Modeling of continuous digesters using adaptive RBF neural network models

Alex Alexandridis, Haralambos Sarimveis, George Bafas

Abstract— **This paper presents a method for developing adaptive nonlinear models for time varying continuous digesters, which are complex nonlinear reactors used in the pulp and paper industry. The models are based on the Radial Basis Function neural network architecture and the adaptive fuzzy means algorithm is used to adapt both the structure and the connecting weights of the network. The proposed approach shows good approximation capabilities in the cases where the dynamics of the system and the operating region change with time.**

Index Terms—**Adaptive Modeling, RBF Networks, Training algorithms, Continuous Digester.**

I. Introduction

RADIAL Basis Function (RBF) networks form a special architecture of neural networks that present important advantages compared to other neural network types, including simpler structure and faster learning algorithms [1-5]. Due to these advantages, RBF networks have been used extensively for modeling a great variety of systems. During the last decade, many new training algorithms have been proposed for selecting automatically the proper structure of RBF networks and improving the network performance. These include an algorithm based on orthogonal least squares [6]; constructive or pruning methods [7-10]; genetic algorithms [11]; and determination of the hidden node locations based on a fuzzy partition of the input space [12].

Unfortunately, all these algorithms deal with time invariant systems while there are only few publications concerning the issue of RBF model development for time varying systems. Among them, a framework of two stages for the recursive determination of the network structure and the connecting weights has been proposed [13], but as reported in the paper, the algorithm is not guaranteed to be stable, since large modeling errors may lead to a possible breakdown of the weight-updating algorithm. Another publication suggested a procedure that combines an online candidate regressor selection with the Givens QR recursive parameter estimator for adaptive supervised network training [14], but the algorithm may need to reinitiate the clustering procedure, in case the structure of the system changes.

In general, modeling time varying systems is a rather difficult task for all system identification methodologies, including neural networks. Another disadvantage of the neural network methods which is due to their limited ability to extrapolate, is that they cannot track successfully changes in the operating region. Obviously, the development of an adaptive modeling scheme that can cope with both changes in system dynamics and the operating region could be very beneficial for modeling and controlling time-varying systems.

Continuous digesters are complex chemical reactors used in the pulp and paper industry [15] and certainly belong to the class of time varying systems. This paper aims to the development of an adaptive model for a simulated continuous digester, which can compensate for changes in the dynamics of the system and the operating region. This is achieved by incorporating the RBF neural network architecture and utilizing an adaptive training scheme that has the ability to modify both the structure and the connecting weights of the neural network model. The training methodology is based on a fuzzy partition of the input space, so that the algorithm can select the hidden node centers among the centers of the corresponding multidimensional fuzzy sets. In a second step, the connecting weights between the hidden and the output layer are adapted using the RLS with exponential forgetting algorithm.

The rest of the paper is structured as follows: The next section describes the RBF network architecture briefly, including a general overview of RBF networks, their use for modeling dynamical systems, and the training algorithms they employ. Section III presents the adaptive fuzzy means algorithm for training RBF networks, and is followed by a case study, where the methodology is implemented in the identification of a continuous digester. Finally section V outlines the most important advantages and sets some directions for future work.

A. Alexandridis is with the National Technical University of Athens, School of Chemical Engineering, 9 Heroon Polytechniou str., 15780, Athens, Greece (e-mail: alexalex@chemeng.ntua.gr).

H. Sarimveis is with the National Technical University of Athens, School of Chemical Engineering, 9 Heroon Polytechniou str., 15780, Athens, Greece (phone: +30-210-7723237; fax:+30-210-7723138; e-mail: hsarimv@central.ntua.gr).

G. Bafas is with the National Technical University of Athens, School of Chemical Engineering, 9 Heroon Polytechniou str., 15780, Athens, Greece (e-mail: gbafas@chemeng.ntua.gr).

