$

Therefore, enhancing the cutting tool efficiency inside tool changing cycles is possible. Although productivity maximisation criterion has no real significance in the case of sudden critical wear rate, the fulfilment of the first constraint set on the tool wear parameter and the machining time could prevent tool breakage.

The material removal rate is defined as

$$MRR = 1000\, v_c\, f\, z\, a_p, \quad (2)$$

where v_c is the cutting speed (m/min), f is the feed (mm/tooth), z is the number of teeth, and a is the depth of the cut (mm); hence, *MRR* can obviously be changed by all three cutting parameters. However, from the technological perspective, feed and cutting speed are the most appropriate in–process adaptation parameters. The relationships between *MRR* and feed and between *MRR* and cutting speed are proportional and equivalent. On the other hand, cutting speed has a higher influence on tool wear intensity than feed [4, 5]. Considering all these facts and in order to fulfil the criteria defined by Eq. (1), the

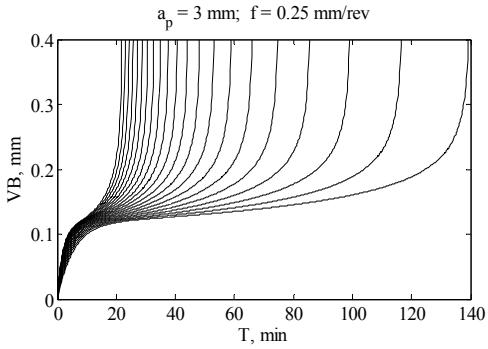

Fig. 2. Flank wear curves (for v_c = 100; 110; 120 … 300 m/min).

cutting speed has to be maximised and used as the only adaptation parameter, while the feed has to be kept on its maximal possible value, which is constrained by the surface quality parameter and/or other technological conditions. Thus, cutting speed is used as a control variable, and the remaining cutting parameters are taken as the additional input parameters of the control algorithm.

In ideal situations, with no disturbances, the maximal productivity of the machining process can be accomplished by the cutting speed, which can provide direct transition from some initial or actual tool wear state to a new and desired one, which is within the defined machining time. The proof of this hypothesis is given in the Appendix. It is based on findings concerning the optimal cutting speed for MRR maximisation in tool wear regulation, using Taylor's tool life equation. In real conditions, the optimal cutting speed is impracticable because of permanent, unavoidable, and often very influential process disturbances that greatly influence the estimation errors of tool wear parameters. Additionally, the discrepancy between the maximum and achieved *MRR* also depends on the characteristics of chosen filtration and control algorithms because their parameters are adapted only on the basis of average tool wear dynamics, which has to be determined experimentally for every combination of cutting conditions. Nevertheless, despite these limitations caused by complex tool wear dynamics, the aforementioned approach for cutting speed adaptation is generally applicable in ensuring high productivity in tool wear regulation processes.

3. Koren-lenz tool wear model

In real situations, the average values of flank wear parameters are supposedly defined on the basis of measurements conducted for the purpose of tool wear estimator structuring. In this work, an experimentally defined analytical tool wear model is used. However, it only serves as a generator of the required average values of the flank wear parameters for different combinations of cutting parameters (Fig. 2) and in analysing the proposed TWR model.

3.1 Model structure

The utilised tool wear model was proposed by Koren and

Lenz [6], which is based on two dominant wear mechanisms, namely, abrasion and diffusion. Later, it has been combined with the crater wear model and is described in detail by Danai and Ulsoy [7]. Unlike the extended form, in this research, we concentrated on the flank wear control problem; thus, only a flank wear model is used, as in the work of Ulsoy et al. [8]. The total value of the flank wear parameter is defined as the sum of its abrasive (VB_A) and diffusion (VB_D) components.

$$VB = VB_A + VB_D . \tag{3}$$

The relationships representing these two components are:

$$\frac{l_0}{v_c}\dot{VB}_A + VB_A = K_1 \cos\gamma_r \, F \frac{1}{fa_p}, \text{ and} \tag{4}$$

$$\dot{VB}_D = K_2 \sqrt{v_c}\, e^{\frac{-K_3}{(273+\theta_f)}}, \tag{5}$$

where l_0, is a constant, v_c is the cutting speed, γ_r is the effective rake angle, and f and a_p are the feed and depth of the cut, respectively. Cutting force is defined as

$$F = [K_9 f^{n_7}(1 - K_{10}\gamma_r) - K_{11} - K_{12} v_c] a_p + K_{13} a_p VB , \tag{6}$$

and the tool-work temperature on the flank side of the tool is defined as

$$\theta_f = K_6 v_c^{n_1} f^{n_2} + K_7 VB^{n_3} . \tag{7}$$

The other parameters are constants, which are determined in experimental measurements [9]. In the model, the following intervals of cutting parameters are defined: a_p = 2 - 4 mm, v_c = 100 - 300 m/min, and f= 0.15 - 0.35 mm/rev.

3.2 Simulation of disturbances

As expected, there will always be some discrepancies between the average and actual tool wear intensity in the real industrial environment; hence, the so-called impulse and estimation error disturbances are additionally added to the simulated tool wear model outputs.

Impulse disturbances are related to the appearance of the sudden augmentation of wear intensity (i.e., unexpected instant damages of cutting edge). They are still not classified as tool breakage. They are chosen as the most intensive forms of tool wear processes. Their implementation is realised through the increment (δVB^I), which is defined as a product of the Koren-Lenz output value VB^M and δ^I factor in every ith simulation step.

$$\delta VB_i^I = VB_i^M \delta_i^I , \quad \delta_i^I = \begin{Bmatrix} 0 & , & i < i' \\ > 0 & , & i = i' \\ 0 & , & i > i' \end{Bmatrix} . \tag{8}$$

Estimation error disturbances are a result of unavoidable and often very influential inaccuracies in the estimation of the flank wear parameter. This type of noise (δVB^E) is implemented using white noise in the intervals of ± 0.05 mm and ± 0.1 mm

$$\delta VB_i^E = e_i , \quad e_i \in [e_{\min}, e_{\max}] . \tag{9}$$

Process disturbances are implemented in the tool wear model as

$$VB_i^M = f\left\{ (v_c, a_p, f_z)\big|_i , \widehat{VB}_{i-1}^{\,M} \right\}, \tag{10}$$

$$\widehat{VB}_i^M = VB_i^M + \delta VB_i^I . \tag{11}$$

The final simulated value of the 'estimated' flank wear parameter is obtained from

$$\widetilde{VB}_i^E = \widehat{VB}_i^M + \delta VB_i^E . \tag{12}$$

4. Modified dynamic neural network

Reducing the highly negative influences of estimation error on the quality of control processes is a necessary precondition in any successful tool wear regulation. With its data processing and filtering capabilities, the Modified Dynamic Neural Network (MDNN) is proposed as an estimated flank wear parameter filter. It is a variant of the dynamic neural network, named Dynamic Multilayer Perceptron Network (DMLP), which was presented by Ayoubi et al. [10]. This type of network is characterised by a dynamic neuron model, the so-called Dynamic Elementary Processor (DEP), which is structured as an Auto Regressive Moving Average (ARMA) filter and is built into the network hidden layer. This way, every hidden layer neuron process previous values of its own activity together with its new input signals. In contrast to the DMLP network, Gauss activation functions are used in the MDNN network, and the structure is simplified by omitting the output layer activation function.

