韓國精密工學會 '96年度 春季學術大會論文集 pp.403~413

Self-Organizing Neuro-tracking of Non-stationary Manufacturing Processes

Gi-Nam Wang and Young Cheol Go
Department of Industrial Engineering
Ajou University, Suwon 441-749, Korea
(Fax) 0331-214-0942 (Tel) 0331-219-2425
(Email Add: gnwang@madang.ajou.ac.kr)

Abstract. Two-phase self-organizing neuro-modeling (SONM), the global SONM and local SONM, is designed for tracking non-stationary manufacturing processes. Radial basis function (RBF) neural network is employed, and self-tuning estimator is also developed for the determination of RBF network parameters on-line. A pattern recognition approach is presented for identifying a correct RBF neural network, which is used for identifying current manufacturing processes. Experimental results showed that the proposed approach is suitable for tracking non-stationary processes.

1. Introduction

Computer-controlled manufacturing systems require process monitoring and control which could be performed effectively by an on-line tracking task. Tracking manufacturing processes are required to identify the model structure and parameters of the current underlying process. When the underlying process is changing dynamically , it would be a difficult task to keep track of the process model structure and parameters. Model identification might be more difficult task in case the underlying process is non-linear and non-stationary.

On-line process tracking has a widely applied in robot control, air force guidance, process control and adaptive prediction system. In quasi-stationary process, which is related to many sensor-based systems, tracking capability is evaluated by how well the model structure and parameters could be identified when they are changing piece by piece. Under this situation, process tracking scheme is directly applied to establish good prediction and control policy.

This study is to design an on-line neuro-tracking scheme which develops a self-organizing neuro-modeling (SONM). Two-phase SONM, the global SONM and local SONM, is presented for describing non-stationary processes. The global SONM is designed for identifying coarse model of the underlying process. A pattern recognition approach, which is based on an automatic feature generation and minimum distance classification, is developed. In case there is no existing appropriate network, a new radial basis function (RBF) neural network is created for modeling the current processes. Least squares and recursive least squares are used for the estimation of new RBF neural network weights. After completion of the global SONM, the local SONM is also designed for polishing the coarse model of the underlying process. The local SONM includes a self-tuning estimation of the RBF network parameters as well as on-line learning network weights.

The paper is organized as follows. Section 2 is a brief statement of the problem. That section describes the problem to be covered in this paper. Section 3 introduces the state-of-the art system identification and model classification by neural network. Especially, conventional techniques in literature of time series model identification and pattern recognition approach are also summarized. Section 4 discusses the basic principle of the proposed SONM. Also presented are the methods of on-line estimation of RBF network parameters and on-line learning paradigm with the detail operations of the global and local SONM. Section 5 presents experimental results of computer simulation that verify the proposed approach. In the final section, some discussion and concluding remarks are given.

2. Statement of the Problem

In the modeling and control of systems, it is quite common that a mathematical model of the plant or process is not initially available. The situation also arises where the underlying process is non-stationary and the corresponding model is not easily identified. The non-linear-non-stationary model is described as equation (1). It is assumed that the model structure is unknown. There is no specific information on model order and parameters.

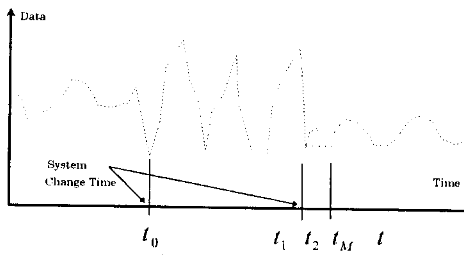
$$y_t = f_t(y_{t-1},\dots,y_{t-n_y}, u_{t-1},\dots,u_{t-n_u},\dots,e_{t-1},\dots,e_{t-n_e},\Theta_t)+e_t$$
(1)

where y_t is the measurement, u_t is the control input, e_t is the gaussian white-noise process with known variance σ_e^2, Θ_t is the piece-wise constant parameter vector, and t is the discrete time index. The parameter vector Θ_t and the function f_t are changing piece by piece through time t. Figure (1) represents the characteristic of non-stationary process which is described by the following Equation (2).

$$y_t = f_t(y_{t-1},\dots,y_{t-n_y}, u_{t-1},\dots,u_{t-n_u},\dots,e_t,\dots,e_{t-n_e},\Theta_t)+e_t$$
$$=$$
$$f_m(y_{t-1},\dots,y_{t-n_y}, u_{t-1},\dots,u_{t-n_u},\dots,e_{t-1},\dots,e_{t-n_e},\Theta_m)+e_t$$

$$\text{for } t_{m-1} \le t < t_m,\quad m=1,2,\dots M, M+1,\dots \quad (2)$$

The model structure f_t and model parameter Θ_t are changing piece by piece through time t. From time t_{m-1} to t_m, the underlying process is described by the non-linear-stationary model.