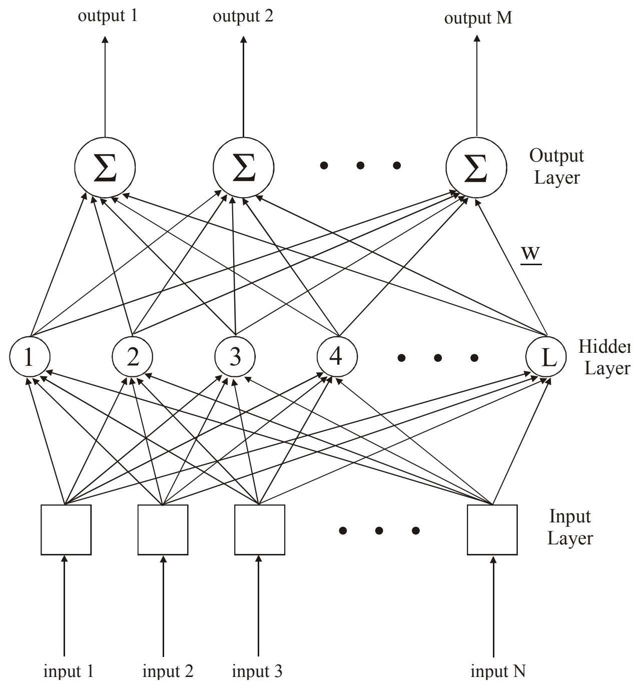


Fig. 1. Structure of an RBF network

II. Radial Basis Function neural networks

A. An overview of RBF Networks

Radial Basis Function networks form a special neural network architecture that consists of three layers (Fig. 1). The values of the input variables formulate an input vector, which is forwarded from the input layer to the hidden layer. The hidden layer is comprised of a number of nonlinear processing units (nodes), which are characterized by the center locations and the nonlinear radial basis function they employ. Each hidden node receives the input vector, calculates the Euclidean distance d between the center location and the input vector and finally performs a nonlinear transformation of the distance, using the radial basis function. The output of each hidden node is then multiplied by a particular synaptic weight w, while the final output of the network is a simple summation of all the weighted hidden node activations.

The most common selections for the radial basis function are:

- The Gaussian function:

$$f(d) = \exp\left(-\frac{d^2}{\sigma^2}\right) \tag{1}$$

where σ is the width of the node

- The thin plate spline function:

$$f(d) = d^2 \log(d) \tag{2}$$

The thin plate spline function, which is used in all the simulations throughout this paper, has the advantage that no width needs to be specified.

B. Dynamical RBF models

When RBF networks are used for modeling dynamical systems, past values of the system input and output variables must be included as inputs to the network. Given a system with R inputs and M outputs, the input vector to the RBF network at time instant k can be written as follows:

$$\begin{aligned}\mathbf{x}^T(k) &= [x_1(k),\ x_2(k),\dots,x_N(k)] \\ &= [u_1(k-1), u_1(k-2),\dots,u_R(k-p_R+1), u_R(k-p_R),\dots, \\ &\quad y_1(k-1), y_1(k-2),\dots,y_M(k-q_M+1), y_M(k-q_M)]\end{aligned} \tag{3}$$

where p_R is the total number of past values of the system input u_R and q_M is the total number of past values of the system output y_M. Following this notation, a dynamic RBF network can be seen as a nonlinear autoregressive with exogenous inputs (NARX) model.

C. Training of RBF networks

Typical offline training of an RBF network, usually involves splitting the problem into two phases: First the centers of the hidden nodes are obtained using the k-means clustering algorithm, and in a second phase the weighting connections are calculated by simple linear regression. Recently, the fuzzy means algorithm has been proposed for substituting the time consuming k-means [12]. The fuzzy means is based on the concept of fuzzy partition of the input space and is faster than the conventional RBF training techniques, while at the same time it exhibits better approximation capabilities.

III. Adaptive fuzzy means algorithm

The adaptive fuzzy means algorithm is presented as an alternative of the standard fuzzy means methodology, for modeling systems with changes in operating region or system dynamics. In order to deal with both of these cases, the algorithm combines two levels of adaptation, in a unified scheme:

- Adaptation of the structure of the hidden layer, based on the notion of fuzzy partitioning
- Adaptation of the connection weights between the hidden layer and the output layer, based on the RLS with exponential forgetting algorithm

An overview of the algorithm is depicted in Fig. 2. It should be noted that the algorithm makes use of some operational parameters that need to be defined beforehand. These are:

- The number of consecutive time steps N_d that a center is not assigned to an input example before it is removed from the hidden layer of the network
- The size of the moving time window N_s, which is used for storing past input-output examples
- The forgetting factor λ for the RLS method.

