

Optimizing a dental milling process by means of Soft Computing Techniques

Vicente Vera, Alvaro Enrique Garcia
Maria Jesus Suarez, Beatriz Hernando
Departamento de Estomatología I & II
Facultad de Odontología, UCM
Madrid, Spain

vicentevera@odon.ucm.es; aegarcia@odon.ucm.es

Raquel Redondo
Department of Civil Engineering
University of Burgos
Burgos, Spain
redondo@ubu.es

Emilio Corchado, Maria Araceli Sanchez, Ana Gil
Departamento de Informática y Automática
Universidad de Salamanca
Salamanca, Spain
escorchado@usal.es

Javier Sedano
Dept. of A.I. & Applied Electronics
Castilla y León Technological Institute
Burgos, Spain
javier.sedano@itcl.es

Abstract—A novel soft computing system to optimize a dental milling process is proposed. The model is based on the initial application of several statistical and projection methods as Principal Component Analysis and Cooperative Maximum Likelihood Hebbian Learning to analyze the structure of the data set and to identify the most relevant variables. Finally, a supervised neural model and identification techniques are applied, in order to model the process and optimize it. In this study a real data set obtained by a dynamic machining center with five axes simultaneously is analyzed to empirically test the novel system in order to optimize the time error.

Keywords—Computational Intelligence; Soft computing Systems; Identification Systems; Artificial Neural Networks; Non-linear Systems; Milling dental process; Exploratory Projection Pursuit.

I. INTRODUCTION

Computer-assisted processing systems in dentistry for the creation of prosthodontic implants are one of the great advances brought about by information science and dental milling machines over recent years. These are an increasingly frequent alternative to conventional methods used in dental laboratories. This study details the way in which a soft computing model may be applied to optimize a step in Computer-Aided Design (CAD)/Computer-Aided Manufacturing (CAM), by optimizing the time error.

New dental milling machines permit the manufacture of individual crowns or complex bridges from data collected directly from the dental scanner.

Soft computing represents a set of various technologies involving non-linear dynamics, computational intelligence, ideas drawn from physics, physiology and several other computational frameworks. It investigates, simulates and analyzes very complex issues and phenomena in order to solve real-world problems: such as the failures detection in dental milling process, which requires a multidisciplinary approach.

A three-step procedure for testing and validating the model is proposed: firstly, the dataset is analyzed using projection methods as Principal Component Analysis (PCA)

and Cooperative Maximum-Likelihood Hebbian Learning (CMLHL) [1] to extract the dataset structure and the key relations between variables and to know if the data set is sufficiently informative. Finally, a model is produced at the modelling stage to estimate production time errors by modeling techniques.

This paper is organized as follows. Section II introduces the unsupervised projection techniques for analysing the datasets in order to extract their relevant internal structures. Section III deals with classical identification techniques used in the system modelling. Section IV describes the case study and Section V describes the nature inspired model presented to solve the case. Section VI presents the experimentation and results and finally, the conclusions are set out and some comments on future lines of work are outlined.

II. SOFT COMPUTING FOR FEATURE SELECTION

Soft Computing is a set of several technologies whose aim is to solve inexact and complex problems [2]. It investigates, simulates, and analyzes very complex issues and phenomena in order to solve real-world problems [3]. Soft Computing has been successfully applied in feature selection, and plenty of algorithms are reported in the literature [4][5], the Principal Component Analysis (PCA) among others [6]. In this study, an extension of a neural PCA version [7] and other extensions are used to select the most relevant input features in the data set as well as to study its internal structure.

A. Data structure analysis using connectionist techniques

1) Projection models.

The Negative Feedback Network [14] is defined as follows. Consider an N-dimensional input vector, (x), and a M-dimensional output vector, (y), with W_{ij} being the weight linking input j to output i and let η be the learning rate.