$f_1(y_{t-1},..u_{t-1},..,e_{t-1},..,\Theta_1) \mid f_2(y_{t-1},..,u_{t-1},..,e_{t-1},..,\Theta_2) \mid \dots \mid$
$f_M(y_{t-1},..,u_{t-1},..,e_{t-1},..,\Theta_M)$

Figure 1. General Description of Non-stationary Process

The main objective of this research is to develop an integrated methodology for on-line tracking non-stationary processes. The global and local modeling approaches are proposed for coarse model identification and fine model estimation respectively. The role of global modeling is to find the best probable model quickly. Once after a coarse model is found, the local modeling procedure is performed to polish the model structure and parameters. Their complementary relationship not only supports quick model classification but also gives an efficient on-line model estimation.

3. System Classification and Neuro-Modeling

A method of classifying stochastic systems is to categorize an unknown system to any one of the reference classes [Saridis and Hofstader 1974]. Pattern features used in the classification are extracted. They are compared with a set of stored reference features. The process of matching the unknown features to one of the references is carried out by clustering and similarity measure algorithm [Eichmann and Kasparis 1989]. Figure 2 shows a block diagram of stochastic system classification and estimation.

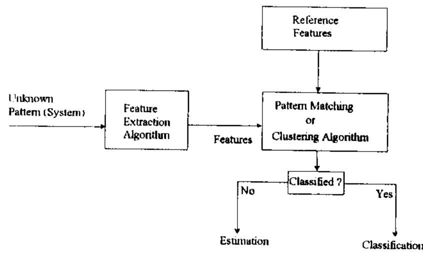


Figure 2. Block Diagram of Stochastic System Classification and Estimation

The task of stochastic system classification has difficulties in that stochastic systems could be modeled by different mappings even if their observations have similar patterns. For the different systems, the external measurements could have similar patterns while their internal mapping mechanisms are completely different. Therefore, the main concern of system classification is how to find a good feature extraction algorithm which could produce separable features for the different systems. Another important task is how to quickly estimate input-output mapping structure and parameters when the underlying process should be described by a new mapping. On-line identification of a new mapping relation would be necessary when the current system could not be described by the reference mappings.

Neural network can approximate unknown mappings [Hecht-Nielsen 1987, Kosko 1992]. Lots of works have been done using the universal function approximation of the mapping neural network. Multilayer neural networks are utilized for capturing stochastic mapping structure. However, capturing stochastic mapping structure. However, the multilayer neural network, based on sigmodal activation function, could not be easily applied for on-line system modeling since it required a lot of training computation.

Radial basis function networks (RBFNs) [Broomhead and Lowe 1988, Moody and Darken 1989] have been used to solve the problem of approximating stochastic system. Figure 3 represents a prototype design of RBF neural network.

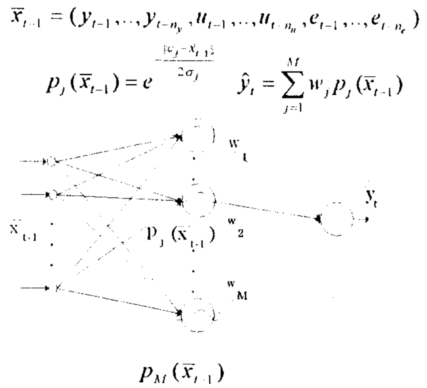


Figure 3. Radial Basis Function Neural Network

There is one hidden layer, and gaussian function is used as an activation function of each processing element. The linear combination of the activation function outputs is used for mapping the RBFN output. For the given model of equation (2), the RBFN is mathematically described as follows:

$$y_t = f_m(y_{t-1},\ldots,y_{t-n_y},u_{t-1},\ldots,u_{t-n_u},\ldots,e_{t-1},\ldots,e_{t-n_e},\Theta_m)+e_t \quad (3)$$

$$\hat{y}_t = \sum_{j=1}^{M} w_j p_j(\bar{x}_{t-1}) \quad (4)$$

for $t_{m-1} \le t < t_m$, $m=1, 2,\ldots, K, K+1,\ldots$

where $\bar{x}_{t-1} = (y_{t-1},\ldots,y_{t-n_y},u_{t-1},\ldots,u_{t-n_u},\ldots,e_{t-1},\ldots,e_{t-n_e})$ is an input vector at time t and $p_j(\bar{x}_{t-1}) = e^{-\frac{\|c_j-\bar{x}_{t-1}\|}{2\sigma_j}}$ is the activation function output of processing element j. The above RBF network is designed by assuming that the possible input data is clustered into M number of representatives. The parameter, c_j, is the centroid vector and σ_j is the width parameter of input cluster j respectively. The unit activation at j processing element, $p_j(\bar{x}_{t-1}) = e^{-\frac{\|c_j-x_{t-1}\|}{2\sigma_j}}$, is then a function of the distance between the current input vector $\bar{x}_{t-1}$ and the centroid vector c_j scaled by $2\sigma_j$. The main advantage of using RBF neural net is that weights can be adjusted using least squares and it's variants, which could be utilized in an on-line estimation scheme. Here the weight element w_j is chosen so that the sum of squares of the errors between the true system output and the predicted output is minimized as follows.