The first level of adaptation is based on the concept of

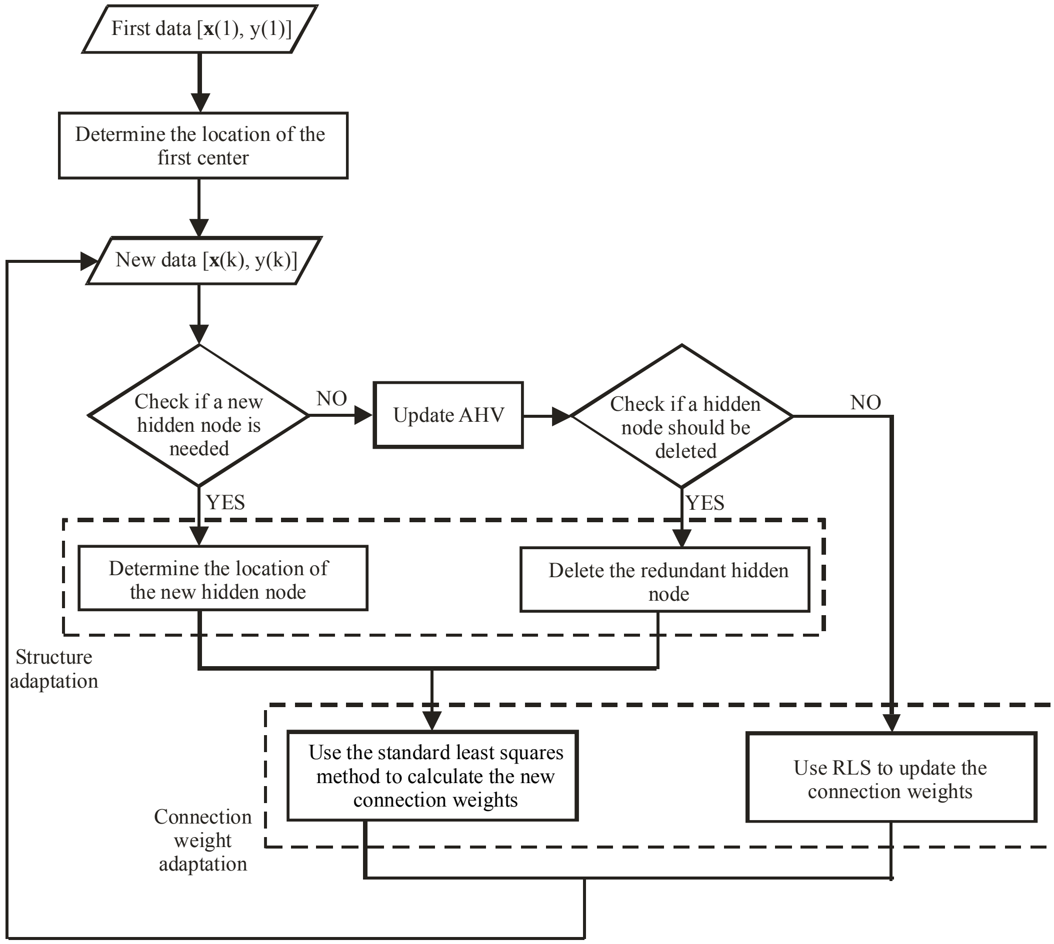


Fig. 2. Generic overview of the algorithm

fuzzy partition of the input space. This involves partitioning the space of each input variable x_i, $i \in 1,...,N$ into c_i triangular fuzzy sets:

$$\mathrm{T}_i = \{\mathrm{A}_{i,1}, \mathrm{A}_{i,2}, ..., \mathrm{A}_{i,c_i}\} \tag{4}$$

where each fuzzy set $A_{i,j}$ is fully described by its center and width elements:

$$A_{i,j} = \{a_{i,j}, \delta a_{i,j}\} \tag{5}$$

The membership function $\mu_{A_{i,j}}(x_i(k))$ of the input $x_i(k)$ to the fuzzy set $A_{i,j}$ is defined as:

$$\mu_{A_{i,j}}(x_i(k)) = \begin{cases} 1 - \dfrac{|x_i(k) - a_{i,j}|}{\delta a_{i,j}} & , if\ x_i(k) \in [a_{i,j} \text{-} \delta a_{i,j}, a_{i,j} + \delta a_{i,j}] \\ 0 & , otherwise \end{cases} \tag{6}$$