4.1 MDNN algorithm

The input value of the kth hidden layer neuron in the ith step is calculated from the sum of the products of all l network input vector elements (x) and their weight factors (v).

$$net_{k,i} = \sum_{l=1}^{L} v_{kl} x_{l,i} , \quad k = 1,..., K , \; i = 1,..., I \tag{13}$$

The obtained sum is then processed in DEP unit (i.e., ARMA filter), which can be written in the form of an impulse transfer function:

$$G_k(z) = \frac{B_k(z)}{A_k(z)} = \frac{\overline{y}_k(z)}{net_k(z)} = \frac{b_{0k} + b_{1k} z^{-1} + b_{2k} z^{-2}}{1 + a_{1k} z^{-1} + a_{2k} z^{-2}} , \tag{14}$$

that is, in the form of a difference equation

$$\bar{y}_{k,i} = b_{0k} net_{k,i} + b_{1k} net_{k,i-1} + b_{2k} net_{k,i-2} - \\ - a_{1k}\,\bar{y}_{k,i-1} - a_{2k}\,\bar{y}_{k,i-2} \quad , \tag{15}$$

where a and b are the filter coefficients, and $\bar{y}_{k,i}$ is the filter output.

The output of the kth hidden layer neuron (DEP unit) is

$$y_{k,i} = e^{-\frac{1}{2}\left(\frac{\bar{y}_{k,i}-t_k}{\sigma_k}\right)^2} \quad , \tag{16}$$

where t_k is the centre and σ_k is the width of the activation function. In the end, by summing the outputs of all hidden layer neurons and belonging weight factors (w), the final mth network output value is defined as

$$O_{m,i} = \sum_{k=1}^{K} y_{k,i}\, w_{mk} \quad , \quad m = 1,...,M \quad . \tag{17}$$

4.2 RPROP learning algorithm

Network parameters (v, w, b_0, b_1, b_2, a_1, a_2, t, σ) are adapted using the fast variant of the Resilient Back-Propagation learning method, the so-called RPROP with New Weight-Backtracking Scheme [11-13], which demonstrates fast error convergence and good generalization characteristics. Modification of the network learning parameter (lp) for every new learning step depends on parameter Δ, which has to be adapted in every nth learning step according to the following expression:

$$\Delta_n = \begin{cases} \min\left(\eta^+ \cdot \Delta_{n-1}, \Delta_{\max}\right), & \nabla E_{n-1}(lp)\cdot\nabla E_n(lp) > 0 \\ \max\left(\eta^- \cdot \Delta_{n-1}, \Delta_{\min}\right), & \nabla E_{n-1}(lp)\cdot\nabla E_n(lp) < 0 \\ \Delta_{n-1} \quad , & \nabla E_{n-1}(lp)\cdot\nabla E_n(lp) = 0 \end{cases}, \tag{18}$$

where $0<\eta^-<1<\eta^+$. Increasing (η^+) and decreasing (η^-) parameters are generally empirically defined. In this study, η^+=1.2 and η^-= 0.5 have been chosen according to the recommendations and conclusions of Riedmiller [12]. The new value of the learning parameter is

$$lp_{n+1} = lp_n + \Delta lp_n \quad , \tag{19}$$

where

$$\Delta lp_n = -sign[\nabla E_n(lp)]\cdot\Delta_n \quad , \\ for\ \ \nabla E_{n-1}(lp)\cdot\nabla E_n(lp) \geq 0 \tag{20}$$

and

$$\Delta lp_n = -\Delta lp_{n-1} \quad , \\ for\ \ \nabla E_{n-1}(lp)\cdot\nabla E_n(lp) < 0\ and\ E_n > E_{n-1} \quad . \tag{21}$$

Parameter E is the sum of the squared errors of all MDNN network outputs (M)

$$E = \frac{1}{2}\sum_{m=1}^{M}\sum_{i=1}^{I}\left(d_{m,i} - O_{m,i}\right)^2 \quad , \tag{22}$$

where $d_{m,i}$ is the desired value of the mth output for the ith learning sample. If the sign of the partial derivative in two sequential steps is changed ($\nabla E_{n-1}(lp)\cdot\nabla E_n(lp) < 0$), the gradient of error function with respect to the learning parameter (lp) is set to zero ($\nabla E_n(lp) = 0$), thus reducing the amount of lp in the next step. If, at the same time, the second condition from Eq. (21) is fulfilled (i.e., $E_n > E_{n-1}$), then $lp_{n+1} = lp_{n-1}$. On the other hand, when $E_n \leq E_{n-1}$, the amount of learning parameter in the actual step does not change ($lp_{n+1} = lp_n$).

At the beginning of the learning process, the initial Δ values of all learning parameters are set to some arbitrary chosen small value, which is proportional to the initial values of the learning parameters (in this work $\Delta = 10^{-2}$). The upper limit is $\Delta_{max} = 50$, and the lower limit is $\Delta_{min} = 10^{-6}$ as suggested by Riedmiller and Braun [11]. MDNN network parameters are adapted using a batch learning procedure. The partial derivatives of error are calculated for every sample, and their sum in the nth learning step is used for network parameters adaptation. The initial values of partial derivatives are set to zero at the beginning of every new learning step. The partial derivative of error with respect to the weights of the network output layer is obtained from

$$\frac{\partial E_{m,i}}{\partial w_{mk}} = \left(\frac{\partial E_m}{\partial O_m}\frac{\partial O_m}{\partial w_{mk}}\right)\Bigg|_i = -\left(d_{m,i} - O_{m,i}\right) y_{k,i} \, . \tag{23}$$

Their sum in the nth step for weight w_{mk}, which connects the kth hidden layer neuron and the mth output layer neuron, is

$$\nabla E_n\left(w_{mk}\right) = \sum_{i=1}^{I}\frac{\partial E_{m,i}}{\partial w_{mk}} \, . \tag{24}$$