$$\text{Minimizing} \sum_{j=1}^{t}(y_j-\hat{y}_j)^2 \quad (5)$$

4. Self-Organizing Neuro-Modeling (SONM)

Two-phase SONM is presented for on-line tracking non-stationary processes: global and local SONM. The role of the global SONM is to identify a coarse model of a RBF network quickly, while the role of local SONM is to tune the coarse model which is identified by the global SONM.

4.1. Global SONM

The global SONM is focused on identifying approximate RBF neural network for an unknown inputs. Identification task includes the classifying of the best probable RBF network if the input can be modeled by the existing RBF networks as well as also including the estimation of a new RBF network, unless the current process cannot be modeled by the existing RBF networks.

Classification by Dynamic Feature Vector

Model identification by the global SONM is based on a pattern recognition approach. Pattern features used in the classification are extracted. Reference and unknown system features consist of the one-step prediction errors. The one-step prediction errors of unknown input are computed from the existing reference RBF network. The detail procedure of feature vector generation is described in Figure 4. For an unknown input $\bar{x}_{t-1} = (y_{t-1},\ldots,y_{t-n_y},u_{t-1},\ldots,u_{t-n_u},\ldots,e_{t-1},\ldots,e_{t-n_e})$

obtained by computing one-step ahead prediction errors for the reference RBF networks as follows.

$$e_t^1 = \hat{y}_t - y_t = RBF^1(\bar{x}_{t-1}) - y_t \tag{6}$$

The one-step ahead prediction error of the first RBF net, e_t^1, denotes the first element of unknown input feature vector, where $RBF^1(\bar{x}_{t-1})$ is the first RBF network prediction for the $\bar{x}_{t-1}$ input. The one-step prediction error e_t^1, is computed by subtracting the network output from actual observation.

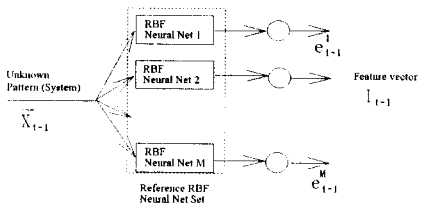


Figure 4. Feature Extraction Preprocessing of Unknown System

Similarly, the j reference feature elements could be obtained by computing the one-step ahead prediction errors of the existing RBF networks using the j cluster's representative C_j as input. The representative vector of each reference RBF net should be carried in order to compute a reference feature vector. The representative vector C_j consists of mean elements of the corresponding j clustered input, which is expressed as follows.

$$C_j = (C_j^{t-1}, C_j^{t-2}, \ldots, C_j^{t-n_y-n_u-n_e}) = (\bar{y}^j{}_{t-1}, \ldots, \bar{y}^j{}_{t-n_y}, \bar{u}^j{}_{t-1}, \ldots, \bar{u}^j{}_{t-n_u}, \ldots, \bar{e}^j{}_{t-1}, \ldots, \bar{e}^j{}_{t-n_e}) \tag{7}$$

The j reference feature vector expressed by $F_j = (f_j^1, f_j^2, \ldots f_j^M)$ consists of one step ahead prediction errors of each representative shown in Figure 5.

$$f_j^1 = \hat{C}_j^t - C_j^t = RBF^1(C_j) - C_j^t \cong RBF^1 C_j) - C_j^{t-1} \tag{8}$$

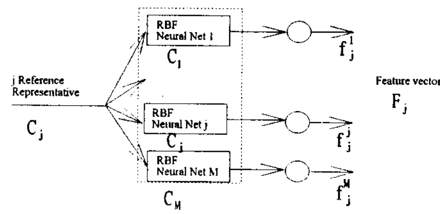


Figure 5. Feature Extraction Preprocessing of j Reference Model

The f_j^1 denotes the first element of j reference feature vector and $RBF^1(C_j)$ is the first RBF network prediction for the C_j input. In order to compute one-step prediction error f_j^1, the network output $\hat{C}_j^t$ is subtracted by C_j^{t-1}. From equation (7), the C_j^{t-1} represents the mean of the latest observation among the representative elements. One centroid element, which indicates the latest observation mean is, C_j^{t-1}, simultaneously used as an input element as well as an actual output in computing one-step prediction error. Since the theoretical derivation of actual output C_j^t is difficult, an approximation is used in this study.

