Nonlinear Adaptive Model Predictive Control Based on Self-Correcting Neural Network Models

Alex Alexandridis and Haralambos Sarimveis
National Technical University of Athens, School of Chemical Engineering, Zografou Campus, Athens 15780, Greece

DOI 10.1002/aic.10505
Published online June 9, 2005 in Wiley InterScience (www.interscience.wiley.com).

Two major issues in process control are the nonlinearities and variations with time that are observed in the dynamics of the processes. In most cases these problems are confronted by robust linear controllers, which are frequently retuned to take into account changes in the operating region or the system dynamics. Obviously, the performance of these controllers is limited by the degree of nonlinearities and the frequency of process variations. In this paper we present a new model predictive control (MPC) framework that can deal with both these issues. The proposed methodology is based on a nonlinear dynamic radial basis function (RBF) model of the process that is able to correct itself as new information about the process dynamics becomes available. The adaptive training algorithm that is used is able to update both the structure and the parameters of the RBF model. The typical formulation of the on-line optimization problem is augmented by a persistent excitation condition that guarantees that enough perturbation is introduced to the system by the control moves. The proposed MPC framework is applied on the digester control problem and proves to be superior to other MPC configurations. © 2005 American Institute of Chemical Engineers *AIChE J,* 51: 2495–2506, 2005
Keywords: model predictive control, adaptive control, nonlinear control, radial basis function networks, digester control

Introduction

Model predictive control (MPC) has received much attention from the academic as well as the industrial community during the last two decades. In fact, MPC is the only control methodology that has found so many industrial applications[1] besides the traditional PID controllers. MPC relies on the following simple idea: If a discrete-time model of the process that correlates the manipulated variables with the controlled variables is available, then by inverting this model we can yield the sequence of moves that drives the process to the desired set point in an optimal way. The idea has been extended, so that it can take into account such phenomena as process/model mismatches, multiple input–multiple output (MIMO) systems, and process constraints. Numerous MPC frameworks have been proposed in the literature, yet all of them result in the formulation of an optimization problem that needs to be solved in real time.

It is obvious from the preceding description that the approximation capabilities of the used model have a profound effect on the performance of the MPC scheme. Most MPC formulations and studies that can be found in the literature and almost all industrial implementations make use of a linear process model. Among them, dynamic matrix control (DMC)[2] has been applied widely in industrial systems, owing its success mainly to the use of a linear finite-step response model whose parameters can be easily determined by running a few step tests and a linear or quadratic objective function that simplifies the solution of the optimization problem.

Still, many processes in the chemical industry are nonlinear and in this case linear model–based controllers provide good performance only close to the operating region used for extracting the linear model. Especially in the case of severe nonlinearities, this region can be very narrow. This led many

Correspondence concerning this article should be addressed to H. Sarimveis at hsarimv@chemeng.ntua.gr.

researchers to consider nonlinear models that offer better approximation capabilities at the price of increasing the complexity of the optimization problem. A variety of nonlinear models has been used, including models based on first principles equations[3]; Volterra polynomials[4,5]; Hammerstein and Wiener models[6,7]; and nonlinear models that are automatically generated by genetic programming.[8]

Artificial neural networks (ANNs) offer an interesting alternative to the problem of nonlinear system modeling and have also been used extensively as models for MPC. A review of the applications of neural networks in MPC techniques is given by Hussain.[9] Radial basis function (RBF) networks[10] constitute a special network architecture that presents some remarkable advantages over other neural network types: simpler structures, less synaptic connections, and faster training algorithms. Not surprisingly, RBF networks have been used as dynamic models in model-based control schemes. Pottmann and Seborg[11] introduced an RBF-based approach, where the control actions were generated by an RBF network trained to approximate the inverse of the process dynamics. Bhartiya and Whiteley[12,13] proposed an MPC approach, where a factorized RBF network serves as the process model. In their work a nonlinear optimization problem is formulated and solved online, where factorization of the RBF network increases the computational efficiency of the control algorithm.

Although the introduction of nonlinear models in MPC formulations may solve problems concerning the nonlinear nature of chemical processes, there are other reasons that may cause model inaccuracies. Very often, process dynamics depend on parameters that vary with time, thus making it impossible to train offline a model that is accurate under all circumstances. Many industrial systems are also subject to frequent changes of the operating region, which can severely deteriorate the performance of a model. For example, when approximating a nonlinear process with a model linearized around a specific steady state, it is obvious that a change in the operating region creates a process/model mismatch. The same problem may appear with black-box nonlinear models because of their inability to extrapolate far from the training region. In any case, the result when implementing a static MPC scheme in a time-varying process could be poor control performance or even instability. To confront the problems that arise in controlling time-varying processes, various methods of adaptive control have been proposed. Maiti and Saraf[14] presented an adaptive DMC where the process is identified in terms of pseudo impulse response coefficients that are updated with time. Dougherty and Cooper developed a multiple model adaptive strategy for single-loop MPC[15] and for multivariable MPC as well.[16] Their approach mainly relies on combining multiple linear DMCs, whereas the final control action is obtained by interpolating between the individual controller outputs. However, as the authors report, their work focuses on processes that are nonlinear with respect to the operating level, but static with time. A more practical approach when dealing with time-varying systems is to update the process model in real time. Genceli and Nikolaou[17] proposed a methodology that combines constrained MPC with simultaneous online model identification (MPCI) for finite impulse response (FIR) models. The objective is to find a way to perturb the process so that sufficient information for the identification procedure is collected, while at the same time a good control performance is achieved. They manage this by adding a constraint in the MPC optimization problem that represents the persistent excitation (PE) criterion. In a follow-up article[18] the methodology is extended so that it covers deterministic autoregressive with exogenous input (DARX) models. Vuthandam and Nikolaou[19] proposed a less computationally expensive formulation of the simultaneous MPCI problem, by expressing the PE constraint in the frequency rather than in the time domain. A detailed summary of the MPCI framework is given by Nikolaou.[20]

