METHODOLOGY OF NEURAL NETWORK BASED MODELING OF MACHINING PROCESSES

Miroslav Radovanovic [1] & Milos Madic [1]

[1] Faculty of Mechanical Engineering, University of Nis, Serbia, Aleksandra Medvedeva 14, 18000 Nis, Serbia

Corresponding author: Miroslav Radovanovic, mirado@masfak.ni.ac.rs

Abstract: Machining is the most important and widely used manufacturing process. As machining is very complex process, in recent years neural network based modeling has been preferred modeling of machining processes. This paper outlines and discusses the basic idea and concept of neural network modeling of machining processes. Furthermore this paper discusses the methodology of developing neural network model as well as proposing some guidelines for selecting the network training parameters and network architecture. For illustration purpose, simple neural prediction model for cutting power was developed and validated.

Key words: Modeling, machining, neural networks.

1. INTRODUCTION

Modeling of machining, aimed at better understanding of process, has attracted many researchers in past years. As a result, modeling of machining processes has been examined from different points of view and using different techniques. Great number of parameters, partially understood relations between parameters, complex, multidimensional, non-linear and stochastic nature of machining make modeling of machining processes a considerable task.

Neural networks (NNs) possess many characteristics which make them suitable for addressing such tasks: universal function approximation capability, resistance to noisy or missing data, accommodation of multiple nonlinear variables for unknown interactions, good generalization capability, adaptive nature etc. In past 20 years a number of researchers successfuly applied NN based modeling of various machining processes. The most of the models are for surface finish and dimensional deviation, tool life and tool wear and for cutting forces, although there are some for other specific purposes. Despite the fact that NNs have been widely implemented in many areas during the past years, there are still certain issues regarding the development of an optimal model. Until now there have been no clear rules that could serve as a basis to be followed in developing the adequate model. The most typical method followed is a repetitive trial-and-error method, where a large number of different models are examined and compared to one another.

In this paper are discussed the most important steps in neural network model development. Furthermore, there are presented some guidelines for selecting the training parameters of multi-layered perceptron (MLP) trained by backpropagation (BP) algorithm and for selecting neural network architecture. The guidelines are based on literature reviews, empirical rules and previous experience. The methodology of neural network modeling was illustrated on a simple example for modeling of cutting power.

2. MODELING OF MACHINING PROCESSES

In the field of machining very complex and highly nonlinear processes are involved. Machining processes are dependable on various known but also unknown parameters as well as on theirs interactions. The machining process is performed in the specific tribo-mechanical system consisting of five basic groups of input parameters with associated elements (figure 1).

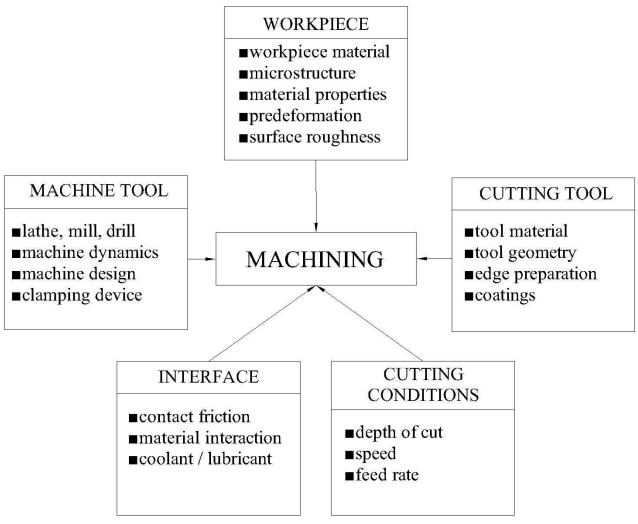

Fig. 1. Machining input parameters

In machining processes practically any parameter can be varied in a wide range. Nevertheless, in many

cases, the true relationship between these parameters and the machining process is not fully understood. The machining process consists of eight groups of output parameters (figure 2).