The classification algorithm is based on the minimum distance classification. Euclidean distances between the new input vector and the reference feature vectors are calculated, and the nearest reference feature j^* is selected as a classification candidate satisfying the following condition.

$$\underset{j}{Min} \sum_{k=1}^{M} (f_j^k - e_t^k)^2 \qquad \text{for } j = 1, 2, \ldots M$$

If this minimum distance is not greater than a prespecified classification parameter, ρ_G, then classification can be performed by selecting the nearest reference feature to the unknown input feature. Classification decision cannot be made when the nearest reference feature is not enough close to the unknown input feature. In this case the current process cannot be modeled by one of the existing RBF networks, and a new reference RBF network should be created. When a new RBF network is created, all the existing reference features are modified. The modification is performed by adding the one-step prediction error of the newly created RBF network to the existing feature elements described in the equation (9).

$$F_j = (f_j, f_j^{M+1}) = (f_j^1, f_j^2, \ldots f_j^M, f_j^{M+1})$$
$$j = 1, 2, \ldots M+1 \qquad (9)$$

where $f_j^{M+1} = RBF^{M+1}(C_j) - C_j^{t-1}$ which is the one-step ahead prediction of the new RBF network.

The creation of a new RBF network is based on the ART 2 [Carpenter and Grossberg 1986], and it requires to determine the total number of network processing element, processing element parameters, and connection weights.

4.2. Local SONM

The objective of the local SONM is to polish the coarse model which is identified by the global SONM. The RBF network parameters include the centroid and width parameters of each cluster as well as the weight connections. The RBF neural network can have an on-line recursive estimation of weight connections (e.g. the least-squares method with extensions). Conventionally, the centroid parameter c_j and the width parameter σ_j are obtained by an off-line data analysis using the pre-determined data set, which is not easy to estimate these parameters on-line [Broomhead and Lowe 1988, Moody and Darken 1989]. These parameters give a great influence on system modeling accuracy [Hoskins et al. 1994]. Therefore, a drawback of RBF neural network is that the centroid and width parameters cannot easily be determined by an on-line estimation. The present paper establishes an on-line recursive estimation of c_j and σ_j which is necessary when the parameters are continuously changing.

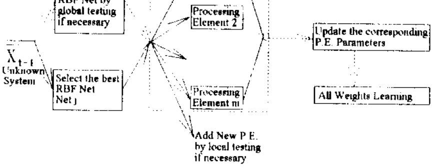


Figure 6. Block Diagram of Global & Local Self-Organizing Neuro-Modeling

Figure 6 shows the overall block diagram of the global and local SONM. The vigilance testing of the ART 2 explores the input vector, and then determines whether the current input is a new cluster or not. Based on this decision, the proper RBF neural network is selected or created. The next procedures are based on polishing the RBF net by the local SONM. Polishing task is performed by a local vigilance testing of the local ART 2 model, and then also determines whether the selected processing element's parameters should be updated or a new processing element should be added. If the input is considered as a new cluster, the new centroid and width parameters should be estimated with creation of new processing element: the new centroid vector is the first input. The centroid parameter, c_j, is just the mean (centroid) vector of the same class. The width parameter, σ_j, is the deviation from the center to the all same class inputs. In order to design an on-line system, following recursive calculation could be proposed.

$$c_j(t_j+1) = \frac{1}{t_j+1}[\bar{x}(t_j+1) + tc_j(t_j)] \quad (10)$$

$$\sigma_j(t_j+1) = \frac{1}{t_j+1}[t_j\sigma_j(t_j) + \|\bar{x}(t_j+1) - c(t_j+1)\|] \quad (11)$$

t_j=0,1, ..., where $c_j(0) = \{\}$, t_j denotes the total number of classes at j cluster, and $\|\bar{x}(t_j+1) - c(t_j+1)\|$ is the Euclidean distance between the input and centroid vector. $\bar{x}(t_j)$ is the input under the assumption that the input is classified as j cluster and the corresponding clustered total number is t_j. This recursive method eliminates the need to carry all previous clustered classes, and so it is very useful for on-line applications. In particular, the architecture of RBF neural network can be properly self-organized according to the incoming input space.

Once after the estimation of the centroid and width parameters, on-line learning is also necessary for updating the weights of RBF neural network. The weights learning by the recursive least squares estimation and unsupervised clustering by ART 2 are presented in Appendix.

The local SONM is performed right after the completion of the global SONM. Once a RBF network is classified or created, the local SONM accomplishes in tuning the selected RBF network parameters to the current input. Therefore, the global SONM can be interpreted as a "coarse modeling" while the local SONM is interpreted as a "fine modeling", which have a complementary relationships.