The idea of fuzzy partitioning can be extended to the entire input space in order to create a number of fuzzy subspaces, where each fuzzy subspace is defined as a combination of N particular fuzzy sets. The multidimensional membership function $\mu_{\mathbf{A}^l}(\mathbf{x})$ of an input vector $\mathbf{x}$ into a fuzzy subspace $\mathbf{A}^l$, is then defined as:

$$\mu_{\mathbf{A}^l}(\mathbf{x}) = \begin{cases} 1 - rd^l(\mathbf{x}), & if \quad rd^l(\mathbf{x}) \le 1 \\ 0 & , otherwise \end{cases} \tag{7}$$

where $rd^l(\mathbf{x})$ is the Euclidean relative distance [16] between $\mathbf{A}^l$ and the input data vector $\mathbf{x}$. A fuzzy subspace that assigns to the input vector the maximum membership degree, is the closest subspace to this vector, since it corresponds to the smallest Euclidean relative distance.

Having defined the concept of fuzzy subspaces, the algorithm works as follows: As soon as the first input data point is available from the system, the fuzzy subspace that is closer to it is calculated. The center of this subspace becomes then the center of the first hidden node. As we proceed to each new time step, a new input example is available, and the algorithm first checks whether it can be assigned to any of the already selected fuzzy subspaces. If this is not possible, a new fuzzy subspace which is the closest one to the particular input vector is selected, and the center of this subspace becomes the center of a new hidden node. In this way, the algorithm adds hidden node centers, in order to cover the area of the input space, where data are present. In the case that a new center is not needed, then the algorithm checks for hidden node centers that have not been assigned recently to an input vector. This check is based on information about the history of node activations which is stored in a vector called Activation History Vector – AHV. If such a center exists, then it is removed from the hidden layer of the network.

The second level of adaptation involves the modification of the connecting weights. In the case where no centers have been added or deleted from the hidden layer, the modification of the weights is performed by using the RLS with exponential forgetting algorithm.

If the structure of the hidden layer has been modified either by adding or deleting a node, then the connection weights need to be recalculated. This calculation is based on a moving time window, where a number of past input-output data are stored. The new connection weights are obtained by regressing the outputs of the hidden layer on the real outputs of the system. Before the regression takes place, the outputs of the hidden layer and the real outputs of the system are weighted in a similar way to the RLS algorithm, so that the influence of the oldest data points is weakened and more importance is given to the new data points.

IV. Application: Adaptive modeling of a continuous digester

The continuous digester is a very important process in a pulp and paper plant, since its role is to convert wood chips to pulp by removing a wood component called lignin. The removal of lignin is achieved through combined chemical and thermal treatment, by means of adding a special mixture called white liquor to the wood chips inside the reactor. The residual amount of lignin in the pulp exciting the digester, is the kappa number which is a very important quality parameter.

Modeling the digester with first principle equations is very difficult, due to its complexity and nonlinearity. Though the use of black box identification techniques is more successful there are still some problems imposed by the fact that the reactor exhibits frequent changes in its dynamic behavior. Moreover, it is very often desirable to produce pulp of different quality, which means that the operating region of the digester is frequently modified.

The objective of this case study was to build an RBF model using the adaptive fuzzy means algorithm, in order to predict the dynamic behavior of the kappa number in the simulated digester which is depicted in Fig. 3. It should be noted that the reactor was simulated by solving a system of ODEs generated by the heat and mass balances along the digester [15]. The RBF network uses as inputs the temperatures of two of the white liquor flows which enter the digester in different locations. In order to take into account the high time lags between the locations of the flows, and the bottom part of the digester where the pulp discharges, we have used 14 past values for each variable, thus summing to a total of 28 input variables.

