Dynamic Model of the Machining Process on the Basis of Neural Networks: from Simulation to Real Time Application

*Rodolfo E. Haber[1,3], R. H. Haber[2], A. Alique[1], S. Ros[1], J.R. Alique[1]

[1]Instituto de Automática Industrial (CSIC).
km. 22,800 N-III, La Poveda. 28500. Madrid.
SPAIN.
rhaber@iai.csic.es
[2]Departamento de Control Automático.
Universidad de Oriente. 90400.
CUBA
[3] School of Computer Science and Engineering.
Universidad Autónoma de Madrid.
Ciudad Universitaria de Cantoblanco
Ctra. de Colmenar Viejo,km 15. 28049 - Madrid
SPAIN
Rodolfo.Haber@ii.uam.es

Abstract. Nowadays, the modeling of complex manufacturing tasks is a key issue. In this work, as a case study is selected the application of a dynamic model to predict cutting force in machining processes. A model created using Artificial Neural Networks (ANN), able to predict the process output is introduced in order to deal with the characteristics of such an ill-defined process. This model describes the dynamic response of the output before changes in the process input command (feed rate) and process parameters (depth of cut). Experimental tests are made in a professional machining centre, with different cutting conditions, on real time data. The model provides sufficiently accurate prediction of cutting force, since the process-dependent specific dynamic properties are adequately reflected.

1 Introduction

New trends in modeling nonlinear processes have demonstrated that Artificial Neural Network (ANN) based algorithms are maybe the most suitable Artificial Intelligent technique to analyze the sensory information and depict process behavior. ANN represents an outstanding approach to model complex processes due to the feasibility for hardware implementation, real-time running, and a few prior assumptions for modeling [1]. Some results show that ANN can yield a more accurate process model than using the regression method [2]. In spite of its excellent interpolation capability, there are difficulties related with the poor extrapolation accuracy of the typical neural

---

* Corresponding author

P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS 2331, pp. 574–583, 2002.

networks and it is necessary to properly train models, requiring experimentation with a wide range of possible working conditions [3].

Among the industrial sectors, today's manufacturing industry has growing up demanding productivity and profitability requirements. Such demands can be satisfied only if the production systems are highly automated and extremely flexible. One of the main activities the manufacturing industry has to deal with is machining. In machining, as in any production process, an optimal performance is always a desirable feature. However, most machining processes such as milling, drilling, grinding and turning exhibit nonlinear and non-stationary behavior that make hard to perform optimization tasks.

The cutting force model of machining process has been extensively studied both analytically and empirically because of its invaluable importance for the assessment of cutting tool deflection, wear, breakage, vibration, and their effects on the quality of the elaborated part. The spectrum of available *conventional* methods for modeling is very wide. However, the different types of models developed up until only a few years have not lived up to expectations. Theoretical models are deduced by making certain approximations, which limits their validity and they may be cumbersome to handle. The empirical models thus developed far are valid only for the experimental conditions under which they were developed, which do not always coincide with industry's actual working conditions. Stochastic models cannot always make adequate prediction beforehand due to the vast variability of their estimates and the numerous variables involved in the complex processes of machining.

Presented in this paper is an intelligent model of the milling process. In order to deal with nonlinear process characteristics, a Neural Network Output Error (NNOE) model, able to predict online the resultant cutting force under actual cutting conditions, is proposed. In section 2 a brief presentation of neural networks that focuses on the type of ANN used, including training algorithm is given. In section 3, a short description of the machining process is shown. In section 4 the experimental set-up, the design considerations, the network topology for the best training result obtained and the experimental results are presented. Finally, the authors conclude on the model suitability and some suggestions for its further improvement and future works.

2 Artificial Neural Networks and Dynamic Process Models

Among all modelling structures, the so-called Output Error (OE) model is one of the most widely used. In this configuration noise is assumed to corrupt the process additively at the output. OE models are often more realistic models and therefore they often perform better than other configurations.

$$\hat{y}(t) = g(\varphi(t), \theta). \tag{1}$$

where $\varphi(t) = [\hat{y}(t-1|\theta), \dots, \hat{y}(t-n_A|\theta), u(t-1), \dots, u(t-n_B)]^T$ is a regression vector, n_A is the number of past predictions used for determining the predictions, n_B is the number of past inputs, $\boldsymbol{\theta}$ is the vector containing the weights and $g(\cdot)$ is the function performed by the neural network.