The partial derivatives of learning errors with respect to DEP unit parameters for the kth hidden layer neuron and their sums are obtained from

$$\frac{\partial E_i}{\partial b_{jk}} = \left(\frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial \bar{y}_k}\frac{\partial \bar{y}_k}{\partial b_{jk}}\right)\Bigg|_i \\ = \left(-\sum_{m=1}^{M}\left(d_{m,i} - O_{m,i}\right) w_{mk}\right) y_{k,i}\left(\frac{\bar{y}_{k,i} - t_k}{{\sigma_k}^2}\right) D_{b_{jk},i} \, , \tag{25}$$

$$\left[D_{b_{jk},i}\right] = \frac{z^{-j}}{A(z)}\left[net_{k,i}\right], \ j = 0,1,2 \, , \tag{26}$$

$$\nabla E_n\left(b_{jk}\right) = \sum_{i=1}^{I}\frac{\partial E_i}{\partial b_{jk}} \, ; \tag{27}$$

$$\frac{\partial E_i}{\partial a_{jk}} = \left(\frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial \overline{y}_k} \frac{\partial \overline{y}_k}{\partial a_{jk}} \right)\Bigg|_i = \left(-\sum_{m=1}^{M} \left(d_{m,i} - O_{m,i} \right) w_{mk} \right) y_{k,i} \left(\frac{\overline{y}_{k,i} - t_k}{\sigma_k^{\,2}} \right) D_{a_{jk},i} , \tag{28}$$

$$\left[D_{a_{jk},i} \right] = \frac{-z^{-j}}{A(z)} \left[\overline{y}_{k,i} \right], \; j = 1,2 , \tag{29}$$

$$\nabla E_n \left(a_{jk} \right) = \sum_{i=1}^{I} \frac{\partial E_i}{\partial a_{jk}} . \tag{30}$$

In the parameters of the Gauss activation function, the sum of their learning error gradients is defined from the following expressions:

$$\frac{\partial E_i}{\partial t_k} = \left(\frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial t_k} \right)\Bigg|_i = \left(-\sum_{m=1}^{M} \left(d_{m,i} - O_{m,i} \right) w_{mk} \right) y_{k,i} \left(\frac{\overline{y}_{k,i} - t_k}{\sigma_k^{\,2}} \right), \tag{31}$$

$$\nabla E_n \left(t_k \right) = \sum_{i=1}^{I} \frac{\partial E_i}{\partial t_k} ; \tag{32}$$

$$\frac{\partial E_i}{\partial \sigma_k} = \left(\frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial \sigma_k} \right)\Bigg|_i = \left(-\sum_{m=1}^{M} \left(d_{m,i} - O_{m,i} \right) w_{mk} \right) y_{k,i} \left(\frac{\left(\overline{y}_{k,i} - t_k \right)^2}{\sigma_k^{\,3}} \right), \tag{33}$$

$$\nabla E_n \left(\sigma_k \right) = \sum_{i=1}^{I} \frac{\partial E_i}{\partial \sigma_k} . \tag{34}$$

In the end, the sum of partial derivatives with respect to network input layer weight factors is quantified as follows:

$$\frac{\partial E_i}{\partial v_{kl}} = \left(\frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial \overline{y}_k} \frac{\partial \overline{y}_k}{\partial v_{kl}} \right)\Bigg|_i = \left(-\sum_{m=1}^{M} \left(d_{m,i} - O_{m,i} \right) w_{mk} \right) y_{k,i} \left(\frac{\overline{y}_{k,i} - t_k}{\sigma_k^{\,2}} \right) D_{v_{kl},i} , \tag{35}$$

$$\left[D_{v_{kl},i} \right] = \frac{B(z)}{A(z)} \left[x_{l,i} \right], \tag{36}$$

$$\nabla E_n \left(v_{kl} \right) = \sum_{i=1}^{I} \frac{\partial E_i}{\partial v_{kl}} . \tag{37}$$

The learning procedure is conducted using noisy data ($\widetilde{VB}^E$) as network inputs and simulated 'average' flank wear values as outputs (VB^M). Every set of training data is related to the belonging combination of cutting parameters. Sixteen network structures with arbitrarily chosen input neurons (3, 5, 7, and 9) and hidden layer neurons (2, 3, 5 and 10) are analysed using tests with different types of disturbances and combinations of cutting parameters. All structures are configured in 10, 000 learning steps, although learning errors practically reach their minimum after 3, 000 steps. The best performance is achieved with the 5-5-1 network structure, that is, the filtered tool wear parameter in the *i*th step ($\widetilde{VB}_i$) is determined on the basis of estimated values from the previous four steps ($\widetilde{VB}^E_{i-4}$, $\widetilde{VB}^E_{i-3}$, $\widetilde{VB}^E_{i-2}$, $\widetilde{VB}^E_{i-1}$) and the *i*th step ($\widetilde{VB}^E_i$). The absence of filtration in the first four steps does not have any relevant influence on the overall control process. The chosen MDNN filter structure has a filtration error of $\Delta VB < 0.02$ mm in 88-97% of the samples, depending on the test types. Higher filtration errors are manifested in tests with higher estimation errors and impulse disturbances (Fig. 3).

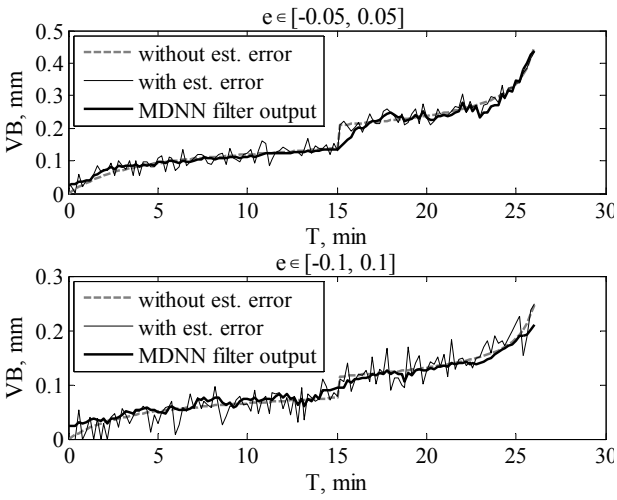

Fig. 3. MDNN filter outputs achieved with a 5-5-1 network structure.

5. RBFNN controller

In order to fulfil defined control objectives, a controller with good approximation characteristics of multidimensional input space to the scalar output is required. For this purpose, a well-known RBF neural network is chosen because of its universal functional approximation and good generalization capabilities.

5.1 Control algorithm

The control variable is obtained from the product of matrix ***H***, which is defined by the outputs of hidden layer neurons and the weight vector ***c***,

$$v_c = \mathbf{Hc} . \tag{38}$$

The elements of matrix ***H*** are determined from

$$H_{ij} = e^{\frac{-\left\| \mathbf{x}_i - \mathbf{t}_j \right\|^2}{\sigma_j^2}}, \; i = 1, \ldots, N, \; j = 1, \ldots, K \tag{39}$$

where $\boldsymbol{x}_i$ is the input vector composed from the *i*th element of all input neurons, $\boldsymbol{t}_j$ is the vector of the *j*th hidden layer neuron position centre and σ is the width of the Gauss (radial basis)

activation function

$$\sigma_j = \sqrt{d_{1j} d_{2j}}\,, \tag{40}$$

where d_{1j} is the Euclidean distance between the jth and (j-*1*)th neuron centre, and d_{2j} is the Euclidean distance between the jth and (j+*1*)th neuron centre.