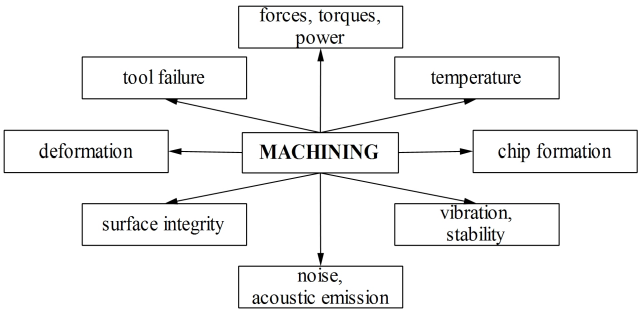

Fig. 2. Machining output parameters

For the past 50 years metal cutting researchers have developed many modeling techniques including analytical techniques, slip-line solutions, empirical approaches, finite element techniques etc. There are many different machining operations and for each of them different aspects can be modelled and different techniques for modelling can be used. Consequently, there exist a wide variety of models dealing in one way or another with machining operations. Models are classified according to: type of operation, main purpose, predicted quantity, time of application and modeling technique. Taking into account the main purpose the models can be classified as: descriptive, predictive, and learning. Modelling technique the models can be classified as: empirical, semi-empirical, and physical (Van Luttervelt, 2001).
Machining processes does not permit pure analytical physical modeling. Recently, semi-empirical models developed using NNs have become the preferred trend which is applied by most researchers. Predictive models, which are capable of predicting quantitatively the influence of the magnitude of input parameters on the magnitude of output parameters, are of special importance in the machining area.

3. MODELING OF MACHINING PROCESSES USING NEURAL NETWORKS

In recent years, modeling techniques using NNs have attracted attention of practitioners and researchers. The learning ability of nonlinear relationship in a cutting operation without going deep into the mathematical complexity, or prior assumptions on the functional form of the relationship between inputs, in-process parameters and outputs makes NN an attractive alternative choice for many researchers to model cutting processes (Mukherjee & Ray, 2006). Generally, any parameter from the figure 1 can be regarded as machining process input parameter, while any of the parameters from the figure 2 can be selected as a machining process output parameter. This type of modeling is often called direct modeling. NNs are massive parallel systems made up of simple processing units (neurons) there are linked with weighted connections where the knowledge possessed by the networks is held. The diagram for a network with a single neuron is shown in figure 3.

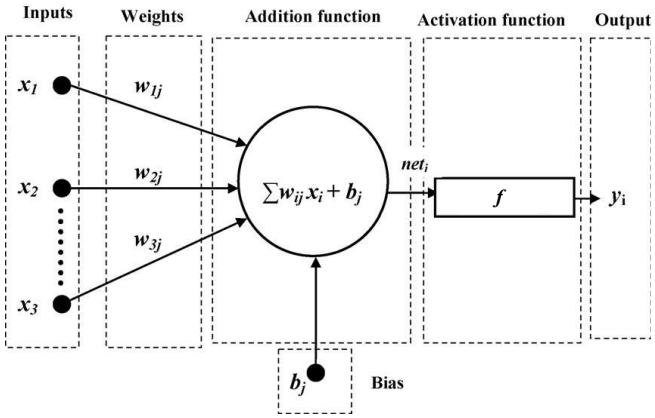

Fig. 3. Structure of artificial neuron (Karayel, 2009)

NNs are characterized by their topology, weight vectors, and activation functions there are used in hidden and output layers of the network. The neurons in the input layer are fed with input data. Each neuron sums its inputs, with one input per neuron in the input layer but many inputs per neuron in the hidden layer. The bias (threshold) is used to scale the input to a useful range to improve the convergence properties of the neural network. Each neuron transfers the data, according to a transfer function (activation function), to all the elements in the next layer. However, each neuron receives a different signal due to different connection weights between the neurons. Finally, the output of each neuron in the output layer is compared to the desired output. In order to minimize the difference between these outputs, weights of the connections between the neurons must be adjusted. There are numerous methods of determining the weights of the connections. Especially there are used some variations of back propagation (BP) learning algorithm, conjugate gradient algorithms, quasi-Newton algorithms and Levenberg–Marquardt (LM) method. A NN is trained using some learning (training) algorithm with a number of data to arrive at an optimum set of weights and tested with other set of data. Once trained, the NN can be used for prediction of output parameters.
The most used NNs for modelling of machining processes are: MLP also known as multilayer feedforward networks, adaptive resonance theory models (ART), self-organizing maps (SOM), radial basis function network (RBFN), etc. The most popular NN for modeling machining processes is MLP with BP training algorithm. MLP BP is adopted by most researches since MLP models are general-purpose models and have good generalization capabilities. The MLP BP networks are most popular

in practice due to their easiness to understand and implement. The standard BP technique with momentum is adopted by most researchers. MLP uses BP algorithm for training the network in a supervised manner. BP algorithm is a steepest descent method, where weight values are adjusted in an iterative fashion while moving along the error surface to arrive at minimal range of error, when input patterns are presented to the network for learning the network. A complete description of the BP algorithm can be found in numerous sources, including (Haykin, 1999). Figure 4 shows three-layered MLP architecture based on the input, hidden and output layers for modeling machining process. The first layer is an input layer where external data are received. The last layer, separated from the input layer by one or more intermediate layers called the hidden layers, computes the network outputs.