The initial situation is that there is no activation at all in the network. The input data is feedforward via the weights from the input neurons (the x-values) to the output neurons (the y-values) where a linear summation is performed to get the output neuron activation value. We can express this as:

$$y_i = \sum_{j=1}^N W_{ij} x_j, \forall i \quad (1)$$

The activation is feedbacked through the same weights and subtracted from the inputs (where the inhibition takes place):

$$e_j = x_j - \sum_{i=1}^M W_{ij} y_i, \forall j \quad (2)$$

After that, the simple Hebbian learning is performed between input and outputs:

$$\Delta W_{ij} = \eta e_j y_i \quad (3)$$

The effect of the negative feedback is the network learning stability. This network is capable of finding the principal components of the input data [14] in a manner that is equivalent to Oja's Subspace algorithm [15], and so the weights will not find the current Principal Components but a basis of the Subspace spanned by these components.

Maximum Likelihood Hebbian Learning [13][16][12] is based on the previous PCA-type rule and can be described as a family of learning rules based on the following equations: There is a Feedforward step (1) followed by a Feedback step (2) and then a weight change as follows:

$$\Delta W_{ij} = \eta \cdot y_i \cdot \text{sign}(e_j) |e_j|^{p-1} \quad (4)$$

It is expected that for leptokurtotic residuals (more kurtotic than a Gaussian distribution), values of $p < 2$ would be appropriate, while for platykurtotic residuals (less kurtotic than a Gaussian), values of $p > 2$ would be appropriate.

By maximising the likelihood of the residual with respect to the current distribution (tuning p parameter), the learning rule is matched to the probability density function of the residual. Maximum Likelihood Hebbian Learning (MLHL) [13][16][12] has been linked to the standard statistical method of Exploratory Projection Pursuit (EPP) [17][12].

2) Cooperative Artificial Neural Architecture

Cooperative Maximum-Likelihood Hebbian Learning (CMLHL) [1] is based on the standard Maximum Likelihood Network [13][16][12] but this time it has a lateral connection (which acts after the feedforward but before the feedback) derived from the Rectified Gaussian Distribution [18][1][19] and using the cooperative distribution. The final architecture is described as follows. There is a Feedforward step (1), then there is a lateral activation passing:

$$y_i(t+1) = [y_i(t) + \tau(b - Ay)]^+ \quad (5)$$

A Feedback step (2) and finally a weight updating (4) takes place. Where:

The parameter τ represents the strength of the lateral connections. The cooperative distribution in the case of N variables is defined by:

$$A_{ij} = \delta_{ij} + \frac{1}{N} - \frac{4}{N} \cos\left(\frac{2\pi}{N}(i-j)\right) \quad (6)$$

and

$$b_i = 1 \quad (7)$$

Where δ_{ij} is the Kronecker delta and i and j represent the identifiers of output neuron. Finally, $[\]^+$ is necessary to ensure that the y -values remain within the positive quadrant.

B. Feature selection and extraction

Feature Selection and extraction [8][9] entails feature construction, space dimensionality reduction, sparse representations and feature selection among others. They are all commonly used pre-processing tools in machine learning tasks, which include pattern recognition. Although researchers have grappled with such problems for many years, renewed interest has recently surfaced in feature extraction.

The feature selection approach in this study is based on the dimension reduction issue. Initially, some projection methods as PCA [10][11], MLHL [12][13] and CMLHL [1] are applied. In a first step they aim to analyse the internal structure of a representative data set of a case of study. If after applying these models a clear internal structure can be identified, this means that the data recorded is informative enough. Otherwise, data must be properly collected.

III. SYSTEM MODELLING USING IDENTIFICATION ALGORITHMS

System identification (SI) aims to obtain mathematic models to estimate one or more behaviours from a physical process whose dynamic equations are unknown. Classic SI refers to the parametrical literature, which has its origin in the linear system analysis [20]. Nevertheless, the increase of the computation capabilities and the availability of soft computing techniques have widen the researches in SI. One of the most interesting soft computing paradigms used in SI is the Artificial Neural Networks (ANNs). In any way, the SI procedure is invariantly the same taking advantage of the modelling technique used.