The performance of the adaptive fuzzy means was evaluated through two different cases:

A. *Case I: Change in the dynamics of the digester*

A common situation in the operation of the digester is that the type of the wood chips entering the reactor changes. This has a profound impact on the dynamics of the digester, since it affects the speed of the delignification procedure. For the particular case, we have simulated a similar situation where the feed to the digester changes from softwood to hardwood. First the digester was simulated with constant feed of softwood for 100 hours of operation, were the inputs were generated by using a random sequence of temperatures that produced kappa number values between 20 and 30. Input – output data were collected with half- hour frequency and split into a training and a validation

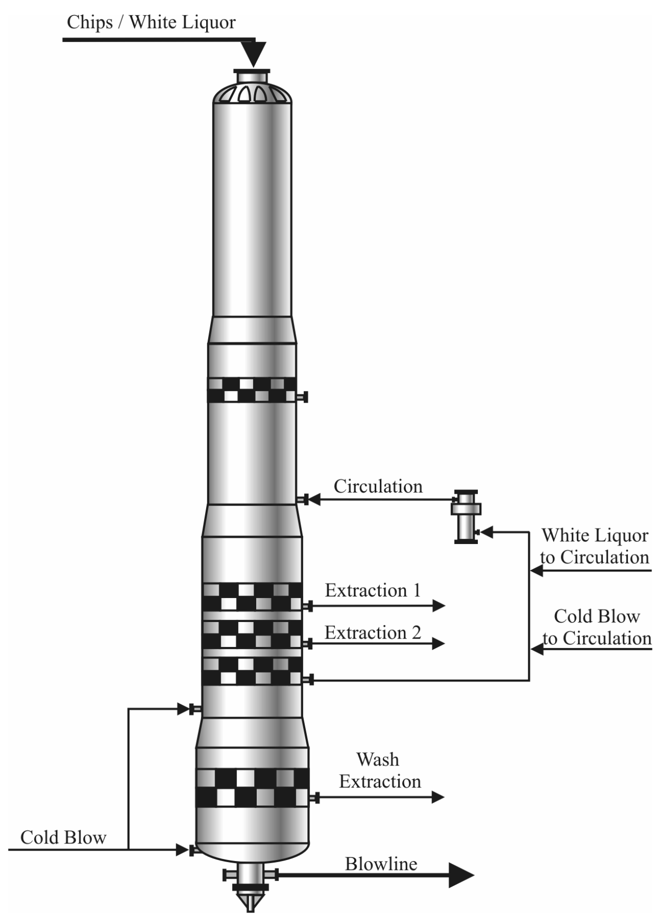


Fig. 3. A typical continuous digester

subset. The first subset was used to train offline an RBF network using the non-adaptive fuzzy means algorithm. Using a fuzzy partition of 7 fuzzy sets, the number of hidden nodes of the network was found to be 25. The performance of the produced network was evaluated on the second set. Fig. 4 depicts the predicted versus the real values of the kappa number. It can be seen that the network predictions are very accurate.

Starting with the network trained offline, we introduced a change in the feed of the reactor, from softwood to hardwood. The adaptive fuzzy means algorithm, with the operating parameters of table 1, was then used to adapt the network. The same set of input data was also fed to the static offline trained network, in order to compare its predictions with the ones produced by the adaptive network. The results are shown in Figs. 5a and 5b for the online and offline trained network respectively. Though the adaptive network initially fails to predict the kappa number, its performance improves gradually, as it adapts to the new data. On the other hand the static network is unable to give correct predictions, since it cannot adapt itself to the changes in the dynamics of the digester.

B. Case II: Change in the operating region of the digester

Another case where an adaptive model of the digester might prove useful is when the operators decide to produce pulp with a different kappa number. This implies a change in the operating region of the digester, since the range of the inputs needs to be shifted as well.

In order to examine this case, we gave random values to the input variables, so that the kappa number ranged between 30 and 40. The neural network trained offline in the previous case was used to simulate the dynamics of the system in two ways: First it was allowed to adapt itself using the fuzzy means algorithm with the operational parameters of table 1and then for comparison reasons it was also used as a static model. The results are shown in Figs 6a and 6b respectively. It can be observed that the adaptive network still outperforms the static one. This is due to the fact that the static network has a limited ability to extrapolate to the new operating region even if the dynamics