According to Fig. 1, the control variable is a function of three variables:

$$v_{c,i} = \left[\Delta T_i,\, \Delta\widetilde{VB}_i,\, \widetilde{VB}_i\right], \tag{41}$$

where ΔT_i is the difference between machining time and time elapsed from the start of machining process (T). $\Delta\widetilde{VB}_i$ is the difference between the maximal referent flank wear parameter value and the actual, filtered one ($\widetilde{VB}_i$). This way, the RBF controller approximates the cutting speed, which can accomplish referent tool wear state from the actual one (within the remaining machining time), based on the average tool wear dynamics used for controller structuring.

5.2 Learning algorithm

The learning phase elements of weight vector $\boldsymbol{c}$ and matrix $\boldsymbol{H}$ need to be found. Weight parameters are determined by the recursive least squares algorithm [14]. It is an iterative method where the total number of iterations is equal to the number of learning samples (N). The weight vector belonging to the mth output neuron in the ith step is

$$\mathbf{c}_{m,i} = \mathbf{c}_{m,i-1} + \mathbf{G}_{i-1}\mathbf{d}_i\left[\mathbf{d}_i^T\mathbf{G}_{i-1}\mathbf{d}_i + 1\right]^{-1}\left[\mathbf{O}_{m,i} - \mathbf{d}_i^T\mathbf{c}_{m,i-1}\right], \tag{42}$$

where d is the column vector of the $\boldsymbol{D}$ matrix ($\boldsymbol{D} = \boldsymbol{H}^T$) and the $\boldsymbol{G}$ matrix is defined as

$$\mathbf{G}_i = \mathbf{G}_{i-1} - \mathbf{G}_{i-1}\mathbf{d}_i\left[\mathbf{d}_i^T\mathbf{G}_{i-1}\mathbf{d}_i + 1\right]^{-1}\left[\mathbf{d}_i^T\mathbf{G}_{i-1}\right]. \tag{43}$$

Initial conditions are set: $\boldsymbol{c}_{m,0} = [0]$ and $\boldsymbol{G}_0 = \alpha\boldsymbol{I}$, where α is an arbitrarily chosen large positive constant ($\alpha = 10^6$) and $\boldsymbol{I}$ is the identity KxK matrix.

In order to define matrix $\boldsymbol{H}$ elements, the number and position of every hidden layer neuron centre ($\boldsymbol{t}$) have to be established first. The procedure is carried out separately for every combination of cutting parameters using the belonging set of flank wear parameters, which represents the average tool wear dynamics. It is based on the algorithm for flank wear parameter samples that combine into groups. The group centres are then chosen as hidden layer neuron centres. For this purpose, the experimentally defined β_A parameter is used as the maximum allowed distance between the group centre and the element.

$$\begin{aligned} &\max_g\left\{\left\|\mathbf{u}_i - \bar{\mathbf{u}}_g\right\|,\ i = 1,\ldots,\ N,\ \ g = 1,\ldots,\ G\right\} \le \beta_A, \\ &\mathbf{u}_i = \left[VB_i^M, T_i\right],\ \bar{\mathbf{u}}_g = \left[\overline{VB}_g^M, \bar{T}_g\right], \end{aligned} \tag{44}$$

where $\boldsymbol{u}_i$ is the ith samples vector, and $\bar{\mathbf{u}}_g$ is the mean value vector of the gth analysed group. Both parameters are normalized within the interval [0, 1]. All samples fulfilling this condition participate in the calculation of the group centre, and the algorithm continues until all samples are grouped into belonging groups.

A higher β_A parameter value influences the reduction of the number of hidden layer neurons. It reduces the oscillations of the control variable in the case of higher input vector variations. However, this reduction may have negative impacts on the quality of cutting speed approximation. Additionally, it is likely that the controller cannot accomplish its task in situations of highly intensive tool wearing. On the other hand, with smaller values of the β_A parameter, these problems can be avoided. However, too small values lead to the higher augmentation of hidden layer neurons, which can cause overfitting (i.e., poor generalisation characteristics). Therefore, a set of cases is analysed with different tool referent states and cutting parameters (Fig. 4). The results are evaluated and compared using the productivity factor

$$\mathrm{P} = \sum_{\mathrm{i}=1}^{I} \mathrm{v_{c,i}}\left(T_{\mathrm{i}} - T_{\mathrm{i-1}}\right). \tag{45}$$

At the end, β_A = 0.05 (317 hidden layer neurons) is finally chosen. In this case, it is a well-balanced value between high productivity and satisfactory approximation capabilities.

Controller parameters are adapted at the beginning of the control cycle for every new combination of machining parameters and tool referent state. Adaptation is also required in cases when cutting parameters and/or the maximal referent flank wear parameter value (VB_{MAX}) is modified during the cutting process. If new and modified values of depth of cut and feed do not participate in the learning phase, the controller output has to be determined using linear approximation of the outputs obtained by different controller structures, which are defined for the closest values of these two cutting parameters to the actual one.

6. Simulation results

In the learning phase, 25 sets of filter and controller parameters related to all combinations of chosen depth of cut (2, 2.5, 3, 3.5, 4 mm) and feed (0.15, 0.2, 0.25, 0.3, 0.35 mm/rev) are established. The learning procedures for every set of parameters are conducted on the base of 16 cutting speed values (100, 110, 120 ⋯ 200, 220, 240 ⋯ 300 m/min). These network structures are then analysed using different sets of cutting parameters, final tool wear conditions, and estimation errors

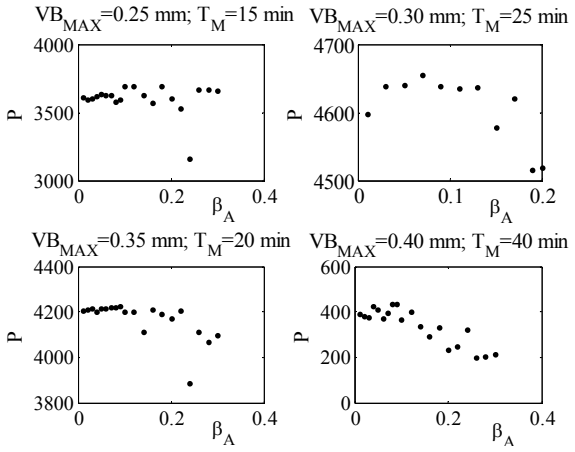

Fig. 4. P-β_A interrelation.

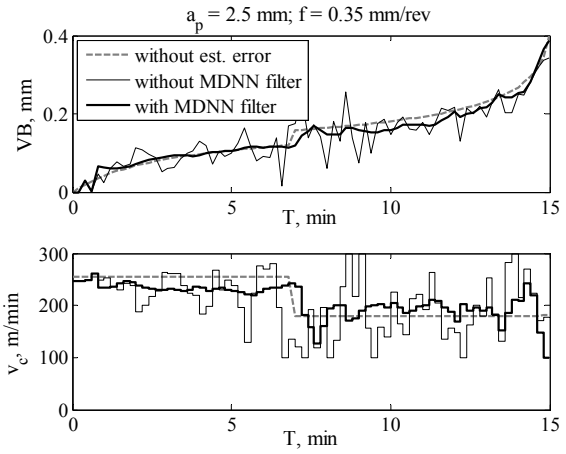

Fig. 5. TWR results for VB_{MAX}= 0.4 mm and T_M= 15 min.

within the same intervals but with different values than the one used in the learning process.