Related works are presented by Kil and Choi [1944] and Wang [1995]. Kil and Choi proposed a global estimation in which the parameters of mapping network are trained before the prediction of time-series while the local estimation model represents the on-line error compensation in which the parameters of error compensation network are adjusted in real time. However, their global estimation is based on an off-line estimation. The application is limited to the specific stationary time series. The k-phase RBF modeling presented by Wang [1995] is different from that of Kil and Choi network in that the k-phase RBF modeling relies on multi-phase local modeling of nonstationary processes while Kil and Choi net is based on single-phase modeling of stationary process. The local modeling procedure [Wang 1995] is based on the iterative re-modeling process using the one-step prediction errors. As a global modeling, the first-phase RBF net is designed. Also designed are the second, the third, and the K^{th} phase RBF nets as a local modeling. The present study is different from the previous works such that a pattern recognition approach is used for global modeling, which is suitable for application of wide input space. Another difference is that a complete on-line system is designed for global modeling and local modeling, which a recursive on-line scheme is proposed for updating the parameters of RBF neural network.

5. Experimental Results

A computer simulation is presented for the validation of the proposed methodology. This implementation is to investigate how well the global & local SONM keep track of non-stationary processes. Three different time series are used for implementation. The first series shown in the Table 1-1 is based on a synthetically generated linear series which is based on the Ph.D. dissertation [Wang 1993]. The second series, which is a non-linear step-wise changing series, is generated synthetically based on the paper Chen *et al.*, [1992]. The third series are real manufacturing signals which are collected from a rolling machine through a vibration sensor. The real manufacturing data is represented by a non-stationary process, which is a sequence of the vibration magnitude. In order to obtain the real data, the vibration sensor are attached to the rolling machine and analog signals are converted to digital signals by $1KHZ$ sampling rate. The prototype machine is shown in Figure 7, which is manufactured by ASeCon Lab. (Automatic Sensoring & Control Laboratory) of Industrial Engineering Department of Ajou University in Korea. According to machine conditions , six different signals are observed: normal condition, mal-functioning condition of outer race, inner race, ball, looseness, and unbalance.

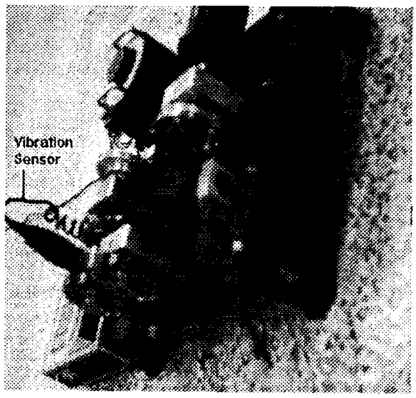


Figure 6. The Prototype Rolling Machine used in Experimental Analysis

The global SONM capability is evaluated by how correctly the underlying process is classified (referred to classification ratio) when it should be, and how well a new RBF network is created (referred to estimation ratio) when necessary. The local SONM performance is evaluated by modeling accuracy. One-step ahead prediction errors are analyzed for an evaluation of modeling accuracy.

From Table 1-2, the model parameters and structure of the second series is presented, and here u_t denotes uniform distribution from 0 to 1. Each input consists of 15 consecutive observations. Moving block of fixed width through 15 observations is used for the design of input. For a given model, 300 inputs which is equvalent to 4500 observations, are used for simulation. It is not considered for the case where two different models share one input vector. The existing segmentation algorithms [Wang 1993, Appel and Brandt 1983] could be used when one input block consist of two different models.

The first experiment is designed for evaluating the local SONM performance which is expressed by the performance of one-step ahead prediction errors. For a given same series, the first 50 numbers of inputs are used for determining initial total number of clusters, their corresponding parameters, and RBF network weights. The next 250 inputs are applied for tuning the number of cluster, parameters, and network weights. The total number of clusters (e.g. 15), which are obtained by considering the next 250 inputs, is increased compared to the number of total clusters (e.g. 5) obtained by considering the initial 50 inputs.

The initial centroid and width parameters are also updated according to the next iterative inputs. The recursive least squares estimation is utilized for learning the network weights. From Table 2-1 to Table 2-3, overall modeling accuracy of the SONM is compared using existing techniques. Mean absolute deviation (MAD) is employed as a modeling performance measure. The mathematical expression of MAD can be described as

$$MAD = \frac{\sum_{t=1}^{N} |e_t|}{N}$$

where e_t denotes a one-step ahead prediction error of a given technique at time t and N is the total number of evaluation.

Three different techniques are compared with the SONM. The Model MAD shown in Table 2-1 denotes the mean absolute deviation of one-step ahead prediction errors of the correct model structure and parameters. The number of total evaluation pattern N, used in MAD calculation for all simulation, is 250. The RBF_i means the RBF neural network which the parameters (e.g. the total number of clusters, centroid and width parameters, and weight connections) are estimated using the first 50 inputs. For the next 250 inputs, the parameters of RBF_i are not updated, and only the weights are adapted to the iterative coming inputs. On the other hand, the proposed SONM continuously updates the RBF parameters as well as the network weights. The RBF MAD in Table 2-1 represents mean absolute deviation (MAD) of one-step ahead prediction errors is the parameters are estimated by off-line data analysis using the whole 300 inputs obtained by the radial basis function neural network. Obviously, this approach is not based on an on-line application since the parameters are predetermined, assuming that the incoming observations is collected.