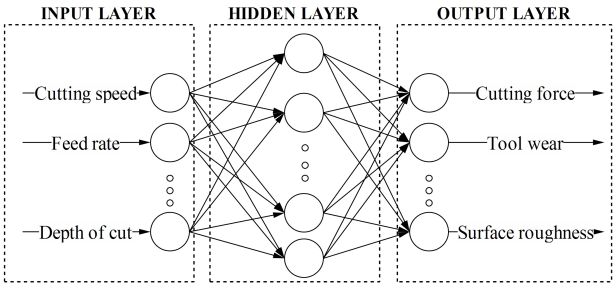

Fig. 4. MLP topology for modeling machining processes

According to the network structure as given in figure 4, decision variables in modeling machining problem could be assigned as the set of input neurons such as the value of cutting speed, feed rate, depth of cut, etc. Response variables are assigned as the set of output neurons such as cutting force, tool wear, surface roughness, etc.

4. MODELING METHODOLOGY OF NEURAL NETWORKS

Development of NN predictive model is a nontrivial task. Modeling issues that affect the performance of an NN must be considered carefully. In order to ensure good performance of an MLP BP models, it is inevitable to develop them in a systematic manner. Such an approach needs to address major factors such as the determination of adequate model inputs, data division and pre-processing, the choice of suitable network architecture, choice of training parameters, etc. Namely, two crucial processes are of importance: model development and issues in selecting MLP parameters, i.e. determining training parameters of BP and topology of MLP.

4.1 Selection of MLP BP network parameters

Configuring MLP architecture usually comprises the decisions about the internal architecture and training parameters.

Internal architecture. Specifying the internal architecture requires determining the number of hidden layers, the number of hidden neurons and transfer functions of the neurons in hidden and output layer. First two parameters especially affect the overall performance of NN i.e. generalization ability of the NN model.

Number of layers. The role of the hidden layers is to map complicated input-output relationships between network inputs and outputs. Until now, no general method for determining the number of hidden layers was proposed. Increasing the number of hidden layers increases the network's processing power, but on the other hand the training time is increased along with more required training examples. Reviewing the literature it could be seen that at most two hidden layers are used. When talking about using NN in machining process modeling and prediction, most authors used only one hidden layer.

Number of hidden neurons. The issue of determining the optimal number of hidden neurons is of crucial importance since the number of neurons determines the "expressive power" of the network. The number of neurons in the hidden layers influences the generalization ability of the network. Adding hidden neurons always results in reduction of the training error, but error decreasing or increasing on test is in direct relationship with actual number of hidden neurons. A typical approach would be to start with a small number of hidden neurons and to slightly increase the number until no significant improvement in model performance is achieved. Using the NN model with minimal number of hidden neurons i.e. simplest architecture is always desirable for both practical and theoretical reasons. It should be underlined that the number of neurons in the hidden layers is data dependent. If the network has more degrees of freedom (the number of connection weights) than the number of training samples, the network is mathematically undetermined. The number of weights is equal to the sum of the product between the numbers of neurons in each layer.

Activation function. The activation function, also called the transfer function, is a mathematical formula that gives the output of a neuron as a function of its input signals. The most used activations functions are: sigmoid (logistic or squashing) function, hyperbolic tangent function, sine or cosine function and linear function. Typically the activation function is chosen based on the kind of data used and type of the layer. For prediction purposes it is common to use a sigmoid or hyperbolic tangent functions in both hidden and output layer or sigmoid or hyperbolic tangent function in hidden layer and linear in the output layer.

Training parameters. Selection of training parameters is very important for proper training of NN. The most important training parameters are: learning coefficient, momentum, epochs and initial weights.

Learning coefficient. Learning coefficient is the rate, at which the network adjusts its weights during training. A high learning coefficient allows the network to learn faster, hence providing faster convergence. With a small learning coefficient, training time is increased, but the probability of reaching the global minimum is increased. Practical guidelines for selecting learning coefficient values are given in (Basheer & Hajmeer, 2000), and (Tarassenko, 1999).

