

## DEVELOPMENT OF AN INVERSE MODEL FOR HONING PROCESSES BY MEANS OF NEURAL NETWORKS

**Maurici Sivatte-Adroer**

**Universitat Politècnica de Catalunya (UPC). Department of Mechanical Engineering.  
Polytechnical School of Vilanova i la Geltrú (EPSEVG). Av. Víctor Balaguer, 1, 08800  
Vilanova i la Geltrú (Barcelona), Spain**

**Xavier Llanas-Parra**

**Universitat Politècnica de Catalunya (UPC). Automatic Control Department.  
Polytechnical School of Vilanova i la Geltrú (EPSEVG).  
Av. Víctor Balaguer, 1, 08800 Vilanova i la Geltrú (Barcelona), Spain**

**Irene Buj-Corral, Joan Vivancos-Calvet**

**Universitat Politècnica de Catalunya (UPC). Department of Mechanical Engineering.  
Industrial School of Engineering of Barcelona (ETSEIB). Av. Diagonal 647, 08028,  
Barcelona, Spain**

### ABSTRACT

*In a previous paper, artificial neural networks were employed for modelling average roughness  $R_a$  in rough honing processes as a function of process variables, namely grain size, density, linear speed, tangential speed and pressure, by means of the direct problem. In addition, neural network model was compared to statistical models for modelling roughness. In the present paper the inverse problem was studied and analyzed by means of neural networks, in which given a certain average roughness  $R_a$  value, the model predicts process variables to be employed. This is not possible with statistical models. Two different approaches were considered: use of a single network or use of five networks.*

**Keywords:** roughness, neural networks, honing

### 1. INTRODUCTION

Abrasive machining processes are in general complex and not linear. In addition, many variables affect the process. For these reasons, different authors have employed neural networks for modeling grinding processes [1, 2, 3]. In addition, neural networks have shown to improve results obtained when modelling roughness in honing processes with respect to statistical models [4].

While the rest of studies in the literature for honing have addressed the direct process, in which surface roughness is predicted as a function of process variables (grain size GS, abrasive density DE, pressure PR, tangential speed VT, linear speed VL), in the present paper an inverse process is presented. In the inverse problem, given a certain roughness value, neural networks will determine process variables to be used in order to obtain it. This is an interesting approach, since it can be used as a support for decision taking when selecting process parameters to be used in industry.

### 2. EXPERIMENTAL METHOD

A total amount of 33 St-52 cylinders of length 100 mm were machined in a honing test machine. Roughness measurements were performed with a Taylor Hobson roughness meter. On each cylinder, 9 measurements were taken along a diametral circumference at 50 mm of both cylinder ends. Measurements were separated 40°. Extreme values were discarded according to the Chauvenet

criterion [5] and finally 207 data sets were considered. Obtained results were presented elsewhere when solving the direct problem [6].

### 3. DATA TREATMENT

In order to train the 207 data sets, they were randomly divided into: training sets (70% or 145 data sets), validation sets (15% or 31 data sets), and testing sets (15% or 31 data sets), according to the recommendations by Goyun [7] and Crowther [8]. In order to generalize results, the study was repeated three times, in a way that three random data divisions were performed, called a, b and c.

Training data are used to train each network in an iterative way. Validation data are used to stop training when decrease in mean quadratic error is lower than a certain previously defined value, in this case  $1 \cdot 10^{-10}$ . Test data are not used in the training process, but will be used for assessing and comparing the properties of the network with other models' properties.

### 4. CONFIGURATION OF THE NEURAL NETWORK

According to previous studies [2, 3, 4], recommended network is a multilayer perceptron with backpropagation learning algorithm, with either one or two hidden layers, with a non-linear activation function.

For one hidden layer, a number of neurons between 4 and 100 were tested, according to Lawrence and Peterson [9], with the configuration that is shown in Figure 1.