The simulation result shows that correct model has the best performance for modeling the linear series while the SONM is the best for modeling the non-linear and real series. This implies the SONM has powerful performance for modeling complicated series rather than doing simple linear series. The SONM MAD is almost equivalent or better than that of RBF MAD, which means that the SOMN's on-line modeling capability is excellent.

The classification ratio quantifies how the global SONM could classify the input patterns correctly. It is defined as the ratio of the number of times a pattern is correctly identified. The estimation ratio also quantifies how correctly global SONM creates a new RBF network if the creation is necessary. During the evaluation of estimation ratio, the new RBF network is not created and the three initial RBF networks are used for evaluating creation ratio for the different remaining process.

Three initial reference features are obtained using the three predetermined RBF neural network. Initially, three RBF neural networks are constructed using the corresponding 50 inputs, and the five internal processing element of the RBF network is computed. This implies that 50 input pattern has 5 initial clusters respectively. From the point of global SONM, three RBF networks are used for classification. The reference feature vector consists of three elements: one-step ahead prediction error of the first initial RBF network, one-step ahead prediction error of the second initial RBF network, and one-step ahead prediction error of the third initial RBF network. Each representative input vectors are utilized for computing each feature vectors. Therefore, each feature vector is characterized by the one-step ahead prediction errors of each representative mean vector.

The fourth RBF network is created newly when the Euclidean distance between the input and the nearest feature vector is larger than a prespecified classification parameter, ρ_G=12. During the simulation test, classification performance was excellent, which was above 94%, for the stochastic system classification. However, the estimation ratio was lower than classification ratio. The right creation of RBF network was greatly dependent on the prespecified classification parameter, ρ_G, which also gave great influence on global SONM. Empirical simulation was performed for computing the classification parameter. Determination of optimal classification parameter, ρ_G would be another research task.

Our focus is given on real application, which is summarized in Table 2.3. The average classification ratio is greater than 95.5%. This means that the machine condition is correctly classified with a reliable classification ratio. The global SONM could be utilized as a technique for machine condition monitoring, which is an identification scheme of machine faults. The on-line modeling accuracy of the local SONM shows also the best modeling performance. This implies that the proposed SONM could be applied for on-line modeling of nonstationary manufacturing processes.

6. Concluding Remarks

An on-line learning neural network scheme, the global SONM and local SONM, is proposed for modeling non-stationary series. The proposed SONM not only has self-tuning estimator of the parameters of RBF neural network but also has automatic identification of RBF network. Those characteristics play an important role in tracking non-stationary manufacturing processes, which have not been studied before. Identification of dynamic machine conditions would be another good application which could be used for machine faults diagnosis. The integration of pattern recognition approach to the self-organizing neuro-modeling would be a novel design for on-line tracking non-stationary processes. Unsupervised feature vector generation is a new approach for an efficient classification of stochastic time series.

As a further research, segmentation algorithm could be considered for the case when the input vector has two different series. Optimal determination of classification parameter would be another problem.

Appendix

a) Recursive Least squares Estimation (Refer to the Figure 3)

$$\hat{y}_t = \sum_{j=1}^{M} w_j p_j(\overline{x}_{t-1})$$

$$= [w_1, w_2, \ldots, w_M]^T$$

$$\overline{x}_{t-1} = (y_{t-1}, \ldots, y_{t-n_y}, u_{t-1}, \ldots, u_{t-n_u}, e_{t-1}, \ldots, e_{t-n_e})$$

$$\varphi(t) = [p_1(\overline{x}_{t-1}), p_2(\overline{x}_{t-1}), \ldots, p_M(\overline{x}_{t-1})]^T$$

$$(t) = W(t-1) + Q(t)\varphi(t+1)[y_t - \hat{y}_t]$$

$$Q(t) = Q(t-1) - \frac{Q(t-1)\varphi(t)\varphi(t)^T Q(t-1)}{[1+\varphi(t)^T Q(t-1)\varphi(t)]}$$

b) Adaptive Resonance Theory (ART 2)