The simulation step (i.e., period of generating a new control variable) is set to T_S= 0.2 min. Generally, its value depends on the characteristics of the tool wear estimation module, tool wear dynamics, and the required quality of the control process. It must be small enough to provide a fast response to different types of disturbances and to ensure a referent tool state, especially in cases of very intense tool wear. On the other hand, it must be high enough to ensure quality tool wear estimation with reduced cutting speed variations and required on-line controller structure adaptation.

From the set of obtained simulation results, several cases of TWR model response are chosen and presented hereafter. In the first two cases, the estimation errors are within the interval of $\pm$0.1 mm. The control model has accomplished the desired tool wear state using a constant depth of cut and feed value (Figs. 5, 6). In contrast to the second case (Fig. 6), impulse disturbance (δ_i^I = 1.3, i = 35) has been added to the tool wear model in the first example (Fig. 5). It is manifested as an additional growth in VB parameter for 0.037 mm in T=7 min. In both cases, it is obvious that the controller has successfully

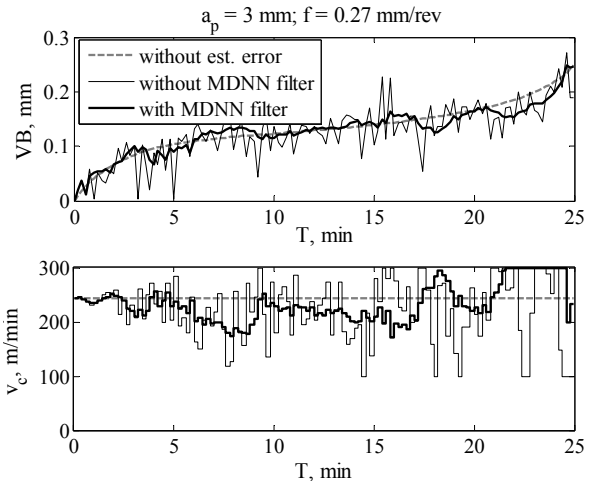

Fig. 6. TWR results for VB_{MAX}= 0.25 mm and T_M= 25 min.

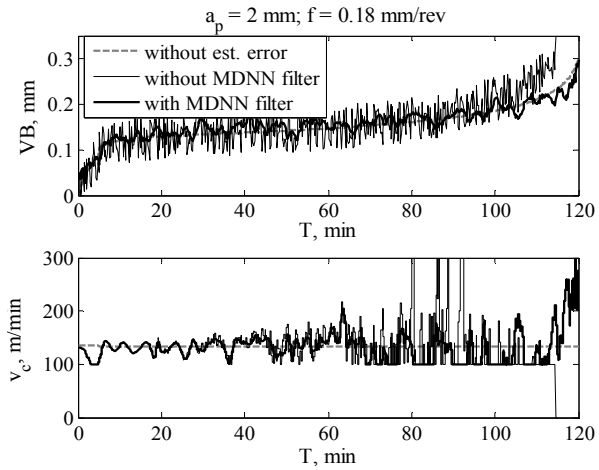

Fig. 7. TWR results for VB_{MAX}= 0.3 mm and T_M= 120 min.

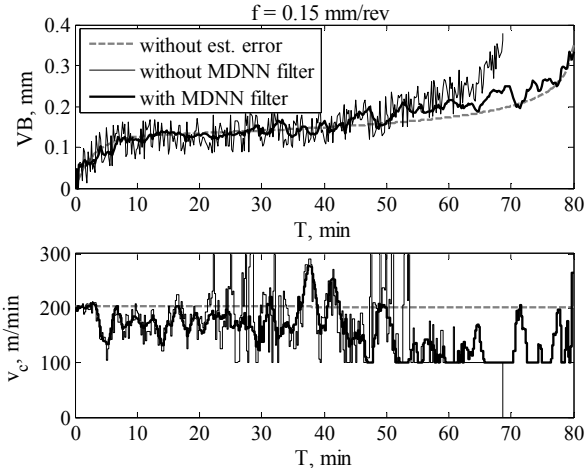

Fig. 8. TWR results for VB_{MAX}= 0.35 mm and T_M= 80 min.

maintained a tool wear rate inside the set up bounds, with good response on impulse disturbance. Furthermore, the control model with MDNN filter achieves better results with re-

duced oscillations of the control variable and a smaller control error.

The influence of estimation errors is particularly noticeable in the next two cases presented in Figs. 7, 8. In these cases, the estimation errors are within the interval of ± 0.05 mm. In the first one (Fig. 7), the depth of cut and feed are also kept at a constant value, and impulse disturbance ($\delta_i^I = 1.2$, $i = 25$) is added, which influences the growth in the *VB* parameter for 0.019 mm in T=5 min. In the second case (Fig. 8), the depth of cut is linearly reduced from a_p = 4 mm at the beginning of the cutting process to a_p = 2 mm at the end. Although the interval of estimation error is smaller than in the first two examples, the control model without filtration does not manage to reach the desired referent tool states because of the mutual influence of too high cutting speed oscillations and tool wear process dynamics.

7. Conclusions

A new approach in designing the tool wear regulation model has been proposed. It is built upon two types of neural networks. The first one is tasked to filtrate the output signals of the tool wear estimator, thus providing the necessary preconditions for the second one to generate an adequate control variable. Due to unavoidable and sometimes relatively high estimation errors, it has been shown that the implementation of the proposed filter is necessary to successfully reduce control variable oscillations and control error in achieving the maximum allowed tool wear intensity in the predefined machining time. The filter is structured using a recurrent type of neural network with good filtering capabilities because of every hidden layer neuron is built in the form of an IIR filter. The controller is configured in the form of a RBF neural network because of its good approximation properties used in the calculation of cutting speeds, which could enable high machining process productivity. Both types of networks adjust their structures using fast learning algorithms, and they can be adapted to various forms of tool wear dynamics. Their learning procedures do not require additional sets of process data other than the one used for the purpose of tool wear estimator configuration.

Acknowledgment

The authors wish to thank the Ministry of Science, Education and Sport of the Republic of Croatia for funding this research through national scientific projects.

References

[1] S. Liang, R. L. Hecker and R. G. Landers, Machining Process Monitoring and Control: The State–of–the–Art, *ASME Journal of Manufacturing Science and Engineering*, 126 (2004) 297-310.

[2] Y. Koren, *Computer Control of Manufacturing Systems*, Mc Grow-Hill Inc., New York, USA, (1983).

[3] R. G. Landers, A. G. Ulsoy and R. J. Furness, Process Monitoring and Control of Machining Operations, *Mechanical Systems Design Handbook,* CRC Press, USA, (2002).

[4] J. C. Filho and A. E. Diniz, Influence of Cutting Conditions on Tool Life, Tool Wear and Surface Finish in the Face Milling Process, *Journal of the Brazilian Society of Mechanical Sciences*, 24 (1) (2002) 10-14.