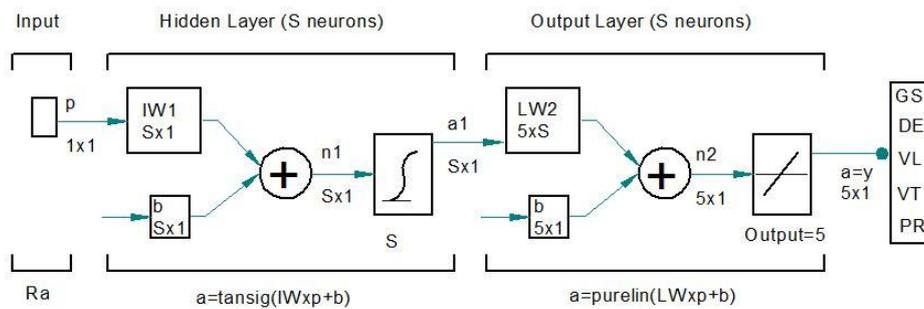


Figure 1. Configuration of the network with one hidden layer

For networks with two hidden layers (Figure 2), the optimal number of neurons found for one hidden layer was increased in 50 %. Resulting number was divided into two groups: 2/3 of neurons were used in the first layer and 1/3 of neurons were used in the second layer. From this starting point, also other combinations of neurons were tested.

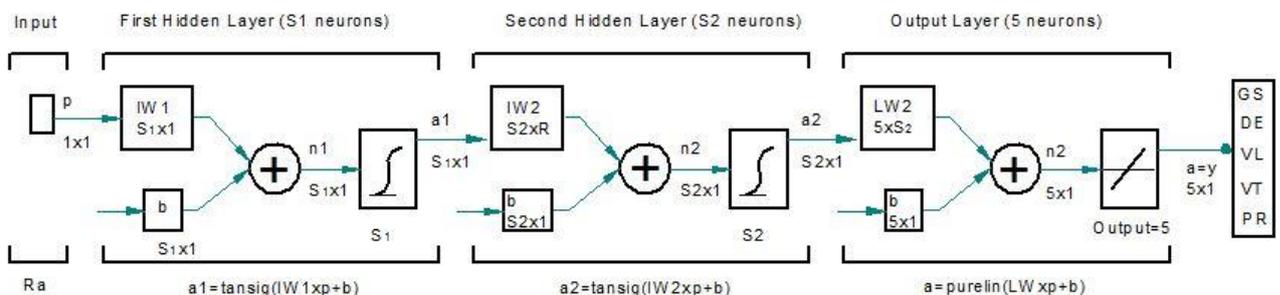


Figure 2. Configuration of the network with two hidden layers

Two different approaches are presented. In the first one a unique network has a single input (desired roughness Ra) and 5 outputs (GS, DE, VT, VL and PR) (Figure 3).

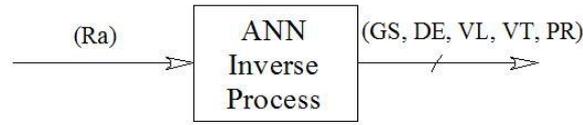


Figure 3. Functional diagram of model with one neural network

The second approach consists of using 5 networks simultaneously, with the same input Ra and one output for each variable, as shown in Figure 4.

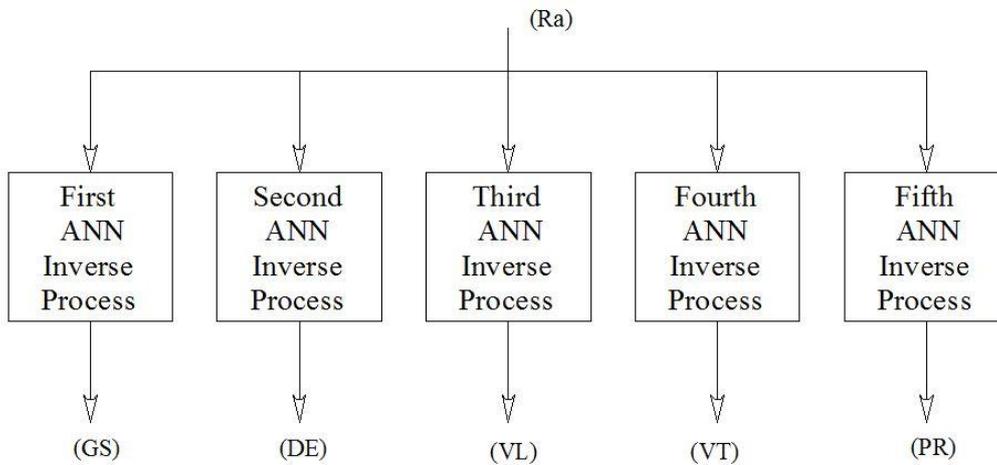


Figure 4. Functional diagram of model with five neural networks

## 5. METHODOLOGY FOR SELECTING THE MULTILAYER PERCEPTRON

For each data division a, b and c, and for each approach, the perceptron configuration that best fits the training and validation data was searched. Quadratic mean error (mqe) of the validation set was used as a selection parameter.

$$mqe = \sqrt{\sum_{i=1}^n \frac{(t_i - y_i)^2}{n}} \quad (1)$$

Where  $t_i$  = experimental Ra value of the  $i^{\text{th}}$  pattern,  
 $y_i$  = simulated Ra value of the  $i^{\text{th}}$  pattern, and  
 $n$  = number of patterns.