References

1. U. Appel and A. V. Brandt, Adaptive, Sequential Segmentation of Piecewise Stationary Time Series, *Information Science* **29**, 27-56 (1983).
2. K. J. Astrom and B. Wittenmark, *Computer-Controlled Systems Theory and Design*, Englewood-Cliffs, Prentice Hall, New Jersey (1984).
3. M. Basseville and A. Benveniste, Sequential Segmentation of Non-stationary Digital Signals Using Spectral Analysis, *Information Science* **29**, 27-56 (1983).
4. S. A. Billings, H. B. Jamaluddin, and S. Chen, Properties of Neural Networks with Applications to Modeling Non-linear Dynamical Systems, *International Journal of Control* **55(1)**, 193-224 (1992).
5. G. E. P. Box, and G. M. Jenkins, *Time Series Analysis Forecasting and Control*, Holden-Day Inc., San Francisco, (1976).
6. D. S. Broomhead and D. Lowe, Multivariate Functional Interpolation and Adaptive Network, *Complex Systems* **2**, 321-355 (1988).
7. G. A. Carpenter and S. Grossberg, Neural Dynamics of Category Learning and Recognition: Attention, Memory Consolidation, and Amnesia, in J. Davis, R. Newburgh, and E. Wegman (Eds.) Brain Structure, Learning, and Memory, *AAAS Symposium Series*, (1986).
8. C. F. N. Chen, S. A. Billings, and P. M. Grant A Parallel Recursive Prediction Error Algorithm for Training Layered Neural Networks, *International Journal of Control* **51** 1215-1228 (1990).
9. G. Cybenko, Approximation by Superposition's of a sigmoidal function. *Mathematics of control, signals and systems* **2**, 303-314 (1989).
8. G. Eichmann and T. Kasparis. Pattern classification using a linear associative memory. *Pattern Recognition*, **22**,733-740, (1989).
10. G. C. Goodwin and K. S. Shin, *Adaptive Filtering Prediction and Control*, Prentice-Hall, Englewood Cliffs, New Jersey, (1984).
11. G. Govind and P. A. Ramamoorthy, Multi-layered Neural Networks for Arbitrary Approximation: an Explanation and Simulations, *The First International Conference on Neural Networks in Engineering, University of Missouri-Rolla*, 589-601, (1991).

12. R. Hecht-Neilsen, Theory of the Back-Propagation Neural Network. *In Proceedings of the international joint conference on neural networks*, Washington, D.C (pp I:217-I:225). New York: IEEE Press, (1989).

13. R. Hecht-Nielsen, *Neuro-computing*, Addison-Wesley Publishing Company, New York, (1990).

14. J. C. Hoskins, P. Lee, and S. V. Chakravarthy, Polynomial modeling behavior in radial basis function network. *World Congress on Neural Networks*, **IV**, IV693-IV699, July 11-15, (1993).

15. T. Kohonen, *Self-Organization and Associative Memory*, Springer-Verlag, New York, 1984.S. Kolliass and D. Anastassiou, An Adaptive Least Squares Algorithm for the Efficient Training of Artificial Neural Networks, *IEEE Transactions on Circuits and Systems* **36(8)**, (1989).

16. B. Kosko, *Neural Networks and Fuzzy Systems*, Prentice Hall, New Jersey, (1992).

17. A. Lapedes and R. Farber, Nonlinear Signal Processing Using Neural Networks: Prediction and System Modelling, *Preprint LA-UR-87-2662*, (1987).

18. A. Lapedes and R. Farber, How Neural Nets Work, *Technical Report Los Almos National Laboratory*, . 331-346, January (1988).

19. R. P. Lippman, An Introduction to Computing with Neural Nets, *IEEE Transactions on Acoustics, Speech, and Signal Processing* 4-22 (1987).

20. L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, MA., (1983).

21. J. Moody and C. J. Darken, Fast Learning in Networks of Locally-Tuned Processing Units, *Neural Computation* **1** (1989) 281-294

22. K. S. Narendra and K. Parthasarathy, Identification and Control of Dynamical Systems using Neural Networks, *IEEE. Transactions on Neural Networks* **1** 4-27 (1990).

23. Y. Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, New York, (1989).

24. T. Sastri, and G. N. Wang, "Adaptive Model Identification Method," *Neural Parallel & Scientific Computations*, **1**, 117-140 (1993).

25. T. Sastri, J. English, and G. N. Wang, "Neural Networks for Time Series Model Identification & Change Detection," *Advances in Artificial Intelligence in Economics, Finance, and Management*, **1**, 55-68 (1993).

26. Gi-Nam Wang, An Adaptive Hybrid Neural Network Approach to Prediction of Nonstationary Processes, *Hybrid Intelligent System Applications*, Edited by Dr. Jay Liebowitz, IOS Press, (1995).