One of the most widely used ANN paradigms, for its suitability for modelling and control applications, is the so-called Multilayer Perceptron (MLP). The class of MLP considered here consists in only one hidden layer with hyperbolic tangent activation function *H*, and a linear activation function, *L*, at the output. Such configuration is advantageous considering a foreseeable real-time implementation of the network.

$$\hat{y}_i(\mathbf{w},\mathbf{W}) = L_i\left[\sum_{j=1}^{Q} W_{ij} H_j\left(\sum_{k=1}^{M} w_{jk} u_k + w_{j0}\right) + W_{i0}\right]. \tag{2}$$

where Q is the number of output neurons, M is the number of neurons in the hidden layer, $\mathbf{u}$ are the inputs, and $\hat{y}_i$ is the output of the network. The weights are specified by the matrices $\mathbf{w}$ (input-to-hidden layer weights) and $\mathbf{W}$ (hidden-to-output layer weights). Both matrices are included in $\boldsymbol{\theta}$.

The identification problem can be viewed as the determination of the mapping from the set of data $z^{\mathrm{N}} = [\mathbf{u} \quad \mathbf{y}]^{\mathrm{T}}$ (training set) to the set of possible weights ($\hat{\boldsymbol{\theta}}$) so that the network can produce a prediction $\hat{y}(t)$ as close as possible to the actual output $y(t)$ [4].

$$z^N \rightarrow \hat{\boldsymbol{\theta}}. \tag{3}$$

Using a prediction error identification method

$$E(\boldsymbol{\theta},z^{\mathrm{N}}) = \frac{1}{2N}\sum_{t=1}^{N}\left(y(t)-\hat{y}(t|\boldsymbol{\theta})\right)^{\mathrm{T}}\left(y(t)-\hat{y}(t|\boldsymbol{\theta})\right). \tag{4}$$

The weights are calculated as

$$\hat{\boldsymbol{\theta}} = argmin\left(E(\boldsymbol{\theta},z^{\mathrm{N}})\right). \tag{5}$$

As the training algorithm a version of the Levenberg-Marquardt method was selected [5].

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} + \mu^{(i)} \cdot \mathbf{S_D}^{(i)}. \tag{6}$$

where the search direction is calculated from

$$\mathbf{S_D}^{(i)} = -\left[\mathbf{R}\left(\boldsymbol{\theta}^{(i)}\right) + \lambda^{(i)}\mathbf{I}\right]^{-1}\mathbf{d}\left(\boldsymbol{\theta}^{(i)}\right). \tag{7}$$

where $\mathbf{R}$ is the Hessian approximation, λ a positive scalar, μ the step size for iterations, $\mathbf{d}$ the gradient of the predictions computed with respect to the weights, and $\mathbf{I}$ the identity matrix. The size of the elements of the diagonal matrix added to $\mathbf{R}$ is adjusted according to the size of the ratio between actual and predicted decrease. For a step to be accepted the ratio

$$\rho^{(i)} = \frac{E\left(\boldsymbol{\theta}^{(i)}, z^{\mathrm{N}}\right) - E\left(\boldsymbol{\theta}^{(i)} + \mathbf{S}_{\mathrm{D}}{}^{(i)}, z^{\mathrm{N}}\right)}{E\left(\boldsymbol{\theta}^{(i)}, z^{\mathrm{N}}\right) - \Lambda\left(\boldsymbol{\theta}^{(i)} + \mathbf{S}_{\mathrm{D}}\right)} . \tag{8}$$

must exceed some small positive number, where

$$\Lambda\left(\boldsymbol{\theta}^{(i)} + \mathbf{S}_{\mathrm{D}}\right) = \left\| y(t) - \hat{y}\left(t, \boldsymbol{\theta}^{(i)}\right) - \mathbf{S}_{\mathrm{D}}{}^{\mathrm{T}} \left[\frac{\partial \hat{y}(t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right]_{\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}} \right\|_2^2 . \tag{9}$$

$$\Lambda\left(\boldsymbol{\theta}^{(i)} + \mathbf{S}_{\mathrm{D}}\right) = E\left(\boldsymbol{\theta}^{(i)}, z^{\mathrm{N}}\right) + \mathbf{S}_{\mathrm{D}}{}^{\mathrm{T}} \mathbf{d}\left(\boldsymbol{\theta}^{(i)}\right) + \tfrac{1}{2} \mathbf{S}_{\mathrm{D}}{}^{\mathrm{T}} \mathbf{R}\left(\boldsymbol{\theta}^{(i)}\right) \mathbf{S}_{\mathrm{D}} . \tag{10}$$

The algorithm adjusts λ according to whether $E\left(\boldsymbol{\theta}^{(i)}, z^{\mathrm{N}}\right)$ is increasing or decreasing.