[5] E. Kilickap, O. Cakir, M. Aksoy and A. Inan, Study of tool wear and surface roughness in machining of homogenised SiC-p reinforced aluminium metal matrix composite, *Journal of Materials Processing Technology*, 164-165 (2005) 862-867.

[6] Y. Koren and E. Lenz, Mathematical model for the Flank Wear while Turning Steel with Carbide Tools, *CIRP Proceedings on Manufacturing Systems*, (1972) 127-139.

[7] K. Danai and A. G. Ulsoy, A Dynamic State Model for On-Line Tool Wear Estimation in Turning, *ASME Journal of Engineering for Industry*, 109 (4) (1987) 396-399.

[8] A. G. Ulsoy, Y. Koren, T. R. Ko and J. Park, Model-Based Tool Wear Estimation in Metal Cutting, *Proceedings of 15th Conference on Production Research and Technology*, University of California, Berkeley, USA, (1989) 237-239.

[9] J.-J. Park and A. G. Ulsoy, On-Line flank wear estimation using an adaptive observer and computer vision - Part 1: Theory, *ASME Journal of Engineering for Industry*, 115 (1993) 30-36.

[10] M. Ayoubi, M. Schefer and S. Sinsel, Dynamic neural units for nonlinear dynamic systems identification, *Lecture Notes in Computer Science*, 930 (1995) 1045-1051.

[11] M. Riedmiller and H. Braun, A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm, Proceedings of the IEEE International Conference on Neural Networks (ICNN), San Francisco, USA, (1993) 586-591.

[12] M. Riedmiller, Advanced Supervised learning in Multi–layer Perceptrons - From Backpropagation to Adaptive Learning Algorithms, International Journal of Computer Standards and Interfaces, Special Issue on Neural Networks, 16 (1994) 265-278.

[13] C. Igel and M. Husken, Improving the RPROP Learning Algorithm, Proceedings of the Second International Symposium on Neural Computation, Berlin, Germany, (2000) 115-121.

[14] J. Leski and E. Czogala, A new artificial neural network based fuzzy inference system with moving consequents in if–then rules and selected applications, Fuzzy Sets and Systems, 108 (1999) 289-297.

[15] D. S. Mitrinovic, J. E. Pecaric and A. E. Fink, Classical and New Inequalities in Analysis, Kluwer Academic Publishers, Netherland, (1993).

Appendix : Optimal cutting speed for productivity maximisation in tool wear regulation

Theoretically, the maximal *MRR* between two tool wear states that can be achieved within the predefined machining time (tool life) is characterised by one optimal cutting speed. This speed is always higher than the average value of any combinations of cutting speeds, which can ensure the required tool wear transition dynamics.

In order to prove the aforementioned statement, Taylor's tool life equation is used:

$$v_c T^n = C \,. \tag{A.1}$$

A combination of only two cutting speeds ($v_{c1} > v_{c2}$) is also used:

$$\begin{aligned} v_{c1}\left[\frac{T-T_A}{T_B-T_A}\right] + v_{c2}\left[\frac{T_B-T}{T_B-T_A}\right] < v_{cR} \quad for\ v_{c1} \to v_{c2}\,, \\ v_{c2}\left[\frac{T-T_A}{T_B-T_A}\right] + v_{c1}\left[\frac{T_B-T}{T_B-T_A}\right] < v_{cR} \quad for\ v_{c2} \to v_{c1}\,, \end{aligned} \tag{A.2}$$

where v_{cR} , for now, is the referent cutting speed characteristic of the tool wear curve, which connects both tool wear states. At this point, it is irrelevant if the cutting speed is first higher ($v_{c1} \rightarrow v_{c2}$) or lower ($v_{c2} \rightarrow v_{c1}$) than the referent one (Figs. A.1-A.2).

Every other combination of cutting speeds can be reduced to the combination of these two; hence, it can be concluded that if the inequalities in Eq. (A.2) are true, the referent cutting speed is actually the optimal one (i.e., it is higher than the average value of any arbitrary chosen combination of cutting speeds).

A.1. Case 1 ($v_{c1} \rightarrow v_{c2}$)

In the first case (Fig. A.1), it is supposed that the cutting speed is first higher (v_{c1}) than the referent one (v_{cR}), and then it decreases to v_{c2}, hence achieving a new tool wear state (VB_B, T_B). The first inequality in Eq. (A.2) can thus be rewritten in the following form:

$$v_{c1}\left(T - T_A\right) + v_{c2}\left(T_B - T\right) < v_{cR}\left(T_B - T_A\right). \tag{A.3}$$

Using the notation from Fig. A.1, it can be rewritten as

$$v_{c1}\Delta T_{v_{c1}} + v_{c2}\Delta T_{v_{c2}} < v_{cR}\left(\Delta T_1 + \Delta T_2\right), \text{ i.e.} \tag{A.4}$$

$$\frac{v_{c1}}{v_{cR}}\Delta T_{v_{c1}} + \frac{v_{c2}}{v_{cR}}\Delta T_{v_{c2}} < \Delta T_1 + \Delta T_2 \,. \tag{A.5}$$

By equalising Taylor's expressions $v_{c1} T_{v_{c1}}^n = C_1$ and $v_{cR} T_{v_{cR}}^n = C_1$, it can be concluded that

$$v_{c1} T_{v_{c1}}^n = v_{cR} T_{v_{cR}}^n \,. \tag{A.6}$$

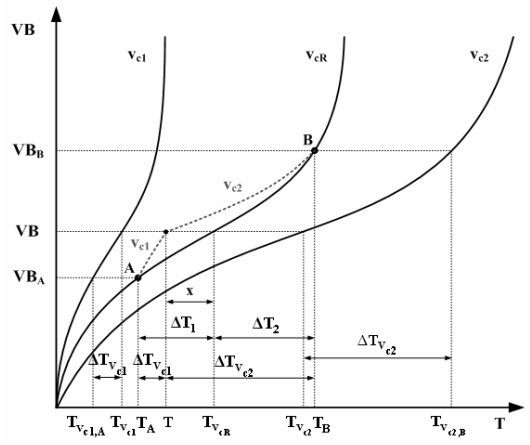

Fig. A.1. Transition from state A to B in the case of $v_{c1} \rightarrow v_{c2}$.

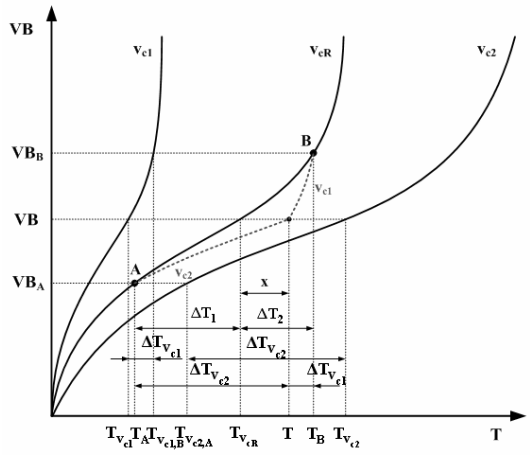

Fig. A.2. Transition from state A to B in the case of $v_{c2} \rightarrow v_{c1}$.