Momentum. Momentum is a training parameter used to reduce training time of the BP algorithm and to enhance the stability of the learning. The method involves adding a term to the weight adjustment that is proportional to the amount of the previous weight change. A high momentum reduces the risk of the network being stuck in local minimum, but it increases the risk of skipping over the solution. Using a small value for momentum will lead to prolonged training. The values of 0.01 for learning rate and 0.9 for momentum are adopted by most researchers because a BP network with these settings has the best prediction performance with the least number of epochs.

Epochs. The epochs of the training cycle is the number of times the training data has been presented to the network. The BP algorithm guarantees that total error in the training set will continue to decrease as the number of epochs increases. With each epoch, the weights are modified to decrease the error on the training patterns. As training progresses, the amount of change in the error function becomes smaller. Convergence occurs when the change in the error function is less than a specified threshold. Similar to the problem of over-fitting, the problem of over-training may occur if the number of epochs is considerably high. The number of epochs required for proper generalization is often determined by trial and error method and cross validation method.

Initial weights. BP network is sensitive to initial values of weights. Too small initial weights will the training time and difficulties in converging to an optimal solution may occur. If initial weights are too large the network may get unstable weights. The initial connection weights must also be specified prior to training. The weight initialization can be set in random way or using evolutionary algorithms. Typically, weights and biases are initialized uniformly in a relatively small range within ±0.5 or ±1.

4.2 NN model development

Several steps need to be considered when developing NN models. Generally, the developing process include: selection of input and output parameters, data collection, data filtering, data pre-processing, selection of training and testing set and model validation and performance measures.

Selection of input and output parameters. The idea of using NN to model the process is to create network that take process parameters as inputs and produces process responses as outputs. In order to developed optimal NN model it is necessary to identify process parameters that affect the process response and to assign them as network inputs. The selection of the suitable response for network outputs requires understanding of the given process. NN can be trained to predict both single-response and multi-response.

Data collection. For a successful application of NN modeling, it needs to collect as much data as. Instead of volume, the quality and representativeness of the collected data is important when NN performance is considered. It is not possible to say how many data items are appropriate, because this depends on the complexity of the modeling problem. Data can be collected in various ways: from simulation, by experimental research or actual process data. In any case, it is necessary to filter and pre-process the data.

Data filtering. Data filtering include data integrity check and extreme data removal. Errors such as incorrectly entered data, duplicated and missing data have to be corrected, because the NN model performance in direct link with the quality of the data. Very extreme data should be also removed because it can interfere with the training process.

Data pre-processing. The NN can only work with data within certain ranges and in specified formats. Data pre-processing usually speeds up the learning process and it usually performed. Pre-processing can be in the form of data scaling (normalization) and transformation. Scaling the data is essential and is closely related to the activation function used. Scaling to [-1,1] for the hyperbolic tangent transfer function and to [0,1] for the sigmoid transfer function is often applied. However, some researchers recommended that the data be normalized between slightly offset values such as [0.1, 0.9] or [0.2, 0.8] rather than between 0 and 1 to avoid saturation of the sigmoid function leading to slow or no learning. In some situations, where parameters have an exceptionally large range, it may be beneficial to take logarithm of data prior to scaling.

Selection of training and testing set. After data collection and pre-processing, all the data should be randomly divided into two sets: training set and testing set. The training set is used for NN model development and the test set is used for estimating the prediction ability of the model. If enough data is available, it is recommended to divide the data set into a training set, a validation set, and a test set. The, validation set (stopping set) is not used for weights update, but to assess the performance of the model. By using a validation set, over-fitting and over-training problems could be avoided. The proportion of training to testing data varied considerably in the published research. It is common to divide all available data so that 90, 80 or 70% are selected for training purposes and the remaining for testing. When

using a validation set, dividing the data into the three parts: 50% for training, and 25% each for validation and testing could be reasonably.

Model validation and performance measures. Once the model is developed, the performance of the trained model should be validated. The purpose of the model validation phase is to assess the model generalization ability. This is achieved by measuring the performance of trained NN on test set, which contain the data that NN has not seen. The most important measure of performance is the prediction accuracy. An accuracy measure is often defined in terms of the prediction error which is the difference between the actual (desired) and the predicted value. There are a number of measures of accuracy in the literature and each has advantages and limitations. They are listed in (Zhang et al., 1998).

The coefficient of correlation (R), the root mean squared error (RMSE) and the mean absolute percentage error (MAPE) are the main criteria that are often used to evaluate the prediction performance of NNs models. The coefficient of correlation is a measure that is used to determine the relative correlation and the goodness-of-fit between the predicted and observed data.