The SI procedure includes several steps [21][22]: the selection of the models and their structure, the learning methods [23][24], the identification and optimization criteria and the validation method. The validation ensures that the selected model meets the necessary conditions for estimation and prediction. Typically, validation is carried out using three different methods: the residual analysis $\varepsilon(t, \hat{\theta}(t))$ -by means of a correlation test between inputs, their residuals and their combinations-; the mean squared error (MSE) and the generalization error value -normalized sum of squared errors (NSSE)- and finally a graphical comparison between the desired outputs and the model outcomes through simulation.

The remaining of this section describes the use of ANN in SI.

A. The ANN in the identification process

The identification criterion consists in evaluating the best suite group of candidate models that best describes the dataset gathered for the experiment; i.e., given a certain model $M(\theta_*)$, its prediction error may be defined as in (8). The aim is to obtain a model that meets the following

premise [21]: a good model is one that makes good predictions and which produces small errors when the observed data is applied.

$$\varepsilon(t, \theta_*) = y(t) - \hat{y}(t | \theta_*) \quad (8)$$

The use of ANN in the process of identification requires the selection of several parameters: the number of layers, the number of neurons per layer and the activation functions [25][26]. It has been established that two feedforward layers using sigmoidal or hyperbolic functions in the hidden layer can learn any input-output relationship, nevertheless, more layers might learn complex relationships [27]. A feedforward network with two layers is shown in “Fig. 1”.

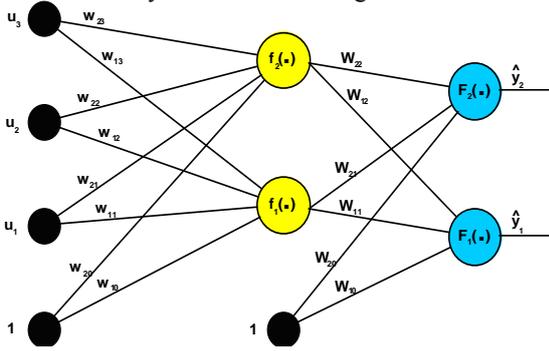


Figure 1. A feedforward network with two layers, with two nodes per layer, and three inputs. W is the weight matrix between the hidden and output layer, while w is the weight matrix between the inputs and the hidden layer. The network has two bias nodes with value 1.

The number of neurons per layer is also a relevant design parameter. It should be analyzed in order to avoid over fitting [28][29][30]. Each algorithm will introduce some restrictions in the weight matrix. The most widely used training algorithms are the Levenberg-Marquardt method [31], the recursive Gauss-Newton method [21], the quasi-Newton algorithm [32] and the batch and the recursive versions of the back-propagation algorithm [33].

When using ANN, the purpose of an identification process is to determine the weight matrix based on the observations Z^t , so as to obtain the relationships between the network nodes. The weight matrix is usually referred as w , W or θ .

The supervised learning algorithm is then applied to find the estimator θ , so as to obtain the identification criterion. In this case, the minimization of the mean square error criterion as defined in (9) and (10) is used. The iterative minimization scheme is defined in (11), where $f(t)$ represents the search direction and $\mu(t)$ the step size.

$$V_N(\theta, z^t) = \frac{1}{N} \sum_{t=1}^N [y(t) - \hat{y}(t | \theta)]^T [y(t) - \hat{y}(t | \theta)] \quad (9)$$

$$\hat{\theta} = \arg \min_{\theta} V_N(\theta, Z^t) \quad (10)$$

$$\theta(t+1) = \theta(t) + \mu(t)f(t) \quad (11)$$

IV. CASE STUDY

CAD/CAM in the field of dental applications may be structured into three steps: digitization (of a tooth stump in

the mouth); computer software design, specific to each system, allowing the design of the abutment of the prosthetic structure.