After this brief explanation, becomes obvious that the attention is focused on determining a nonlinear OE model, i.e., a dynamic system, by training a two layer neural network with a Levenberg-Marquardt method.

3 The Machining Process

For the case study, among the various machining operations, is selected the milling process (see figure 1). This choice obeys a pessimistic criterion, since milling is one of the most complex of machining operations [6].

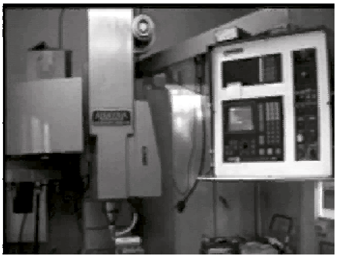


Fig. 1. Overall View of a Typical Machining Centre

Among the enormous quantity of variables and parameters involved in the machining process, the most relevant factors (see fig. 2a), in terms of control tasks are:

The spatial position of the cutting tool, considering the Cartesian axes (x_p, y_p, z_p),

Spindle speed (s) [*rpm*],

Relative feed speed between tool and worktable (f, feed rate) [*mm/min*],

Cutting power invested in removing metal chips from the workpiece (P_c)[*kW*],

Cutting force exerted during the removal of metal chips (*F*, cutting force) [*N*],
Radial depth of cut (*a*, cutting depth) [*mm*],
Diameter of the cutting tool (*d*) [*mm*].

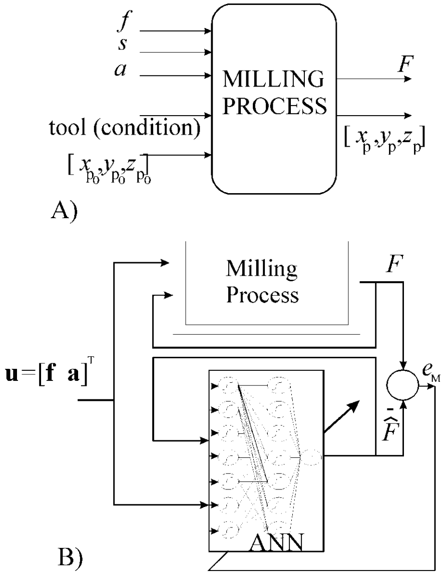


Fig. 2. a) Milling process representation b) Scheme for identification

On the basis of the previous system approach and aiming at intelligent supervision, the milling process can be formally described by a discrete nonlinear relationship

$$\hat{F}(t) = G(\mathbf{F},\mathbf{f},\mathbf{a}). \tag{11}$$

where G is an unknown function to identify and $\mathbf{f},\mathbf{a}$ and $\mathbf{F}$, are the inputs and output respectively defined as $\mathbf{F} = \left[F(t-1) \ \cdots \ F(t-n)\right]$, $\mathbf{f} = \left[f(t-1), \ \cdots \ ,f(t-m)\right]$, $\mathbf{a} = \left[a(t-1) \ \cdots \ a(t-m)\right]$, *t*, is the discrete time instant and $n, m \in Z$.

If the model is simulated by means of an ANN parallel-identification scheme, the resulting estimated output is

$$\hat{F}(t) = g(\hat{\mathbf{F}},\mathbf{f},\mathbf{a}). \tag{12}$$

where g represents the ANN input-output mapping, $\hat{\mathbf{F}} = \left[\hat{F}(t-1), \ \cdots \ ,\hat{F}(t-n)\right]$ and $\hat{F}(t)$ is the one-step prediction of the model.

4 Experimental Set-up

The experimental tests are conducted on a 5.8kW-4 axes milling machine equipped with CNC, which is interfaced with a personal computer by an RS-232 communication link. The tailor-made architecture is illustrated in figure 3.