Different weights were assigned to optimal number of neurons of each data division, taking into account mqe of each data division, and a final optimal number of neurons was found.

## 6. RESULTS

### 6.1. Neural model with one network and one or two hidden layers

For one network with one hidden layer, best network has 70 neurons in the hidden layer. For one network with two hidden layers, best network has 71 and 36 neurons in the first and second hidden layer respectively.

### 6.2. Neural model with five networks and one or two hidden layers

For five networks with one hidden layer, best networks are the ones with 40 neurons for GS, 48 neurons for DE, 55 neurons for VL, 93 neurons for VT and 57 neurons for PR.

In a similar way, for five networks with two hidden layers, best results were obtained for: 43 and 21 neurons for GS, 51 and 27 neurons for DE, 53 and 27 neurons for VL, 84 and 40 neurons for VT, and 56 and 27 neurons for PR.

In order to decide if it is better to use one or two hidden layers in each case, a fourth data division  $d$  was employed. Networks were compared using mean quadratic error with validation data.

Final results are as follows: 40 neurons for GS, 51 and 27 neurons for DE, 53 and 27 neurons for VL, 84 and 40 neurons for VT, 57 neurons for PR.

### 6.3 Selected neural model

When comparing results for one neural network and five different neural networks, using a similar approach that takes into account mqe, a final neural network was selected with two hidden layers of 71 and 36 neurons respectively.

## 7. CONCLUSIONS

When addressing the inverse problem with neural networks in honing processes, which consists of determining process variables to be used in order to obtain a certain roughness value, two different approaches were used. In the first one only one network was used, with either one or two hidden layers. In the second one, five different networks were defined, one for each process variable. In this case, also one and two hidden layers were studied. Best results were obtained for one network with two hidden layers.

## 8. REFERENCES

- [1] Feng C.X., Wang X.F. (2002). Surface Roughness Predictive Modeling: Neural Networks versus Regression. *IIE Transactions on Design and Manufacturing*, 35(1): 11-27.
- [2] Liao T.W., Chen L.J. (1994). A Neural Networks approach for grinding processes: Modeling and optimization. *International Journal of Machine Tools and Manufacture*, 34(7): 919-937.
- [3] Fredj N.B., Amamou R., Rezgui M.A. (2002) Surface Roughness Prediction Based Upon Experimental Design and Neural Network Models. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 5: 129-134.
- [4] Feng C.X., Wang X., Yu Z. (2002). Neural Networks Modeling of Honing Surface Roughness Parameters Defined by ISO13565. *SME Journal of Manufacturing Systems*, 21(5), 2002.
- [5] Buj-Corral I., Vivancos-Calvet J. (2011). Roughness variability in the honing process of steel cylinders with CBN metal bonded tools. *Precision Engineering*, 35(2): 289-293.
- [6] Sivatte-Adroer M., Llanas-Parra X., Buj-Corral I., Vivancos-Calvet J. (2013). Selection of a Neural Network for Modeling the Honing Process. *Proceedings of the 17<sup>th</sup> International Research/Expert Conference "Trends in the Development of Machinery and Associated Technology" TMT 2013*, 1-4.
- [7] Guyon I. A scaling law for the validation-set training-set size ratio. *AT&T Bell Laboratories*. 1-11.
- [8] Crowther P; Cox R. (2005). A Method for Optimal Division of Data Sets for Use in Neural Networks. *Knowledge-Based Intelligent Information and Engineering Systems. Lecture Notes in Computer Science*, 3684:1-7.
- [9] Lawrence M., Petterson A.L. (1993). Introduction to Brain Maker neural networks. In: *Brain Maker user's guide and reference manual*. Nevada City: California Scientific Software Press. p. 1-22.