Still, all these approaches use linear process models. There is relatively little work done for adaptive nonlinear MPC. In Morningred et al.,[21] an adaptive nonlinear predictive controller is proposed, where the process is modeled by a nonlinear autoregressive moving average with exogenous inputs (NARMAX) model, chosen with a priori process knowledge. Tsai et al.[22] introduced a nonlinear adaptive MPC scheme, but the adaptation in their work is performed through a neural adaptive controller that runs in parallel with a model predictive controller. The final control decision is calculated by weighting the outputs of the two controllers.

In this article, we propose a new adaptive nonlinear MPC configuration, where the model used for predicting the future behavior of the process is a discrete-time dynamic RBF network. The adaptation of the model is implemented in a straightforward way using the adaptive fuzzy means algorithm.[23] The advantage of this algorithm is that it updates both the structure of the network and the connection weights, thus ensuring that the model is suitably adapted to account for changes in the dynamics as well as the operating region of the process. Moreover, the PE criterion has been incorporated into the proposed MPC scheme, after making the necessary modifications to integrate it with the RBF network model.

To test the efficiency of the proposed adaptive MPC configuration, we implemented it on a complex nonlinear industrial reactor used in the pulp and paper industry: the continuous pulp digester.[24] A rigorous first-principles model[25] was used to simulate the digester behavior and various case studies that included set-point tracking and disturbance rejection were simulated. The superior performance of the proposed framework compared to other linear and nonlinear MPC schemes illustrated the efficiency of the method.

The rest of this article is organized as follows: A short introduction to the fuzzy means algorithm is given in the next section, followed by the presentation of the proposed MPC framework. Next follows a discussion on the digester control problem and a description of the first-principles model used for simulating the digester dynamic behavior. Then, the case studies are presented, where the proposed methodology was applied and compared with other MPC configurations. In the final section we draw conclusions and set some directions for future research.

The Fuzzy Means Algorithm

The two main objectives in an RBF network training algorithm are the selection of number and locations of the RBF hidden node centers and the calculation of the connecting weights between the hidden and output layers. The formulation of a complete optimization problem that simultaneously calculates all the network parameters—based on the minimization of the prediction error over a set of input–output training data—is difficult because of the numerous discrete decision variables that are involved in the problem. Therefore, most RBF training network techniques try to meet the two objectives in separate phases. The first objective is the most difficult and time consuming, whereas the connection weights are trivially calculated through simple regression between the hidden layer responses and the real outputs of the system. For the selection of the hidden node centers, an unsupervised clustering algorithm is usually utilized, which is applied on the available input training data. A typical methodology is the k-means algorithm where the centers are calculated through an iterative procedure. Recently, a faster and more efficient alternative was proposed, called the *fuzzy means algorithm*.[26]

The speed of the fuzzy means algorithm arises from the fact that it requires only one pass of the training examples. Another important advantage is that it has the ability to automatically determine the structure of the hidden layer. The key concept behind the algorithm is the idea of partitioning the input space into a number of fuzzy sets, which we are going to describe briefly in the next subsection.

Fuzzy partition of the input space

Assuming a system with N input variables, the universe of discourse (domain) of each variable x_i, $i \in 1, \ldots, N$ can be evenly partitioned into c_i triangular fuzzy sets

$$T_i = \{A_{i,1}, A_{i,2}, \ldots, A_{i,ci}\} \qquad 1 \leq i \leq N \tag{1}$$

Supposing that $a_{i,j}$ is the center element of the fuzzy set $A_{i,j}$ and δa_i is half of the respective width, 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

$$\mu_{A_{i,j}}[x_i(k)] = \begin{cases} 1 - \dfrac{|x_i(k) - a_{i,j}|}{\delta a_i} & \text{if } x_i(k) \in [a_{i,j} - \delta a_i, a_{i,j} + \delta a_i] \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

By using the principle of fuzzy partitioning to the entire input space, we can partition it into C fuzzy subspaces, where

$$C = \prod_{i=1}^{N} c_i \tag{3}$$

Each fuzzy subspace $\mathbf{A}^l$ $(1 \leq l \leq C)$ can then be defined as a combination of N particular fuzzy sets and is represented as

$$\begin{aligned}\mathbf{A}^l &= [A^l_{1,j_1}, A^l_{2,j_2}, \ldots, A^l_{N,j_N}] \\ &= \{[a^l_{1,j_1}, a^l_{2,j_2}, \ldots, a^l_{N,j_N}], [\delta\alpha_1, \delta\alpha_2, \ldots, \delta\alpha_N]\}\end{aligned} \tag{4}$$