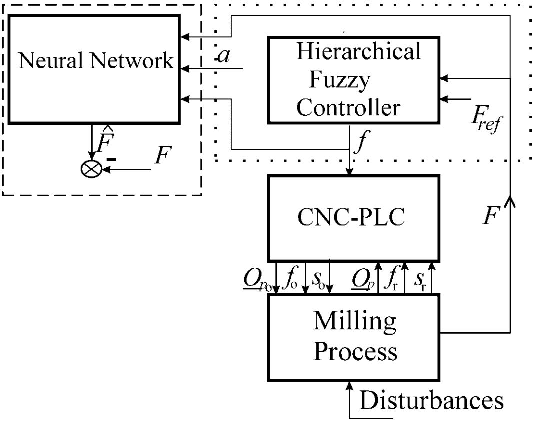


Fig. 3. Scheme for control and modeling of milling process

A personal computer is added in order to carry two important tasks. Firstly, to evaluate the measured values of the cutting force F and to perform the generation of f values in the block named hierarchical fuzzy controller. Details concerning hierarchical control system can be found in [7]. Secondly, and the main goal of this work, is to implement a neural network model (12). The model objective is to predict in real-time, one step ahead, the resultant mean cutting force $\hat{F}$. The input data for the model identification are obtained from the feed rate command signal f, the depth of cut a and the cutting force F.

For the training phase of the ANN, only new milling cutters 25 mm in diameter are used. Two workpieces with several changes in a are chosen for training the ANN (see fig. 4A) and online running of the model (see fig. 5B). Slot milling operation is supposed to be done in one direction only (see fig. 4C).

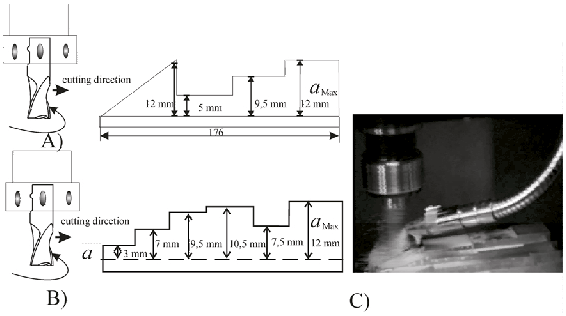


Fig. 4. Workpieces for training (A), and validation tests (B), (C) picture of slot milling operation on profile (A)

4.1 Model of the Milling Process. Training and Verification Methodology

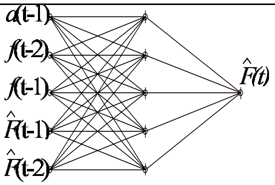
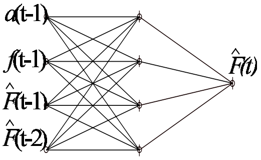
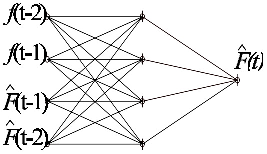
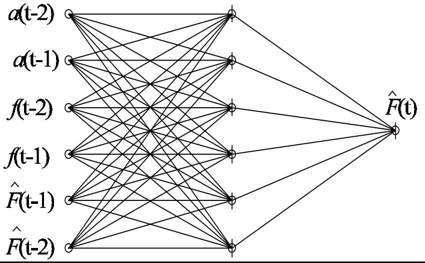
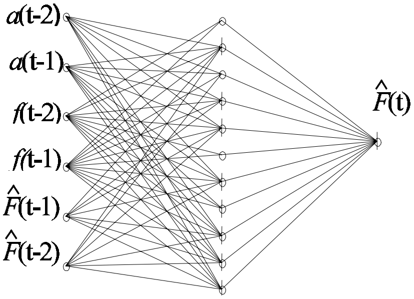
A preliminary processing was applied to the data before performing the training procedure (e.g., standardisation, mean value and standard deviation of the signals) [8]. The training algorithm was developed using the program MATLAB. The topology was initially chosen as follows: two inputs f and a, one output $\hat{F}$, a linear activation function at the output, and one hidden layer using hyperbolic tangent for the activation function. The type of model was selected starting from the *a priori* knowledge of the milling process and the types of models considered in previous works (see fig. 2a). An ANN with 6 neurons in the input, 12 neurons in the hidden layer and one in the output layer, was selected.

Modifying (12) for a second order output error model, the one step prediction is evaluated with the previous model outputs

$$\hat{F}(t)=g\left(\hat{F}(t-1),\ \hat{F}(t-2),\ f(t-1),\ f(t-2),\ a(t-1),\ a(t-2)\right). \tag{13}$$

The initial values of the weights were randomly chosen. The initial cutting conditions used were f_0=100 mm/min., s_0=1000 rpm., a_0=6 mm. At the end of the training stage a pruning algorithm was performed in order to optimise the size of the network, removing the superfluous weights [9]. In order to validate the model, data analysis (mean squared value analysis, auto correlation analysis and transient data study) were done. The robustness property of the model, considering changes in cutting conditions (material hardness, tool wear, and disturbances in depth of cut), is also examined. Additionally, real-time performance of the model before these same situations was investigated. The comparison among different models analysed on the basis of final prediction error (FPE) is depicted in Table 1.