Finally, there is a mechanized process, in which milling instruments work on different materials such as ceramics, titanium, chromo-cobalt and compound resins. Therefore, at present, the latest dental milling machines manufacture individual crowns and complex bridge structures from the data collected by the dental scanner.

In this study, it is showed how Soft computing models can be applied to optimize the last step of a Computer Aid Design/Computer Aid Manufacturing (CAD/CAM) system, by optimizing the time errors detection for manufacturing metal dental pieces as showed in Fig.2.



Figure 2. Examples of metal pieces obtained by laser milling

The case study is described by an initial data set of 35 samples with 7 input variables (Number of pieces, Radius, Type of tool, Revolutions, Feed rate X, Y and Z) and 2 output variables (Erosion and Real time of work) as showed in “TABLE I”. Time errors for manufacturing are the difference between the estimated time by the machine itself and production real time.

Variable (Units)	Range of values
Number of pieces	1,2,4
Radius	0.4 to 1.5
Type of tool	Plane, toric, spherical and drill
Revolutions per minute	9600 to 24000
Feed rate X	0 to 4000
Feed rate Y	0 to 4000
Feed rate Z	50 to 3000
Erosion (mm.)	0 to 0.0452
Real time of work (s.)	39 to 1918
Time errors for manufacturing (s.)	0 to 550

TABLE I. VALUES OF EACH VARIABLE USED IN THE PROCESS.

V. A METHOD FOR OPTIMIZING A MILLING DENTAL PROCESS

A method is proposed for time errors detection for manufacturing dental pieces by means of comparing the properties evolution of the manufacturing process –time errors- in normal operation with respect to its estimated time behaviour. Firstly, the dental process of manufacturing is parameterised and its dynamic performance in normal operation is obtained by real manufacturing of dental pieces. Then, the gathered data is processed using PCA and CMLHL to recognize data set structures in order to determine the ability of the data set to be modeled and to identify the most relevant variables.

Once the model has been obtained –in the third step-, then it is used as a reference model to calculate in normal operation conditions the best conditions in dental milling

process for manufacturing dental pieces-, so if the operator wants to make a dental piece with a particular tool -toric, drill, etc.- he may determine the best machining conditions to minimize manufacturing time errors compared to the manufacturing estimated time which is given by the machine itself. The election of the operation conditions has been carried out manually; it is intended to include a new step to automatically include this task in the process.

This section deals with the description of each step once the data set is collected (see Section IV). In the next subsection the generation of the data set in the process is described. Sub-Section A presents the PCA and CMLHL step, while in Sub-Section B the procedure to obtain the time errors model is detailed.

A. Selection of the relevant features

As it was detailed in Section II, PCA and CMLHL, which were both applied to this real-life case study, are techniques for identifying the internal structure of a data set and also to identify the most relevant variables. Then, by means of projection methods we analyse if the data set is representative enough of a case study or not, and we identify the most relevant variables to reduce the computational cost in the third and last step.

B. Modelling the normal milling dental operation

Once the relevant variables and their transformations have been extracted from the production data, then a model to fit the normal manufacturing operation should be obtained in order to identify bias in the production estimated time, which is in the end used for time errors detection for dental pieces manufacturing. The different model learning methods used in this study were implemented in Matlab© [34].

The experiment followed the identification procedure detailed in Section III: the model structures were analyzed in order to obtain the models that best suited the dataset.

Moreover, several different indexes were used to validate the obtained models. The indexes are well-known and widely used measures in system identification [21][22]: the percentage representation of the estimated model; the graphical representation for the prediction $-\hat{y}_1(t|m)-$ versus the measured output $-y_1(t)-$; the loss function or error function (V) and the generalization error value.