which in a more compact vector form can be written as follows

$$\mathbf{A}^l = \{\boldsymbol{\alpha}^l, \delta\boldsymbol{\alpha}\} \tag{5}$$

The idea of the membership function can be expanded similarly to multiple dimensions, by introducing the notion of the multidimensional membership function $\mu_{\mathbf{A}^l}[\mathbf{x}(k)]$ of an input vector $\mathbf{x}(k)$ into $\mathbf{A}^l$

$$\mu_{\mathbf{A}^l}[\mathbf{x}(k)] = \begin{cases} 1 - rd^l[\mathbf{x}(k)] & \text{if } rd^l[\mathbf{x}(k)] \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $rd^l[\mathbf{x}(k)]$ is the Euclidean relative distance between $\mathbf{A}^l$ and the input data vector $\mathbf{x}(k)$

$$rd^l[\mathbf{x}(k)] = \frac{[\sum_{i=1}^{N} [a^l_{i,j_i} - x_i(k)]^2]^{1/2}}{[\sum_{i=1}^{N} (\delta a_i)^2]^{1/2}} \tag{7}$$

The notion of the multidimensional membership function can be used to determine the closest fuzzy subspace to a given input data vector $\mathbf{x}(k)$. It can be easily proved that this fuzzy subspace is defined as the combination of fuzzy sets in each dimension that assign the maximum membership degrees to the particular inputs $x_1(k), x_2(k), \ldots, x_N(k)$.

The fuzzy means algorithm exploits the idea of fuzzy partitioning because it considers all the fuzzy subspaces as potential hidden node centers in the produced RBF network. The final hidden node centers are selected so that there is at least one fuzzy subspace that assigns a nonzero multidimensional degree to each input training vector.

The adaptive fuzzy means algorithm

Although the fuzzy means algorithm presents some remarkable advantages over the traditional training approach for RBF networks, it still cannot take into account changes that might occur in a process after training the model. This limitation was the motivation for the development of a different version of the fuzzy means algorithm,[23] targeting at the online identification of RBF models for time-varying systems. This work introduced a complete adaptive training procedure for dynamic RBF network models using a twofold adaptation scheme:

(1) Adaptation of the structure of the hidden layer, based on the fuzzy partition of the input space.

(2) Adaptation of the connecting weights between the hidden layer and the output layer, based on the recursive least squares (RLS) with exponential forgetting algorithm.

The algorithm starts with the first input data point and calculates the closest fuzzy subspace. The first RBF center is assigned to the center of this subspace. Then, as soon as a new input becomes available, the algorithm first checks whether there exists at least one selected fuzzy subspace that assigns to this data point a nonzero membership degree. If the answer is negative a new node is added to the hidden layer of the RBF network and is located on the center of the closest fuzzy subspace. Otherwise, the algorithm checks whether there is any center that has not been activated recently [information about the history of node activations is stored in a vector called *activation history vector* (AHV)]. If such a center exists, it is removed from the hidden layer. These steps constitute the structure-adaptation mechanism of the algorithm. With respect to the adaptation of the connection weights, there are two alternatives.

If the structure of the hidden layer has been modified either by adding or by deleting a node, then the connection weights need to be recalculated. This calculation is based on a moving