The interpretation of correlation coefficient is as follows:

|R|>=0.8 – strong correlation exists

0.8>=|R|>=0.2 – correlation exists

|R|<=0.2 – weak correlation exists

The RMSE and has the advantage that large errors receive much greater attention than small errors. MAPE is scale less and therefore useful for ease of interpretation, but in contrast to RMSE does not treat small errors symmetrically with large errors. A good NN model should have a correlation coefficient over |0.8| and RMSE and MAPE should be as close to 0 as possible.

5. MODELING OF CUTTING POWER: AN EXAMPLE

The methodology of NN modeling on the example of modeling cutting power is described. Formula for the cutting power are given in figure 5. Since the relationships shown in the data have obvious trends, analytical regression analysis can be used to describe the relationships between inputs and outputs, with much less effort. However, the aim of this example is to present methodology of NN modeling.

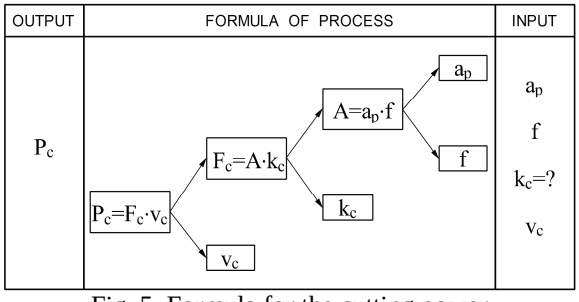

Fig. 5. Formula for the cutting power

The data used for modeling is attained in next conditions: machining operation is longitudinal turning, the work piece material is grey cast-iron, low-tensile, with hardness of HB=180, and ferritic structure (C2.5-4.5%, Si max 3%, P max 1.5%, S<0.2%, Mn<1.5%), cutting tool is with 75° entering angle, 6° positive rake angle, and Coromant grade GC315 (K15).

Net power requirement in dependence of feed rate, cutting speed and depth of cut is done in table 1.

Table 1. Net power requirement

f [mm/rev]	v_c [m/min]	P_c [kW]			
		a_p [mm]			
		1	2	3	4
0.1	275	1.0	1.5	2.5	3.0
0.2	220	1.0	2.0	3.0	4.0
0.3	190	1.0	2.0	3.5	4.5
0.4	165	1.0	2.5	3.5	4.5
0.5	150	1.5	2.5	4.0	5.0
0.6	135	1.5	2.5	4.0	5.5
0.8	110	1.5	2.5	4.0	5.0
1.0	90	1.5	2.5	4.0	5.0
1.2	80	1.5	2.5	4.0	5.0

Considering important steps in NN development and important training and architectural parameters of MLP BP network, optimal cutting power model was developed.

A three layer MLP network with three input neurons representing cutting speed, feed rate and depth of cut, one hidden layer and one neuron in the output representing cutting power was designed and trained with the BP training algorithm.

The upper limit of number of hidden neurons was determined knowing that the number of weight doesn't exceed the number of training samples. It is easy to calculate that for three inputs, the maximum allowed number of hidden neurons is 7 for 28 training data.

Various network architectures were developed. The hyperbolic tangent sigmoid transfer function is used in the hidden layer and linear transfer function in the output layer. The following training parameters were used: learning coefficient 0.01, momentum 0.9, initial weights uniformly sampled from the range ±0.5 and the number of epochs was determined using trial and error approach.

The data was randomly divided in two sets: training set with 28 samples, and testing set with 8 samples. Since hyperbolic tangent transfer function was used in hidden layer the data were scaled to [-1,1] range. In order to determine the best network structure of the ANN prediction models, the combination of RMSE, MAPE and R was used. The 3-6-1 network architecture was found to be optimal. The performance measures on trying and testing are given in table 2.

Table 2. NN model performances

	TEST	TRAINING
R	0.994	0.998
MAPE	5.420	3.332
RMSE	0.372	0.176

Figure 6 presents the graph that shows the comparisons of NN model prediction and target data.