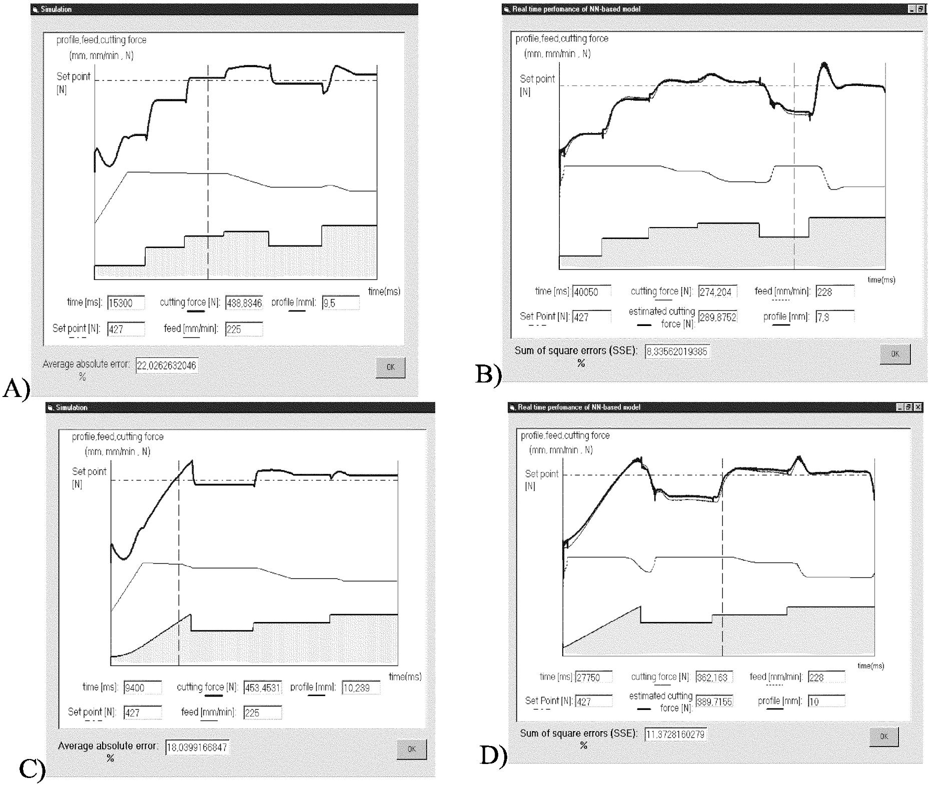
Table 1. Neural Network Output Error (NNOE) models.

	Final Prediction Error (%)	
Models	Training	Verification
 a(t-1) f(t-2) f(t-1) $\hat{F}$(t-1) $\hat{F}$(t-2) $\hat{F}$(t)	14.3	46.1
 a(t-1) f(t-1) $\hat{F}$(t-1) $\hat{F}$(t-2) $\hat{F}$(t)	18.3	50.4
 f(t-2) f(t-1) $\hat{F}$(t-1) $\hat{F}$(t-2) $\hat{F}$(t)	11.3	23.5
 a(t-2) a(t-1) f(t-2) f(t-1) $\hat{F}$(t-1) $\hat{F}$(t-2) $\hat{F}$(t)	13.2	16.3
 a(t-2) a(t-1) f(t-2) f(t-1) $\hat{F}$(t-1) $\hat{F}$(t-2) $\hat{F}$(t)	8.7	9.4

In order to evaluate the behaviour of the model considering actual working conditions, tailor-made software for simulation and real time supervision of machine tools was developed. The application for Windows NT was programmed in Visual C++. The libraries of controllers (e.g., different types of fuzzy controllers) and process models (e.g., neural networks) were programmed in C++ and compiled in two dynamic link libraries (DLL).

For the sake of clarity, only the results of two simulations and two real-time applications are shown in figure 5. The vector of initial conditions used was $\hat{F}(\mathrm{t}=1)=g(100\mathrm{N}, 75\,\mathrm{N}, 9\,\mathrm{mm/min}, 0\,\mathrm{mm/min}, 3\mathrm{mm}, 0)$.