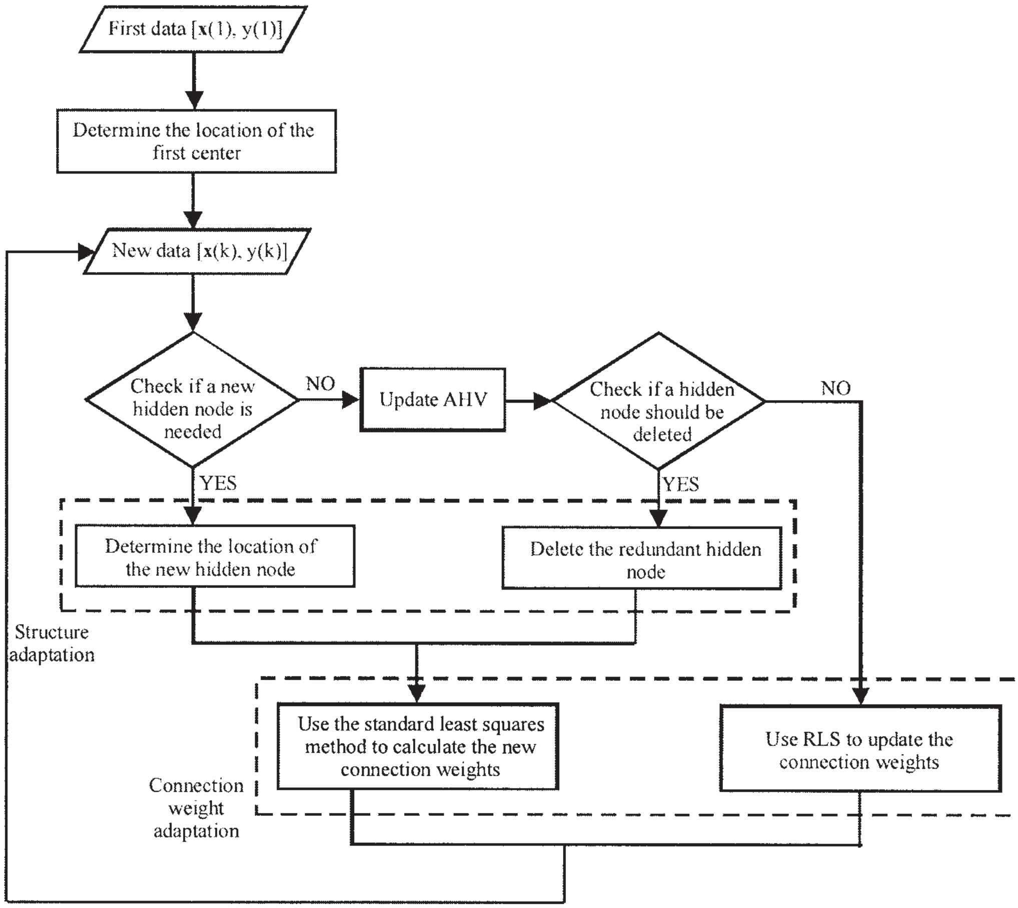


Figure 1. An overview of the adaptive fuzzy means algorithm.

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, for the entire time window. Assuming that M is the number of output variables in the model, N_s is the length of the moving time window, and $L(k)$ denotes the number of hidden nodes at time point k after the structure adaptation, the equation that calculates the output weights is

$$\mathbf{W}(k) = \mathbf{P}(k) \cdot \mathbf{R}^T(k) \cdot \mathbf{Y}(k) \tag{8}$$

where $\mathbf{W}(k)$ is the $L(k) \times M$ matrix that stores the new values of connection weights, $\mathbf{R}(k)$ is an $N_s \times L(k)$ matrix containing the responses of the current hidden layer structure over the entire length of the moving time window, and $\mathbf{Y}(k)$ is an $N_s \times M$ matrix where the real outputs of the system over the moving time window are stored. $\mathbf{P}(k)$ is the inverse of the information matrix corresponding to the hidden node outputs

$$\mathbf{P}(k) = [\mathbf{R}^T(k) \cdot \mathbf{R}(k)]^{-1} \tag{9}$$

On the other hand, when there has been neither addition nor deletion of an RBF center during the structure-adaptation mechanism, the algorithm updates the existing connection weights using the RLS with exponential forgetting algorithm, which is described by the following set of equations[27]

$$\mathbf{W}(k) = \mathbf{W}(k-1) + \mathbf{q}(k)[\mathbf{y}(k) - \mathbf{r}(k)\mathbf{W}(k-1)] \tag{10}$$

$$\mathbf{q}(k) = \mathbf{P}(k-1)\mathbf{r}^T(k)[\lambda + \mathbf{r}(k)\mathbf{P}(k-1)\mathbf{r}^T(k)]^{-1} \tag{11}$$

$$\mathbf{P}(k) = [\mathbf{I} - \mathbf{q}(k)\mathbf{r}(k)]\mathbf{P}(k-1)/\lambda \tag{12}$$

where λ is the forgetting factor, $\mathbf{r}(k)$ is a $1 \times L(k)$ vector containing the responses of the hidden layer nodes at time point k and $\mathbf{y}(k)$ is a $1 \times M$ vector containing the respective true outputs of the system. An overview of the adaptive RBF training algorithm is shown in Figure 1.

It should be noted that both the adaptive and static versions of the fuzzy means algorithm are briefly described in this paper for facilitating the comprehension of the adaptive MPC configuration. The interested reader can find more detailed descriptions in the references that are provided.

The Adaptive MPC Framework

Any model used in an MPC framework for predicting the process future behavior does not represent perfectly the real dynamics. Feedback can overcome some effects of poor models, but may not be truly effective.[28] Therefore, despite the fact that MPC systems implicitly include integral action and have great robust characteristics that produce good control performance, even in the presence of modeling er-

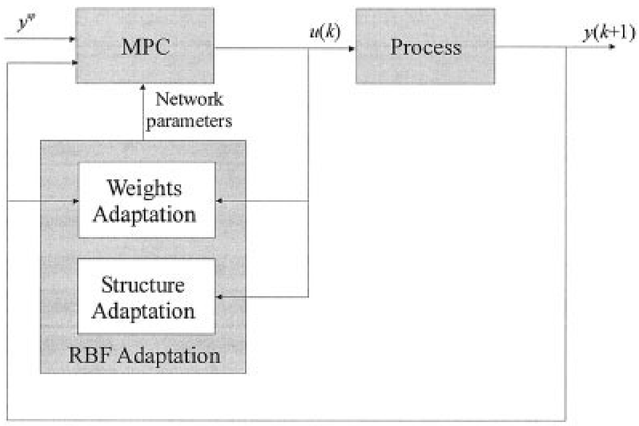


Figure 2. Structure of the adaptive MPC framework.

rors, the need for adapting the model of the process frequently arises. As discussed earlier, even if a good model of the process is initially available, it may substantially deteriorate with time as a consequence of modifications in the dynamics of the process or changes in the operating region. In time-varying systems extreme situations may occur, where even the static performance of an MPC scheme is unacceptable. An example can be found in Genceli and Nikolaou,[17] where a typical MPC scheme fails to achieve zero steady-state offset in a step set-point change. Adaptation of the process model under feedback control is known as the closed-loop identification problem.