27. Gi-Nam Wang, *An Adaptive Hybrid Neural Network Approach for Learning Nonstationary Manufacturing Processes*, Ph.D Dissertation, Texas A&M University (1993)

28. G.N. Wang, Gwang-Sup Kim, and Y.S. Jung, Two-Phase Radial Basis Hybrid Neural Network for On-line Modeling of Machine Condition, *Journal of the Society of Plant Engineers Japan*, **6(3)**, 20-28 (1994)

Model	# of Inputs	E(y)	Model Parameters
AR(2)	300	10.0	(1.60, 1.49,-0.65)
AR(6)	300	10.4	(22.0,-0.77,-0.51,0.31, 0.11,-0.03,-0.21)
AR(2)	300	10.5	(27.5,-1.07,-0.52)
AR(4)	300	10.1	(63.2,-1.87,-1.78,-1.20,-0.37)
AR(5)	300	9.80	(0.96, 1.84,-0.89,-0.61, 0.87,-0.31)
AR(3)	300	10.3	(0.19, 2.40,-2.41, 1.88,-1.58, 0.90,-0.20)
AR(6)	300	10.5	(0.19, 2.40,-2.41, 1.88,-1.58, 0.90,-0.20)
AR(3)	300	15.0	(6.50, 0.92,-0.26,-0.26)
AR(4)	300	10.5	(6.4, 0.96,-0.41,-0.30, 0.14)
AR(5)	300	10.1	(21.0,-0.37,-0.32, 0.10, 0.02,-0.50)

AR(2): $y_t = \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2}$

Table 1-1. Sequences of Linear Stationary Input Series with Model Parameters

Model	# of Inputs	E(y)	Model Parameters (a,b,c,d,d_1,f,g,h,k)
NLX1	300	4.95	(0.75,0.38,-.03,0.37,0.06,0.12,0.06,-.73-.36)
NLX2	300	3.58	(0.54.0.58,-.03,0.35,0.04,0.22,0.10,-.13,-.26)
NLX3	300	6.79	(0.64,0.98,0.23,0.42,0.24,-.13,0.20,0.13,-.12)
NLX4	300	3.63	(0.25,0.68,0.03,0.34,0.56,0.23,0.42,0.73,0.36)
NLX5	300	4.31	(-.72,0.18,0.24,0.65,0.86,0.22,0.96,0.47,-.36)

NLX=$y_t = ay_{t-1} + bu_{t-1} + cy_{t-1}^2 + dy_{t-1}u_{t-1} + d_1u_{t-1}^2 + fy_{t-1}u_{t-1}^2 + gu_{t-1}^3 + he_{t-1} + ku_{t-1}e_{t-1} + e_t$

Table 1-2. Sequences of Nonlinear Stationary Input Series with Model Parameters

Machine Condition	# of Inputs	E(y)	Standard Deviation
Normal	300	0.021	0.0292
Outer Race	300	-0.018	0.8123
Inner Race	300	0.013	0.5837
Ball	300	-0.062	1.3994
Unbalanced	300	-0.017	0.0404
Looseness	300	-0.010	0.0975

Table 1-3. Sequences of Real Vibration Signals Series of Machine Condition

Model	Model MAD	RBF_i MAD	RBF MAD	SONM MAD	Classification Ratio	Estimation Ratio
AR(2)_a	0.635	0.921	0.623	0.688	99.5 %	Initial
AR(6)_a	1.345	1.624	1.023	1.121	100.0 %	Initial
AR(2)_b	0.694	1.269	0.712	0.701	99.9 %	Initial
AR(4)_a	0.785	1.180	0.823	0.792	100.0 %	97.0 %
AR(5)_a	0.903	1.410	0.945	0.989	100.0 %	94.0 %
AR(3)_a	0.724	1.121	0.873	0.715	99.2 %	96.1 %
AR(6)_b	1.765	2.012	1.843	1.623	100.0 %	95.3 %
AR(3)_b	0.899	0.817	0.383	0.356	98.9 %	95.2 %
AR(4)_b	0.936	1.216	0.995	0.834	99.9 %	94.3 %
AR(5)_b	1.320	1.702	1.001	1.212	100.0 %	95.2 %

Table 2-1. Comparison of Global & Local SONM Performance (Linear Series)

Model	Model MAD	RBF_i MAD	RBF MAD	SONM MAD	Classification Ratio	Estimation Ratio
NLX1	1.435	1.921	1.223	1.128	99.9 %	Initial
NLX2	1.625	2.224	1.231	1.115	99.8 %	Initial
NLX3	1.342	1.416	1.201	1.002	99.9 %	Initial
NLX4	1.519	1.534	1.216	1.121	99.7 %	98.5 %
NLX5	1.601	1.530	1.311	1.182	99.9 %	96.5 %

Table 2-2. Comparison of Global & Local SONM Performance (Nonlinear Series)

Model	Model MAD	RBF_i MAD	RBF MAD	SONM MAD	Classification Ratio	Estimation Ratio
Normal	0.153	0.323	0.141	0.185	96.2 %	Initial
Outer Race	1.309	1.432	0.533	0.534	94.1 %	Initial
Inner Race	1.019	1.234	0.419	0.343	97.2 %	Initial
Ball	1.655	1.643	0.674	0.665	95.4 %	95.4 %
Looseness	0.471	0.534	0.207	0.216	94.4 %	94.4 %
Unbalance	0.314	0.545	0.127	0.214	95.7 %	95.7 %

Table 2-3. Comparison of Global & Local SONM Performance (Actual Series)