The percentage representation of the estimated model is calculated as the normalized mean error for the prediction (FIT1). The loss function or error function (V) is the numeric value of the mean squared error (MSE) that is computed with the estimation data set. Finally, the generalization error value is the numeric value of the normalized sum of squared errors (NSSE) that is computed with the validation data set (NSSE1) and with the test data set (NSSE2).

VI. EXPERIMENTATION AND RESULTS

This initial data set has been analyzed in order to select the features that best described the relationships with manufacturing time errors [35].

CMLHL is a powerful technique for identifying internal dataset structures. It is applied to a data set, in order to select

the features that best describe the relationships between variables, and determine whether the dataset is sufficiently informative.

As it can be seen in "Fig. 3", PCA "Fig. 3.a" and CMLHL "Fig. 3.b" have found a clear internal structure in the dataset. Both methods have identified 'revolutions' as relevant variable. CMLHL projection gives us more information because it has recognized the 'number of pieces' as another important variable. CMLHL provides a sparser representation than PCA.

Having analyzed the results obtained with the CMLHL model, "Fig. 3.b", it is concluded that CMLHL has identified three different clusters ordered by 'number of pieces'. It is also concluded that CMLHL identified different clusters ordered by 'revolutions'. It has identified eleven clusters ordered by 'number of pieces' and 'revolutions'.

Inside each cluster there are further classifications by 'real time of work' and the dataset can be said to have an interesting internal structure.

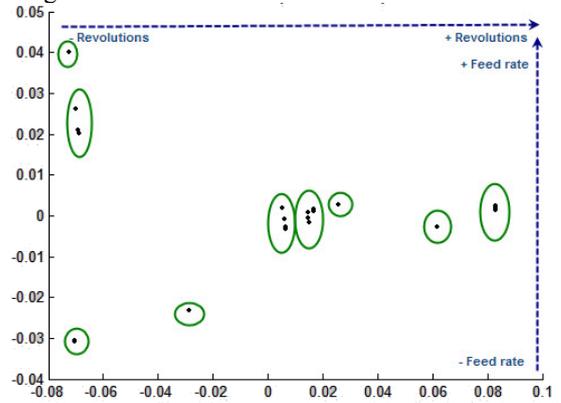


Figure 3 (a). PCA projections.

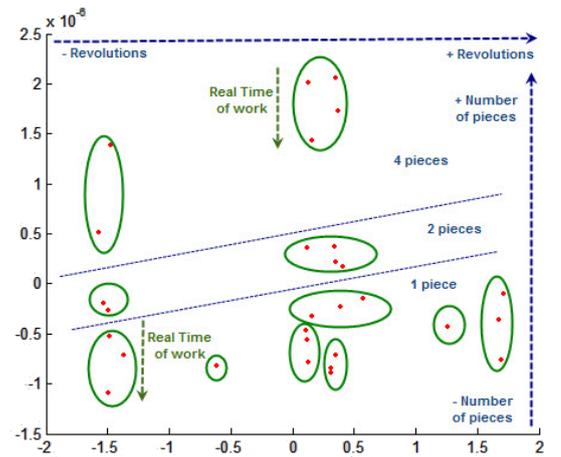


Figure 3 (b). CMLHL projections after 100000 iterations using a learning rate of 0.05, $p=0.5$ and $\tau=0.05$.

Figure 3. PCA projections (Fig. 3.a) and CMLHL projection (Fig. 3.b)

When the dataset is considered sufficiently informative, the step for modelling the relations between inputs and production time errors in the process begins, through the application of several conventional modelling ANN systems.

Thus, an ANN was used to monitor the time errors detection for dental pieces manufacturing. Using the preprocessed data set from the inputs and output normalizing step –zero mean and unity standard deviation-, the reduction of the input vectors dimension –data set gathered in the previous step- and the use of early stopping and Bayesian regularization techniques [30], strategy to generalize new situations. “TABLE II” shows the characteristics and qualities of estimation and prediction of the chosen ANN, along with their indexes.