In the case of MPC systems, it requires the online adaptation of a discrete-time dynamic model of the process, which is based on experimental measurements. The quality of these data is crucial for the success of the adaptation scheme.[27] Obviously, a parameter estimation under perfect feedback control (steady-state signals) cannot lead to useful parameter adaptation. Thus, the process must be minimally perturbed to obtain sufficient information for adapting the model. This process variation is achieved by introducing the notion of persistent excitation, which is a way to characterize input signals.[27] Because of the linear nature of the connection between the hidden layer and the output layer in the RBF architecture and the benefits of the adaptive fuzzy means algorithm, the aforementioned adaptation mechanism can be perfectly integrated into the MPC framework, by requiring the signal produced by the hidden layer of the network to be persistently exciting. The structure of this framework is depicted in Figure 2 and assumes that at each time point k a nonlinear optimization problem is formulated and solved. The objective is to calculate the optimal values of the manipulated variables $\mathbf{v}$ over a control horizon h_c, so that the error between the RBF model predictions and the desired set point over a prediction horizon h_p is minimized, while at the same time the control energy is also kept to a minimum. As soon as the optimization problem is solved, the first control move $\mathbf{v}(k)$ is implemented. Subsequently, the adaptive fuzzy means algorithm is used to update the RBF network model. In the sequel, the proposed adaptive nonlinear MPC configuration is presented in detail.

Formulation of the optimization problem

The optimization problem consists of the objective function that describes both the aforementioned control targets and a number of constraints

$$\min_{\Delta\mathbf{v}(k),\Delta\mathbf{v}(k+1),\ldots,\Delta\mathbf{v}(k+h_c)} \sum_{i=1}^{h_p} \|\mathbf{\Theta}[\hat{\mathbf{y}}(k+i) - \mathbf{y}^{sp}]\|_2^2 + \sum_{i=0}^{h_c} \|\mathbf{\Omega}\Delta\mathbf{v}(k+i)\|_2^2 \quad (13)$$

subject to

$$\hat{\mathbf{y}}(k+i) = \mathbf{NN}(k+i) + \mathbf{E}(k) \qquad 1 \le i \le h_p \quad (14)$$

$$\mathbf{E}(k) = \mathbf{y}(k) - \mathbf{NN}(k) \quad (15)$$

$$\mathbf{v}_{\min} \le \mathbf{v}(k+i) \le \mathbf{v}_{\max} \qquad 1 \le i \le h_c \quad (16)$$

$$-\Delta\mathbf{v}_{\max} \le \Delta\mathbf{v}(k+i) \le \Delta\mathbf{v}_{\max} \qquad 1 \le i \le h_c \quad (17)$$

$$\Delta\mathbf{v}(k+i) = 0 \qquad h_c + 1 \le i \le h_p \quad (18)$$

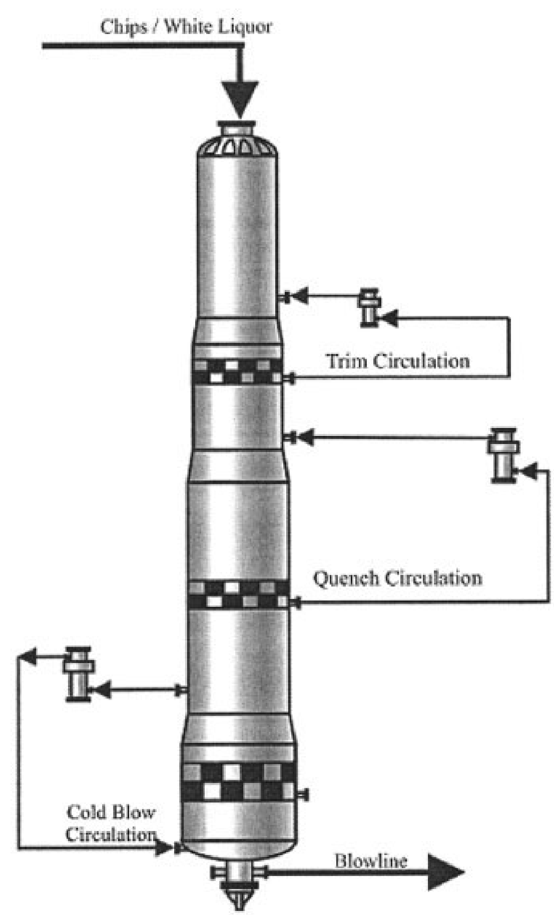


Figure 3. A typical one-vessel continuous digester.

Table 1. Parameters Regarding the Type of Wood Entering the Digester

Parameter	Value before the Disturbance	Value after the Disturbance
Preexponential factors		
$A_{1,1}$ (m^3/kg min)	0.2809	0.26945
$A_{1,2}$ (m^3/kg min)	6.035×10^{10}	5.1815×10^{10}
$A_{1,3}$ (m^3/kg min)	6.4509	4.287
$A_{1,4}$ (m^3/kg min)	1.5607	1.0093
$A_{1,5}$ (m^3/kg min)	1.0197×10^4	10634
$A_{2,1}$ (m^3/kg min)	9.26	8.937
$A_{2,2}$ (m^3/kg min)	0.489	0.3506
$A_{2,3}$ (m^3/kg min)	28.09	18.409
$A_{2,4}$ (m^3/kg min)	10.41	6.665
$A_{2,5}$ (m^3/kg min)	5.7226×10^{16}	5.9724×10^{16}
Fractions of solid components that do not react during pulping		
$a_{s_1}^{\infty}$	0	0
$a_{s_2}^{\infty}$	0	0
$a_{s_3}^{\infty}$	0.71	0.716
$a_{s_4}^{\infty}$	0.25	0.25
$a_{s_5}^{\infty}$	0	0
Stoichiometric coefficients for the consumption of effective alkali and hydrosulfide		
β_{OHL} (kg OH/kg lignin)	0.166	0.1616
β_{OHC} (kg OH/kg carbohydrate)	0.395	0.3855
β_{HSL} (kg HS/kg lignin)	0.039	0.0379