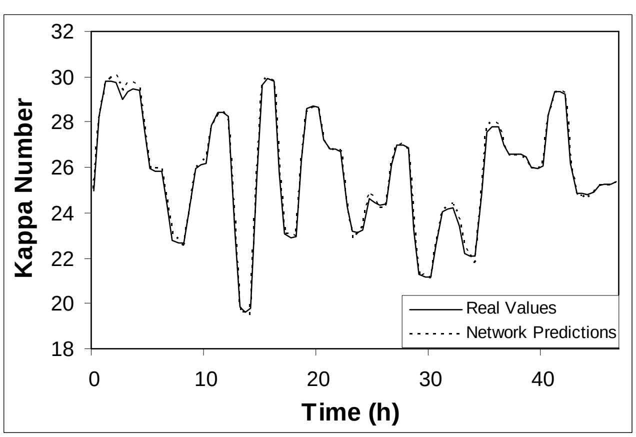
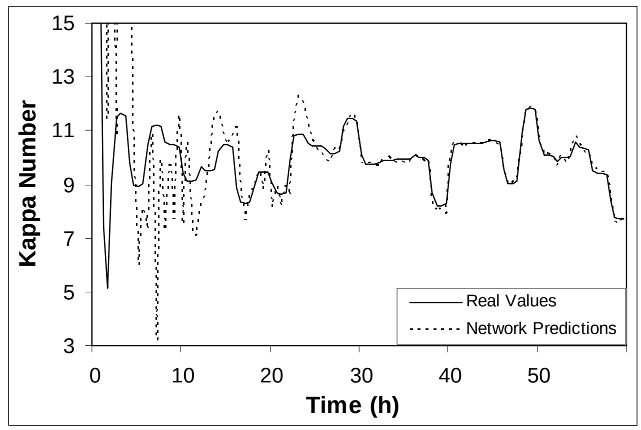


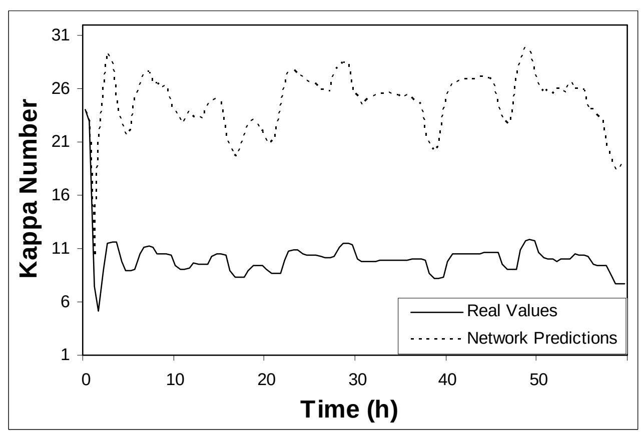
Fig. 4. Case I: Kappa number predictions for the static network in the validation data, before the change from softwood to hardwood occurs

Table 1. Operational Parameters for the fuzzy means adaptive algorithm

Parameter	Value
N_d	80
N_s	100
λ	0.85



(a)



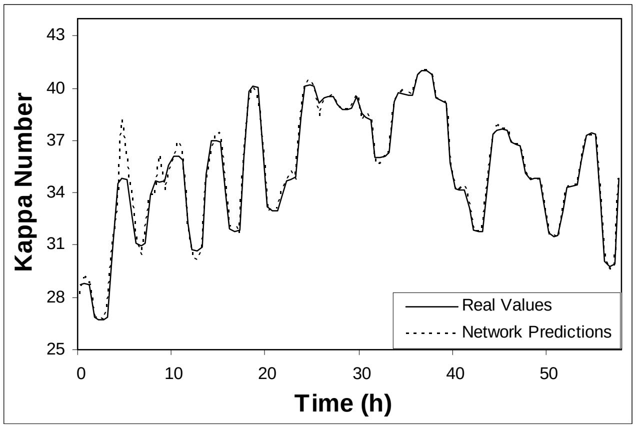
(b)

Fig. 5 Case I: Kappa number predictions, after the change from softwood to hardwood occurs, for (a) the adaptive network, (b) the static network

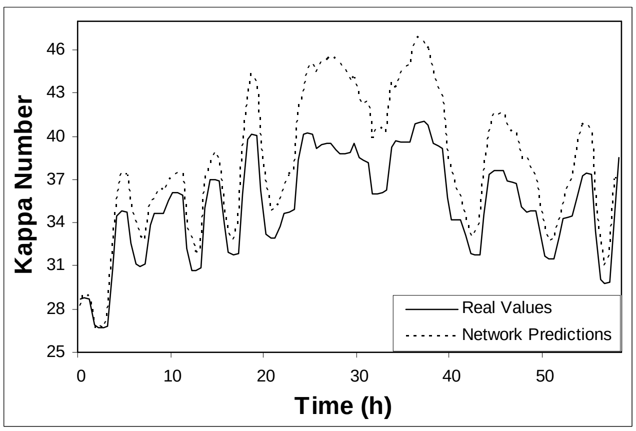
of the digester are not changing with time. On the contrary, the adaptive network, rearranges the locations of the hidden node centers so as to describe better the new data and at the same time it calculates suitable values for the connecting weights.