The graphic representations of $\hat{y}_1(t|m)$ for time errors detection for manufacturing dental pieces manufacturing - $y_1(t)$ - is shown in “Fig. 4” and “Fig. 5” for a feedforward network structure. In “Fig. 4”, the x-axis shows the number of total samples and the y-axis represents the normalized output variable range: which are the normalized time errors for manufacturing. In “Fig. 5”, the y-axis represents the output unnormalized. For the technique of early stopping, the estimation, validation and test data sets include 23, 6 and 6 samples, respectively. For the technique of Bayesian regularization, the estimation data set includes 35 samples.

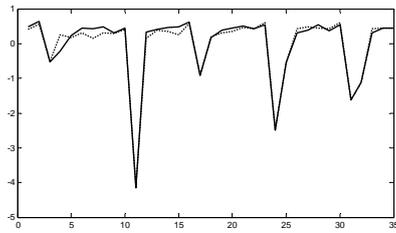


Figure 4 (a) Feedforward network with early stopping.

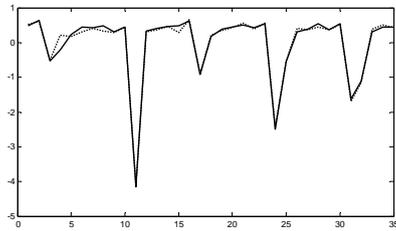


Figure 4 (b) Feedforward network with Bayesian regularization.

Figure 4. Normalized output response of the model: The current output (solid line) is graphically presented with prediction (dotted line). In Fig. 4.a is used the quasi-Newton algorithm for training, while in Fig. 4.b in training the Lenvenberg-Marquardt method has been used

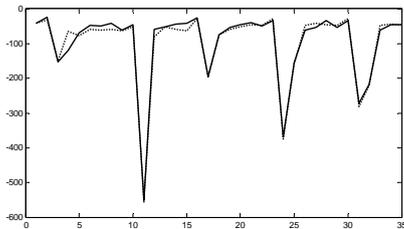


Figure 5 (a). Feedforward network with early stopping.

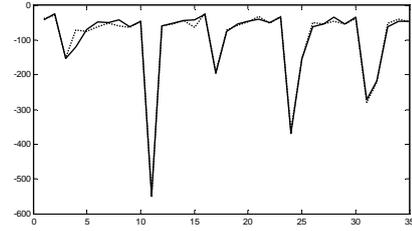


Figure 5 (b) Feedforward network with Bayesian regularization

Figure 5. Unnormalized output response of the model: The current output (solid line) is graphically presented with prediction (dotted line). In Fig. 5.a is used the quasi-Newton algorithm for training, while in Fig. 5.b in training the Lenvenberg-Marquardt method has been used.

From “Fig. 4”, it can be concluded that the feedforward networks are able to simulate and predict the behaviour of time errors for dental pieces manufacturing –as a consequence of the production process- and it is capable of modelling more than 89% of the actual measurements. In “TABLE II” the value of the indexes obtained for the models are shown. The model thus obtained can be used not only to predict time errors for dental pieces manufacturing but also to determine the normal operating conditions of dental milling process.

Model	Indexes
Feedforward network for the milling dental process. The model was obtained using the Bayesian regularized criterion. The ANN structure has 5 hyperbolic tangent units -layer 1-, 30 hidden hyperbolic tangent units -layer 2- and 1 linear output unit. The network is estimated using the Lenvenberg-Marquardt method.	FIT1:89.95% V: 0.0098
Feedforward network for the milling dental process. The model was obtained using the early stopping technique. The ANN structure has 5 hyperbolic tangent units -layer 1-, 30 hidden hyperbolic tangent units -layer 2- and 1 linear output unit. The network is estimated using the quasi-Newton method.	FIT1:87.98% V: 0.016 NSSE1:0.0083 NSSE2:0.009

TABLE II. THE VALUE OF THE QUALITY INDEXES OBTAINED FOR THE PROPOSED MODEL

VII. CONCLUSIONS AND FUTURE WORK

Dental milling process is a novel process for manufacturing dental pieces for the implementation in medical applications such as dental implants.