where $\mathbf{NN}(k)$ is the prediction at time step k employing an RBF network model that uses past values of the manipulated variables as inputs, $\mathbf{E}(k)$ is the current error between the actual output measurement and the model prediction and $\mathbf{\Theta}$, $\mathbf{\Omega}$ are the error and move suppression weights, respectively. Equation 14 denotes that the modeling error measured at time point k is assumed to be the same over the entire prediction horizon. Equations 16 and 17 pose upper and lower bounds on the values of the manipulated variables and the control moves as well, whereas Eq. 18 denotes that no control moves are allowed after the end of the control horizon.

The persistent excitation (PE) condition

The main objective in a control scheme is to keep the controlled variables as close to the desirable set points as possible. This drives the manipulated variables to almost constant values for long time periods. On the other hand, an online identification scheme requires a minimum level of variance in the information it receives from the process, so that it can correctly estimate the parameters of the identified model. This requirement can be expressed for linear models through the persistent excitation condition. A signal u is said to be persistently exciting of order n if at each time instance k there exists an integer m such that

$$\rho_0 \mathbf{I} \geq \sum_{i=0}^{m} \boldsymbol{\varphi}(k-i)\boldsymbol{\varphi}^T(k-i) \geq \rho_1 \mathbf{I} \tag{19}$$

where $\rho_1, \rho_2 > 0$ and $\boldsymbol{\varphi}(k) = [u(k-1)u(k-2)\ldots u(k-n)]^T$.[27] This condition guarantees that the information matrix $\boldsymbol{\varphi}\boldsymbol{\varphi}^T$ is positive definite and bounded and thus it is invertible and well conditioned.

Apparently, a conflict arises when trying to simultaneously perform process control and model parameter identification. Genceli and Nikolaou[17] have proposed an elegant way to face this conflict by integrating the persistent excitation condition as a constraint in the MPC optimization problem. In this article, the persistent excitation condition is reformulated so that it can be incorporated into the nonlinear MPC framework.

Although RBF networks provide nonlinear mappings between the input and output variables, the relationship between the responses of the hidden nodes and the outputs of the network is linear. This enables us to calculate the connection weights by linear regression through the inversion of the information matrix corresponding to the outputs of the hidden nodes (Eq. 9). Consequently, the persistent excitation condition can be formulated so as to guarantee that at each time step, the aforementioned information matrix is well conditioned. As the outputs of the hidden nodes are in essence explicit functions of the manipulated variables, the persistent excitation condition forces the manipulated variables to perturb sufficiently enough so that successful identification of the connection weights can be achieved. Thus, in the context of the proposed MPC framework, the persistent excitation condition for the outputs of the hidden nodes can be formulated as follows

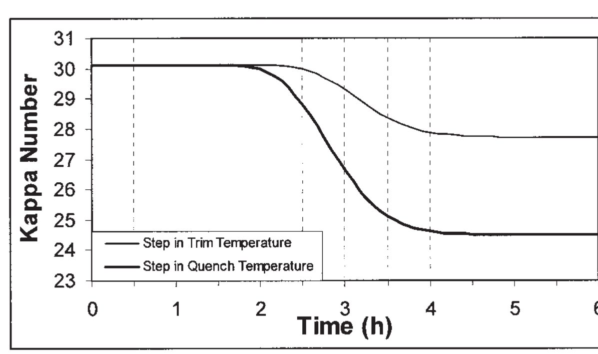


Figure 4. Kappa number response to step changes on the manipulated variables T_{trim} and T_{quench}.

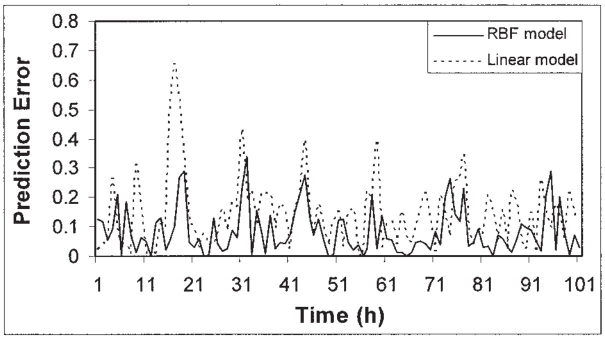


Figure 5. Prediction errors for the RBF and linear models.

$$\sum_{i=0}^{N_S-1} \mathbf{r}(k-i+j)^T\mathbf{r}(k-i+j) \geq \rho_1\mathbf{I} \qquad j=0,1,\ldots,h_p \tag{20}$$

An upper bound is not necessary for the information matrix since the values of the manipulated variables are bounded from above by constraint 16 and so are the responses of the hidden layer nodes. When the structure of the network is modified, the least-squares calculation is performed on a moving time window of length N_s, so $N_s - 1$ has replaced the parameter m that appears in inequality 19.