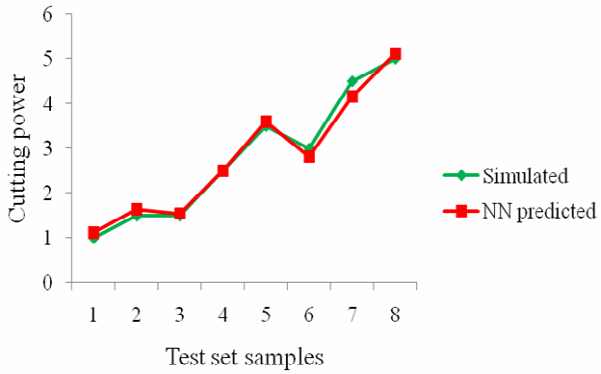

Fig. 6. Comparison of NN predicted and target data

As it can be seen from the figure, the NN prediction model provided high accuracy in predicting cutting power for test samples.

6. CONCLUSIONS

This paper has discussed neural network based modeling of machining processes. Basic ideas and concept of applying neural networks for modeling of machining processes have been given. Modeling with neural networks is not a trivial task, since that there are no formal theories for determining, a priori, the optimal network model. In this paper the methodology of modeling with MLP BP neural network was presented to assist current and potential users. Methodology of neural network modeling comprised of two integral parts: model developing and selection of training parameters. We discussed six crucial steps of model building phase and presented some guidelines for determining the training parameters. These issues are of crucial importance for developing neural network based prediction models. The methodology of neural network modeling was successfully applied for modeling of cutting power.

7. REFERENCES

1. Basheer, A., Hajmeer, M. (2000). *Artificial neural networks: fundamentals, computing, design, and application*. Journal of Microbiological Methods, 43, 1, ISSN 0167-7012, pp. 3-31

2. Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*, Prentice Hall, ISBN-13 978-0132733502, New Jersey.

3. Karayel, D. (2009). *Prediction and control of surface roughness in CNC lathe using artificial neural network*. Journal of Materials Processing Technology, 209, 7, ISSN 0924-0136, pp. 3125-3137

4. Madic, M., Radovanovic M. (2009). *Artificial neural networks and their application in manufacturing processes*, Journal "IMK-14, research and development", year XV, No 3-4, ISSN 0354-6829, pp. 39-44

5. Madic, M., Radovanovic, M. (2009). *Application of Artificial Neural Networks in Manufacturing Technologies*, International Scientific Conference UNITECH'09, ISSN 1313-230X, pp. II.593-II.599, Gabrovo, Bulgaria.

6. Madic, M., Radovanovic, M., Lazarevic, D. (2009). *Artificial neural networks in non-conventional machining processes*, Second International Conference on Diagnosis and Prediction in Mechanical Engineering Systems (DIPRE'09), CD-ROM, ISBN 978-606-8008-33-2, Galati, Romania.

7. Mukherjee, I., Ray, P.K. (2006). *A review of optimization techniques in metal cutting processes*. Computers & Industrial Engineering, 50, 1, ISSN 0360-8352, pp. 15-34

8. Radovanovic, M., Madic, M., Jankovic, P. (2008). *Artificial Neural Network Modeling of Cutting Force Components by Turning*, International Scientific Conference UNITECH'08, ISSN 1313-230X, pp. II-486-II-490, Gabrovo, Bulgaria.

9. Radovanovic, M., Madic, M., Jankovic, P. (2008). Application of Neural Networks in Metal Cutting, *8th International Conference "Research and Development in Mechanical Industry"- RaDMI 2008*, pp. 322-328, ISBN 978-86-83803-24-3, Užice, Serbia.

10. Radovanovic, M., Madic, M., Jankovic, P. (2008). *Comparison of Regression Model and Artificial Neural Network Model for Predicting the Main Cutting Force by Turning*, Buletinul Institutului Politechnic din Iasi, tom LIV, fac.2, ISSN 1011-2855, pp. 95-104, , Iasi, Romania

11. Tarassenko, L. (1998). *A Guide to Neural Computing Applications*, Arnold Publishers, ISBN 0 340 70589 2, London.

12. Zhang, G., Patuwo, B.E. & Hu, M.Y. (1998). *Forecasting with artificial neural networks: the state of the art*. International Journal of Forecasting, 14, 1, ISSN 0169-2070, pp. 35-62

13. Zhang-Xue, F., Zhi-Guang, Y., Kusiak, A. (2006), *Selection and validation of predictive regression and neural network models based on designed experiments*, IIE Transactions, 38, ISSN 0740-817X, pp. 13-23

14. Van Luttervelt, C.A. (2001). *Typology of models and simulation methods for machining operations*. Machining Science and Technology, 5, 3, ISSN 1091-0344, pp. 415-428

Received: 25 May 2010 / Accepted: November 20, 2010 / Abstract published online: December 5, 2010