V. Conclusion

This paper presents a methodology for developing adaptive RBF network models for time varying continuous digesters, which are key elements in the pulp and paper industry. The method is based on the fuzzy means algorithm and offers a unified scheme for synchronous structure and weight adaptation of the RBF network. Thus, the algorithm is suitable for complex systems such as



(a)



(b)

Fig. 6 Case II: Kappa number predictions, after the change of operating region occurs, for (a) the adaptive network, (b) the static network

the continuous digesters that exhibit changes in their dynamics and/or their operating region. The efficiency of the adaptive fuzzy means algorithm is illustrated through a number of simulations where it is clearly shown that it outperforms the performance of non-adaptive RBF models.

The authors are currently working on an extension of the method, so that it can be incorporated into an adaptive Model Predictive Control (MPC) framework. This could provide a suitable scheme for controlling nonlinear systems with time varying behavior.

REFERENCES

[1] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations". In *Fifth Berkeley Symp. Math. Stat. and Prob.; Proceedings*, U. C. Berkeley Press, 1967, 281.

[2] C. Darken, J. Moody, "Fast Adaptive K-Means Clustering: Some Empirical Results", *IEEE INNS Int. J. Conf. On Neural Networks; Proceedings,* 1990, Vol. II, 233.

[3] J. Moody, C. Darken, "Fast Learning in Networks of Locally-Tuned Processing Units" *Neural Computation*. 1989, Vol. 1, 281.

[4] J.A. Leonard, M.A. Kramer, "Radial Basis Function Networks for Classifying Process Faults", *IEEE Control Systems*, 1991, 31.

[5] M.J.D. Powell "Radial Basis Functions for Multivariable Interpolation: A Review", In *Algorithms for Approximation*; Mason, J. C. and Cox, M. G., Eds.: Oxford, 1987, 143.

[6] S. Chen, S.A. Billings, C.F.N. Cowan, P.W. Grant, "Practical Identification of NARMAX Models using Radial Basis Functions" *International Journal of Control*, 1990, Vol. 52, 1327.

[7] C.J. Platt, "A resource allocating network for function interpolation" *Neural Computation,* 1991, Vol. 4, No. 4, 473.

[8] Q. Zhu, Y. Cai,.L. Liu, "A global learning algorithm for a RBF network" *Neural Networks*,1999, Vol. 12, 527.

[9] M.T. Musavi., W. Ahmed, K.H. Chan, K.B. Faris, D.M. Hummels, "On the training of Radial Basis Function Classifiers", *Neural Networks*, 1992, Vol. 5, 595.

[10] A. Leonardis, H. Bischof, "An efficient MDL-based construction of RBF networks" *Neural Networks*, 1998, Vol. 11, 963.

[11] S.A. Billings, G.L. Zheng, "Radial Basis Function Network Configuration Using Genetic Algorithms", *Neural Networks*, 1995, Vol. 8, 877.

[12] H. Sarimveis, A. Alexandridis, G. Tsekouras, G. Bafas, "A fast and efficient algorithm for training radial basis function neural networks based on a fuzzy partition of the input space" *Industrial and Engineering Chemistry Research*, 2002, Vol. 41, 751.

[13] S. Tan, J. Hao, J. Vandewalle, "Efficient identification of RBF neural net models for nonlinear discrete-time multivariable dynamical systems", *Neurocomputing*, 1995, Vol. 9, 11.

[14] C.F. Fung, S.A. Billings, W. Luo, "On-line supervised adaptive training using radial basis function networks", *Neural Networks*, 1996, Vol. 9, 1597.

[15] P.A. Wisnewski, F.J. Doyle, F. Kayihan, "Fundamental continuous pulp digester model for simulation and control", *AICHE Journal*, 1997, Vol. 43, No. 12, 3175.

[16] J. Nie, "Fuzzy control of Multivariable Nonlinear Servomechanisms with Explicit Decoupling Scheme" *IEEE Trans. Fuzzy Systems*, 1997, Vol. 5, 304.