Although the optimization problem is solved for the entire length of the control horizon h_c, only the first control move is implemented. Thus to minimize the computational burden of the algorithm and at the same time to avoid the conflict between constraint 18 that requires the manipulated variables to remain constant after the end of the control horizon and condition 20 that forces the manipulated variables to fluctuate during the entire prediction horizon, the persistent excitation condition is used only for the current time step. Thus inequality 20 now becomes

$$\sum_{i=0}^{N_S-1} \mathbf{r}(k-i)^T\mathbf{r}(k-i) \geq \rho_1\mathbf{I} \tag{21}$$

or following the notation defined in Eq. 9,

$$\mathbf{R}^T(k)\cdot\mathbf{R}(k) \geq \rho_1\mathbf{I} \tag{22}$$

It should be noted that inequality 22 implies that the matrix $\mathbf{R}^T(k)\cdot\mathbf{R}(k) - \rho_1\mathbf{I}$ is positive semidefinite. Obviously the value of the parameter ρ_1 controls the level of "excitation " and should be kept at a small value so that aggressive moves of the manipulated variables are prevented. Following the above analysis, the persistent excitation condition is integrated into the MPC structure, by adding inequality 22 as a constraint into the MPC optimization problem defined in the previous subsection.

Table 2. Operational Parameters Used by the Four Controllers

Controller Parameter	Value
$\mathbf{\Omega}$	[1 0; 0 1]
$\mathbf{\Theta}$	[1]
h_c	14
h_p	19
$\mathbf{v}_{\min}$	[145 145]
$\mathbf{v}_{\max}$	[165 165]
$\Delta\mathbf{v}_{\max}$	[2 2]

Table 3. Additional Parameters Used by the Adaptive Controllers

Controller Parameter	Value for RBF–MPC	Value for Linear MPC
N_S	60	60
λ	0.95	0.95
ρ_1	10^{-4}	10^{-4}

Solution of the optimization problem

Equations 13–18 together with inequality 22 define a nonlinear optimization problem with linear and nonlinear constraints. The solution of this problem is not trivial and the authors are currently working on developing an evolutionary algorithm that will approximate the global solution to this problem in minimum time. However, for the purposes of this article, which are to present the integrated MPC–adaptive RBF framework, the sequential quadratic programming (SQP) algorithm included in the *fmincon* MATLAB function has been used. The simulations that follow have shown that the SQP algorithm is sufficient to solve the optimization problem in acceptable time, as long as the parameter ρ_1 that controls the excitation level is kept at small values.

Case Study: Adaptive Control of a Continuous Pulp Digester

This section presents an illustrative example where the proposed methodology is used for the adaptive control of a continuous pulp digester. The section is organized in three subsections: In the first subsection, we describe briefly the continuous digester and discuss some aspects regarding the control of this unit. Then the rigorous model that has been used for simulating the dynamic behavior of the digester is described. In the third subsection, the results of the simulations are presented, compared with those obtained by using different control schemes, and discussed.

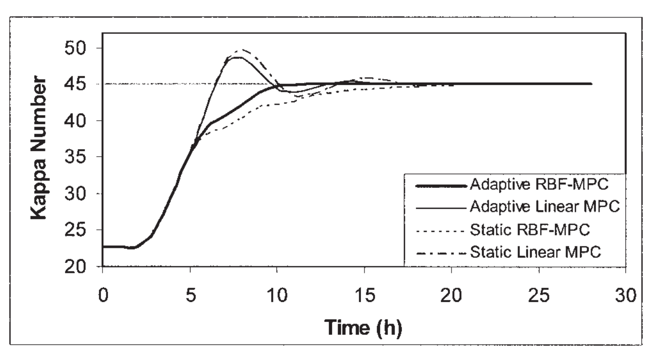


Figure 6. Case I: responses for the four controllers.

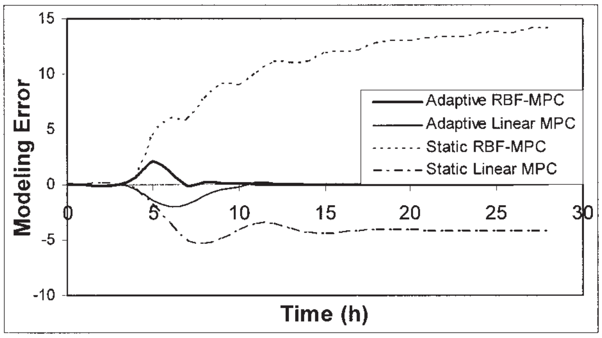


Figure 7. Case I: modeling errors for the four controllers.

Control of continuous digesters

Continuous pulp digesters are complex tubular reactors that are responsible for the conversion of wood chips to pulp. This is a key stage in a pulp and paper plant because it determines the quality characteristics of the brown pulp and, subsequently, of the produced paper. The wood chips enter at the top of the digester, along with a solution of sodium hydroxide and sodium sulfide, which is known as white liquor. A variety of reactions happen between the wood components and the white liquor, but the most important one is the removal of a substance called lignin, which binds the wood fibers together. The presence of the alkaline solution causes fragmentation of the lignin molecules into smaller segments, whose sodium salts are soluble in the cooking liquor. The reaction of delignification is temperature sensitive, and thus a specific temperature profile must be maintained inside the reactor. This is accomplished by extracting flows from the main body of the digester, manipulating their temperature, and reinserting them back into a different zone of the reactor. Figure 3 shows a typical one-vessel digester with three recirculation flows.