In fig. 5 the horizontal dot line means the set point of cutting force control system (not addressed in this work, but the software includes this option). The vertical line represents the actual (simulation or real-time values) values shown at the bottom of the dialog in text boxes. For each case considered, the chosen profile is depicted at the bottom of the picture. The feed rate command signal is shown in the middle of the graph. The behaviour of predicted and measured cutting forces is shown at the top of the picture. For simulations, only one signal is depicted, which corresponds to the output of the process model.


Simulation
profile,feed,cutting force
(mm, mm/min , N)
Set point
[N]
time(ms)
OK
A)
Real time perfomance of NN-based model
profile,feed,cutting force
(mm, mm/min , N)
Set point
[N]
time(ms)
OK
B)
Simulation
profile,feed,cutting force
(mm, mm/min , N)
Set point
[N]
time(ms)
OK
C)
Real time perfomance of NN-based model
profile,feed,cutting force
(mm, mm/min , N)
Set point
[N]
time(ms)
OK
D)

A)
time [ms]: 15300
cutting force [N]: 438,8346
profile [mm]: 9,5
Set point [N]: 427
feed [mm/min]: 225
Average absolute error %: 22,0262632046

B)
time [ms]: 40050
cutting force [N]: 274,204
feed [mm/min]: 228
Set Point [N]: 427
estimated cutting force [N]: 289,8752
profile [mm]: 7,3
Sum of square errors (SSE) %: 8,33562019385

C)
time [ms]: 9400
cutting force [N]: 453,4531
profile [mm]: 10,239
Set point [N]: 427
feed [mm/min]: 225
Average absolute error %: 18,0399166847

D)
time [ms]: 27750
cutting force [N]: 362,163
feed [mm/min]: 228
Set Point [N]: 427
estimated cutting force [N]: 389,7155
profile [mm]: 10
Sum of square errors (SSE) %: 11,3728160279

Fig. 5. Simulation and real-time running using different workpiece

The simulation serves also for evaluating the performance of the intelligent control system (i.e., accuracy, closed-loop stability, etc.) which is an important step to design closed-loop control systems under varying milling process dynamics (see figure 5A and 5C). On the other hand, the dynamic behaviour of the cutting force provides information about tool and process conditions (see figure 5B and 5D). Peaks and undesirable oscillations in cutting force can be the result of inappropriate cutting conditions as well as the change in tool condition. Good properties of the model can be inferred from these results, as well as its generalisation capability. This dynamic neural network-based model provides sufficiently accurate prediction of cutting force, since the process-dependent specific dynamic properties of milling operation are adequately reflected.

Conclusion

Various models of the milling process were obtained using the parallel identification scheme. The real time test of ANN-based models demonstrates how, through simple topologies, quite precise models can be obtained. These models can play an important role in the design and evaluation of controller performance under variations in the milling process dynamics. The results can be used to develop complex adaptive controllers and new monitoring systems.

The strength of the proposed approach lies in the solution provided for modeling. The neural network is able to describe the nonlinear characteristics of the milling process, showing superior learning capability, a suitable noise suppression and inclusion of non-measurable information (e.g., workpiece hardness, material). Drawbacks however are the need of neural network retraining and the building of a wider library of models. In spite of that a sufficient exciting of signals avoids a high number of retraining stages.

References

1. Haykin S. Neural Networks: A comprehensive foundation. 2nd edn. IEEE Press (1999).
2. Das S., Chattopadhyay A.B., Murthy A.S.R. Force parameters for on- line tool wear estimation - a neural network approach. Neural Networks 9(9) (1996) 1639-1645.
3. Dimla D.E., Lister P.M., Leighton N.J. Neural network solutions to the tool condition monitoring problem in metal cutting – a critical review of methods. International Journal of Machine Tools and Manufacturing 37(9) (1997) 1219-1241.
4. Ljung L., System Identification: Theory for the user: 2nd. Edn. Prentice Hall, (1999).
5. Fletcher R., Practical Methods of Optimization. 2nd. Edition, John Wiley&Sons, (2000).
6. King R.J. Handbook of High Speed Machining Technology. Mc.Graw Hill, New York, (1985).
7. Haber R.E., Peres C.R., et al. Towards intelligent machining: hierarchical fuzzy control for end milling process: IEEE Transactions on Control Systems Technology 2(6) (1998) 188-199.
8. Bendat J.S., Piersol A. G. Random Data, Analysis and Measurement Procedures. 3rd. Edition, Wiley & Sons, (2000).
9. B. Hassibi, et al. Optimal brain surgeon: Extensions, streamlining and performance comparisons, Advances in Neural Information Processing Systems 6 (1994) 263-271.