The methodology proposed in this study – for time errors detection for dental pieces manufacturing – in a first step aims to detect the ability of the data set to be modeled. In this step PCA and CMLHL have been the used techniques and they show that data are sufficiently informative and can be modeled in a next step with guarantees of success. Finally, different techniques -supervised neural model- are applied to obtain a suitable model, which will be responsible for detecting the time errors for dental pieces manufacturing.

Future work includes modelling the erosion -errors in length or tooth wear-, which is a measure of the quality of the dental milling process. Finally, an algorithm will be developed to automatically classify the different tools depending on the best operating conditions: minor time errors for dental pieces manufacturing and minor erosion.

FUNDING

This research has been partially supported through projects of the Junta of Castilla and León (JCyL): [BU006A08], TIN2010-21272-C02-01, CIT-020000-2009-12 and PID 560300-2009-11 from the Spanish Ministry of Science and Innovation, project of the Spanish Ministry of Education and Innovation [CIT-020000-2008-2] and project of Spanish Ministry of Science and Technology [TIN2008-06681-C06-04]. The authors would also like to thank ESTUDIO PREVIO (Madrid-Spain) for its collaboration in this research and also the vehicle interior manufacturer, Grupo Antolin Ingeniería, S.A., within the framework of project MAGNO2008 - 1028.- CENIT also funded by the same Government Ministry.