By equalising expressions $v_{c1} T_{v_{c1,A}}^n = C_2$ and $v_{cR} T_A^n = C_2$, we obtain

$$v_{c1} T_{v_{c1,A}}^n = v_{cR} T_A^n \,. \tag{A.7}$$

If Eq. (A.7) is subtracted from Eq. (A.6), the relationship between cutting speeds v_{c1} and v_{cR} can be obtained:

$$v_{c1} \Delta T_{v_{c1}}^n = v_{cR} \Delta T_1^n \,. \tag{A.8}$$

Using the same approach, it is possible to establish the relationship between v_{cR} and v_{c2} where, from $v_{c2} T_{v_{c2,B}}^n = C_3$ and $v_{cR} T_B^n = C_3$, we obtain

$$v_{c2} T_{v_{c2,B}}^n = v_{cR} T_B^n \,, \tag{A.9}$$

and from $v_{c2} T_{v_{c2}}^n = C_1$ and $v_{cR} T_{v_{cR}}^n = C_1$, we derive

$$v_{c2} T_{v_{c2}}^n = v_{cR} T_{v_{cR}}^n \,. \tag{A.10}$$

Finally, by subtracting Eq. (A.10) from (A.9) the relation between these two cutting speeds can be written in the form of

$$v_{c2}\,\Delta T_{v_{c2}}^{n} = v_{cR}\,\Delta T_2^{n}\,. \tag{A.11}$$

If Eq. (A.8) and (A.11) are written in the form of speed ratios, we have

$$\frac{v_{c1}}{v_{cR}} = \frac{\Delta T_1^{n}}{\Delta T_{v_{c1}}^{n}}\,, \quad \frac{v_{c2}}{v_{cR}} = \frac{\Delta T_2^{n}}{\Delta T_{v_{c2}}^{n}} \tag{A.12}$$

and by incorporating Eq. (A.5), we derive

$$\frac{\Delta T_1^{n}}{\Delta T_{v_{c1}}^{n}}\Delta T_{v_{c1}} + \frac{\Delta T_2^{n}}{\Delta T_{v_{c2}}^{n}}\Delta T_{v_{c2}} < \Delta T_1 + \Delta T_2\,, \text{ i.e.} \tag{A.13}$$

$$\Delta T_1^{n}\Delta T_{v_{c1}}^{1-n} + \Delta T_2^{n}\Delta T_{v_{c2}}^{1-n} < \Delta T_1 + \Delta T_2\,. \tag{A.14}$$

Thereafter, if the relations

$$\Delta T_{v_{c1}} = \Delta T_1 - x \quad \text{and} \quad \Delta T_{v_{c2}} = \Delta T_2 + x \tag{A.15}$$

Fig. A.1 are inserted in Eq. (A.14), it expands in the form of

$$\Delta T_1^{n}\left(\Delta T_1 - x\right)^{1-n} + \Delta T_2^{n}\left(\Delta T_2 + x\right)^{1-n} < \Delta T_1 + \Delta T_2\,. \tag{A.16}$$

By rearranging this equation in the form

$$\Delta T_1\left(1 - \frac{x}{\Delta T_1}\right)^{1-n} + \Delta T_2\left(1 + \frac{x}{\Delta T_2}\right)^{1-n} < \Delta T_1 + \Delta T_2\,, \tag{A.17}$$

we can define the final inequality associated with the case where $v_{c1} \rightarrow v_{c2}$

$$\Delta T_1\left[\left(1 - \frac{x}{\Delta T_1}\right)^{1-n} - 1\right] < \Delta T_2\left[1 - \left(1 + \frac{x}{\Delta T_2}\right)^{1-n}\right]. \tag{A.18}$$

A.2. Case 2 ($v_{c2} \rightarrow v_{c1}$)

On the other hand, it is also possible that the tool is first cutting with a slower speed (v_{c2}) than the referent one, and then it increases, hence achieving a desirable tool wear state (VB_B, T_B). From the second inequality in Eq. (A.2), we derive

$$v_{c1}\left(T_B - T\right) + v_{c2}\left(T - T_A\right) < v_{cR}\left(T_B - T_A\right). \tag{A.19}$$

In other words, using the notation from Fig. A.2, the same relation in Eq. (A.5) can be obtained. If Taylor's equation is applied in this case, then

$$v_{c1}\,T_{v_{c1,B}}^{n} = v_{cR}\,T_B^{n}\,, \tag{A.20}$$

$$v_{c1}\,T_{v_{c1}}^{n} = v_{cR}\,T_{v_{cR}}^{n}\,, \tag{A.21}$$

$$v_{c2}\,T_{v_{c2}}^{n} = v_{cR}\,T_{v_{cR}}^{n}\,, \tag{A.22}$$

$$v_{c2}\,T_{v_{c2,A}}^{n} = v_{cR}\,T_A^{n}\,. \tag{A.23}$$

By subtracting Eq. (A.21) from (A.20) and Eq. (A.23) from (A.22), we obtain

$$v_{c1}\,\Delta T_{v_{c1}}^{n} = v_{cR}\,\Delta T_2^{n}\,, \text{ and} \tag{A.24}$$

$$v_{c2}\,\Delta T_{v_{c2}}^{n} = v_{cR}\,\Delta T_1^{n}\,, \tag{A.25}$$

respectively.

If the last two equalities are also written in the form of speed ratios

$$\frac{v_{c1}}{v_{cR}} = \frac{\Delta T_2^{n}}{\Delta T_{v_{c1}}^{n}}\,, \quad \frac{v_{c2}}{v_{cR}} = \frac{\Delta T_1^{n}}{\Delta T_{v_{c2}}^{n}} \tag{A.26}$$

and by incorporating Eq. (A.5), it expands in the form of

$$\frac{\Delta T_2^{n}}{\Delta T_{v_{c1}}^{n}}\Delta T_{v_{c1}} + \frac{\Delta T_1^{n}}{\Delta T_{v_{c2}}^{n}}\Delta T_{v_{c2}} < \Delta T_1 + \Delta T_2\,, \text{ i.e.} \tag{A.27}$$

$$\Delta T_2^{n}\Delta T_{v_{c1}}^{1-n} + \Delta T_1^{n}\Delta T_{v_{c2}}^{1-n} < \Delta T_1 + \Delta T_2\,. \tag{A.28}$$

After the insertion of the relations

$$\Delta T_{v_{c2}} = \Delta T_1 + x \quad \text{and} \quad \Delta T_{v_{c1}} = \Delta T_2 - x \tag{A.29}$$