REFERENCES

- [1] E. Corchado and C. Fyfe. Connectionist Techniques for the Identification and Suppression of Interfering Underlying Factors. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 17(8), pp. 1447-1466, (2003).
- [2] L.A. Zadeh. Soft computing and fuzzy logic. *Integrated Computer-Aided Engineering*, 11(6):48–56, 1994
- [3] J.Sedano, E. de la Cal, L. Curiel, J.R. Villar and E. Corchado. Soft Computing for detecting thermal insulation failures in buildings. *Proceedings of the 9th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE2009*, Gijón, Spain, 2009.
- [4] J. Casillas, O. Cordón, F. Herrera, and P. Villar. A hybrid learning process for the knowledge base of a fuzzy rule-based system. In *Proceedings of the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU 2004*, pp. 2189–2196, Perugia, Italy, 2004.
- [5] T. W. S. Chow, P. Wang, and E. W. M. Ma. A new feature selection scheme using a data distribution factor for unsupervised nominal data. *IEEE Transactions on Systems, Man and Cybernetics - PART B: Cybernetics*, 38(2):499–509, 2008.
- [6] D.T. Lin and D.C. Pan. Integrating a mixed-feature model and multiclass support vector machine for facial expression recognition. *Integrated Computer-Aided Engineering*, 16(1):61–74, 2009.
- [7] C. Fyfe, R. Baddeley, and D.R. McGregor. *Exploratory Projection Pursuit: An Artificial Neural Network Approach*. Research report/94/160. University of Strathclyde, 1994.
- [8] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research, Special Issue on variable and Feature Selection*, 3:1157–1182, 2003.
- [9] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Knowledge and Data Engineering, IEEE Transactions*, 17(4):491–502, 2005.
- [10] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.
- [11] H. Hotelling. Analysis of a complex of statistical variables into Principal Components. *Journal of Education Psychology*, 24:417–444, 1933.
- [12] E. Corchado, D. MacDonald, and C. Fyfe. Maximum and Minimum Likelihood Hebbian Learning for Exploratory Projection Pursuit. *Data Mining and Knowledge Discovery*, 8(3):203–225, 2004.
- [13] C. Fyfe and E. Corchado. Maximum Likelihood Hebbian rules. In *Proceedings of the 10th European Symposium on Artificial Neural Networks (ESANN 2002)*, pp. 143–148, Bruges, Belgium, 2002. D-side Publishers.
- [14] C. Fyfe. Neural network for PCA and beyond. *Neural Processing Letters*, 6:33–41, 1996.
- [15] E. Oja. Neural networks, principal components and subspaces. *International Journal of Neural Systems*, 1:61–68, 1989.
- [16] E. Corchado and C. Fyfe. Orientation selection using Maximum Likelihood Hebbian Learning. *International Journal of Knowledge-Based Intelligent Engineering*, 7(2), 2003.
- [17] J.H. Freedman and J.W. Tukey. Projection Pursuit algorithm for Exploratory data-analysis. *IEEE Transactions on Computers*, 23(9):881–890, 1974.
- [18] E. Corchado, Y. Han, and C. Fyfe. Structuring global responses of local filters using lateral connections. *Journal of Experimental & Theoretical Artificial Intelligence*, 15(4):473–487, 2003.
- [19] H.S. Seung, N.D. Soccia, and D. Lee. The rectified gaussian distribution. *Advances in Neural Information Processing Systems*, 10:350–356, 1998.
- [20] J. Sedano, E. Corchado, L. C. Herrera et al., “The application of a two-step AI model to an automated pneumatic drilling process,” *International Journal of Computer Mathematics*, Vol. 86, No. 10-11, pp. 1769-1777, 2009.
- [21] L. Ljung. *System Identification. Theory for the User*. Prentice-Hall, Upper Saddle River, N.J., USA, 2nd edition.
- [22] M. Nørgaard, O. Ravn, N.K. Poulsen, and L.K. Hansen. *Neural Networks for Modelling and Control of Dynamic Systems*. Springer-Verlag, London U.K., 2000.
- [23] P. Stoica and T. Söderström. A useful parametrization for optimal experimental design. In *IEEE Trans. Automatic. Control*, AC-27, 1982.
- [24] X. He and H. Asada. A new method for identifying orders of input-output models for nonlinear dynamic systems. In *Proc. of the American Control Conf., S.F.*, pp. 2520–2523, California, 1993.
- [25] M.M. Gupta and D.H. Rao. On the principles of fuzzy neural networks. *Fuzzy Sets and Systems*, 61(10):1 – 18, January.
- [26] S.D. Bekiros and D.A. Georgoutsos. Evaluating direction-of-change forecasting: Neurofuzzy models versus neural networks. In *Mathematical and Computer Modelling*, Vol. 46, Italy. Eds. M. Marinaro & P.G. Morasso.
- [27] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.
- [28] L.K. Hansen and M.W. Pedersen. Controlled growth of cascade correlation nets. In *International Conference on Artificial Neural Networks*, pp. 797–800, Italy, 1994. Eds. M. Marinaro and P.G. Morasso.
- [29] B. Hassibi and D.G. Stork. Second order derivatives for network pruning: Optimal Brain Surgeon. In *Advances in neural Information Processing System 5*, pp. 164–171, San Mateo, CA, USA, 1993. Eds. S.J. Hanson et al.
- [30] D.J.C. Mackay. “Bayesian interpolation”, *Neural Computation*, Vol. 4, No. 3, 1992, pp. 415-447.
- [31] R. Fletcher. *Practical Methods of Optimization*. Wiley & Sons, Chichester, UK, 2nd edition, 1987.
- [32] J.E. Dennis and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Englewood Cliffs, NJ: Prentice-Hall, 1983.
- [33] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation*, Vol. 1. Addison-Wesley, Santa Fe Institute Studies In The Sciences Of Complexity, 1991.
- [34] H. Demuth, M. Beale and M. Hagan. *Neural Network Toolbox User’s Guide*, The Mathworks, Inc., 2010.
- [35] J. Sedano, L. Curiel, E. Corchado et al., “A soft computing method for detecting lifetime building thermal insulation failures,” *Integrated Computer-Aided Engineering*, Vol. 17, No. 2, pp. 103-115, 2010.