Fig. A.2 in Eq. (A.28), we obtain the final inequality in the case when $v_{c2} \rightarrow v_{c1}$

$$\Delta T_2^{n}\left(\Delta T_2 - x\right)^{1-n} + \Delta T_1^{n}\left(\Delta T_1 + x\right)^{1-n} < \Delta T_1 + \Delta T_2\,, \tag{A.30}$$

$$\Delta T_2\left(1 - \frac{x}{\Delta T_2}\right)^{1-n} + \Delta T_1\left(1 + \frac{x}{\Delta T_1}\right)^{1-n} < \Delta T_1 + \Delta T_2\,, \tag{A.31}$$

$$\Delta T_1\left[\left(1 + \frac{x}{\Delta T_1}\right)^{1-n} - 1\right] < \Delta T_2\left[1 - \left(1 - \frac{x}{\Delta T_2}\right)^{1-n}\right]. \tag{A.32}$$

A.3. Optimal cutting speed

Eq. (A.18) can now be written in the form of

$$\frac{\Delta T_1}{\Delta T_2}\left[\left(1 - \frac{x}{\Delta T_1}\right)^{1-n} - 1\right] + \left(1 + \frac{x}{\Delta T_2}\right)^{1-n} < 1\,. \tag{A.33}$$

Using variables $d = \dfrac{\Delta T_2}{\Delta T_1}$ and $e_1 = \dfrac{x}{\Delta T_1}$, $x < \Delta T_1$ together with constraints

$$d>0 \rightarrow 0<\left(\Delta T_1, \Delta T_2\right)<\Delta T,\ \Delta T=\left(T_B-T_A\right)>0,$$
$$0<e_1<1, \tag{A.34}$$
$$0<n<1,$$

Eq. (A.33) transforms into

$$\frac{1}{d}\left[\left(1-e_1\right)^{1-n}-1\right]+\left(1+\frac{e_1}{d}\right)^{1-n}<1 . \tag{A.35}$$

Likewise, it is possible to write Eq. (A.32) as

$$\frac{\Delta T_1}{\Delta T_2}\left[\left(1+\frac{x}{\Delta T_1}\right)^{1-n}-1\right]+\left(1-\frac{x}{\Delta T_2}\right)^{1-n}<1 . \tag{A.36}$$

In this case, using $e_2=\frac{x}{\Delta T_2}$, $x<\Delta T_2$, and the same constraints as in Eq. (A.34), except the constraint $0 < e_1 < 1$ that is replaced by $0 < e_2 < 1$, we derive

$$\frac{1}{d}\left[\left(1+e_2 d\right)^{1-n}-1\right]+\left(1-e_2\right)^{1-n}<1 . \tag{A.37}$$

The inequalities in Eq. (A.2) are now transformed into inequalities in Eq. (A.35) and Eq. (A.37), which are subject to constraints in Eq. (A.34) and $0 < e_2 < 1$. The proof that these two inequalities are valid is based on Bernoulli's inequality [15] where

$$\left(1+y\right)^a<1+ay , \tag{A.38}$$

provided that $0<a<1$ and $-1<y\neq 0$. Eq. (A.35) can be transformed into

$$\frac{1}{d}\left[\left(1-e_1\right)^{1-n}-1\right]+1+\left(1-n\right)\frac{e_1}{d}<1 \text{ , i.e.} \tag{A.39}$$
$$\frac{\left(1-e_1\right)^{1-n}-1+\left(1-n\right)e_1}{d}<0 , \tag{A.40}$$

because $0 < (1\text{-}n) < 1$, $(0 < n < 1)$, and $-1<\frac{e_1}{d}\neq 0$, $(0 < e_1 < 1, d > 0)$. With $d > 0$ the numerator must be

$$\left(1-e_1\right)^{1-n}-1+\left(1-n\right)e_1<0 . \tag{A.41}$$

If the numerator is written as

$$D_1-1+\left(1-n\right)e_1<0 , \tag{A.42}$$

where $D_1=\left(1-e_1\right)^{1-n}$, it turns out that in the case when

$$D_1=\left(1-e_1\right)^{1-n}=1-\left(1-n\right)e_1 , \tag{A.43}$$

the left side of Eq. (A.41) will be equal to zero. However, because from Eq. (A.38)

$$D_1=\left(1-e_1\right)^{1-n}<1-\left(1-n\right)e_1 , \tag{A.44}$$

it is clear that the inequality in Eq. (A.41), and consequently in Eq. (A.35), are valid.

Likewise, Eq. (A.37) can be reformulated into

$$\frac{1}{d}\left[\left(1+e_2 d\right)^{1-n}-1\right]+1-\left(1-n\right)e_2<1 \text{ , i.e.} \tag{A.45}$$
$$\frac{\left(1+e_2 d\right)^{1-n}-1-\left(1-n\right)e_2 d}{d}<0 . \tag{A.46}$$

Given that $d > 0$, the numerator must be

$$\left(1+e_2 d\right)^{1-n}-1-\left(1-n\right)e_2 d<0 . \tag{A.47}$$

Using parameter $-D_2=\left(1+e_2 d\right)^{1-n}$, it can be written as

$$D_2-1-\left(1-n\right)e_2 d<0 . \tag{A.48}$$

Given that

$$D_2=\left(1+e_2 d\right)^{1-n}=1+\left(1-n\right)e_2 d , \tag{A.49}$$

the left side of Eq. (A.47) will also be equal to zero. According to Eq. (A.38),

$$D_2=\left(1+e_2 d\right)^{1-n}<1+\left(1-n\right)e_2 d . \tag{A.50}$$

Hence, it is obvious that the inequalities in Eq. (A.47), and consequently in Eq. (A.37), are true. By proving the validity of the inequalities in Eq. (A.35) and Eq. (A.37), the inequalities in Eq. (A.2) are also finally proven to be valid, which means that the referent cutting speed (v_{cR}) is the optimal speed in the productivity maximization criterion in the tool wear regulation process.

Danko Brezak received his B.Sc., M.Sc. and Ph.D. in Mechanical Engineering from the Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Croatia in 1998, 2003 and 2007, respectively. He is currently a senior research assistant in the Department of Robotics and Production Systems Automation at the Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb. His research interests include neural networks, fuzzy logic systems, machining process monitoring and control algorithms.

Dubravko Majetic received his B.Sc., M.Sc. and Ph.D. in Mechanical Engineering from the University of Zagreb, Croatia, in 1988, 1992 and 1996, respectively. He is currently a professor in the Department of Robotics and Production Systems Automation at the Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb. His research is oriented towards automatic control and artificial intelligence.

Toma Udiljak finished Mechanical Engineering at the University of Zagreb, Croatia, where he also received his M.Sc. in 1988 and Ph.D. in 1996. He became a professor in 2004. He is currently serving as head of the Machine Tools Chair. His research is oriented towards cutting theory, CAM, and autonomous manufacturing systems.

Josip Kasac received his B.Sc. in Physics from the University of Zagreb in 1995. He then received his M.S. and Ph.D. in Mechanical Engineering from the University of Zagreb in 1998 and 2005, respectively. He is currently an Assistant Professor in the Department of Robotics and Production Systems Automation at the Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Croatia. His research interests include robot control, optimal control, neural network, and fuzzy control.