The residual amount of lignin in the pulp that discharges from the digester is a key indicator of its quality and is represented by a parameter called the *kappa number*. The parameter is directly proportional to the ratio of the residual lignin over the sum of all the wood substances in the discharging pulp and is measured either using online concentration analyzers or offline in the lab.

With respect to the control of the continuous digester, the objective is to keep the kappa number at the desirable set point, thus producing pulp of steady quality. This is not an easy task because of a number of difficulties:

- The nonlinear dynamics, which make it difficult to develop an accurate dynamic model of the process.
- The large retention times, which can be up to 8 h.
- The numerous disturbances that affect the quality of the produced pulp. The most important disturbance is the quality of wood chips, which varies during the daily operation, due to the different sizes and ages of the trees from which the chips originate.
- The frequent grade changes.

Some attempts have been reported in the literature for applying model-based controllers in continuous digesters. Wisnewski and Doyle[29] proposed an approach for selecting the manipulated variables and secondary measurements and then implemented the selections in a linear MPC scheme for disturbance rejection. Amirthalingam and Lee[30] and Doyle and Kayihan[31] used subspace identification techniques to identify linear models that are incorporated into MPC schemes for kappa number control. In the latter work, MPC is used to control the profile of the kappa number along the reactor.

Nonetheless all these approaches use linear model–based controllers. In this paper, the proposed adaptive MPC configuration is used in the digester control problem, successfully addressing the two important problems in kappa number control that were mentioned above, that is, the nonlinear dynamic behavior and the frequent grade changes.

Simulator of the digester dynamic behavior

A rigorous first-principles model was used to simulate the dynamic behavior of the digester during the case study. This model is a variation of the extended Purdue model[25] that approximates the plug flow by partitioning the digester into a number of zones. Each zone is modeled separately as a CSTR that contains three phases: the wood phase, the free liquor phase, and the bound liquor phase. It should be mentioned that as the number of CSTRs increases, the model approximates more accurately the plug flow. A set of nonlinear ordinary differential equations (ODEs) is created by formulating the mass and energy balances for all three phases in each CSTR. The state variables in the set of ODEs are the concentrations of the various components and the temperatures in all three phases. The CSTRs are coupled because the output of each CSTR is used as input to the next one. The resulting system of

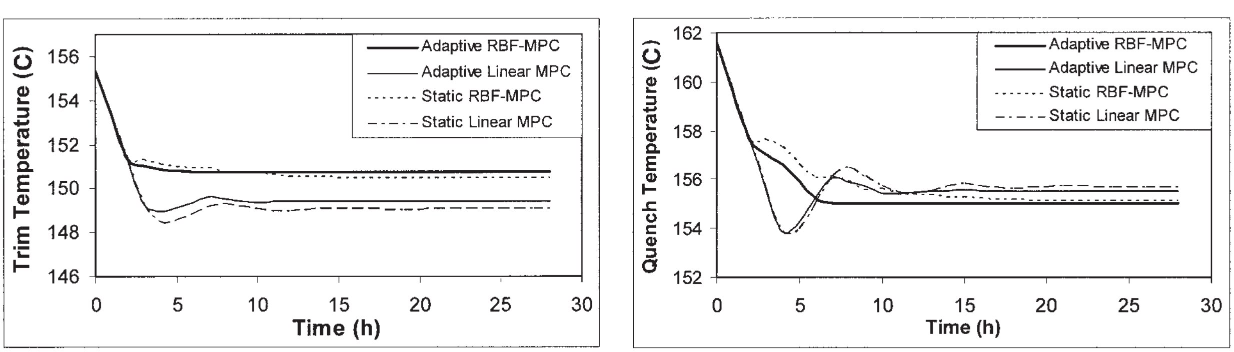


Figure 8. (a) Case I: control moves for T_{trim}; (b) Case I: control moves for T_{quench}.

ODEs is solved with numerical methods, thus providing a detailed dynamic simulation of the digester.

In contrast to the extended Purdue model, where the user must define whether the free liquor flows upward or downward in each zone, in this variation the direction of the free liquor flow is calculated automatically, by performing a mass balance on the free liquor throughout the digester. A similar idea appears in Bhartiya et al.[32]

The particular case study involved the simulation of a typical one-vessel digester similar to the one depicted in Figure 3. The digester was split into 50 zones. The physical dimensions of each zone are those reported by Wisnewski et al.,[25] but in our simulations the trim and quench flows are assumed to leave the digester from zones 10 and 16 and reenter at zones 9 and 15, respectively, whereas the cold blow flow leaves at zone 37 and reenters at the last zone of the reactor. This configuration produced a set of 950 nonlinear ODEs that were solved using the *ode23* solver of MATLAB.

Simulations—Results

In the simulated scenarios that are presented here, the target was to control the kappa number characterizing the pulp that discharges from the digester. The type of wood entering the digester was assumed to be softwood and the parameters that characterize this type of wood are shown in the first column of Table 1, retaining the notation used by Wisnewski et al.[25] The temperatures of the quench and trim flows were used as manipulated variables, so that the control system can manipulate the temperature in the upper and middle parts of the digester, where the bulk of the delignification process occurs. This is a typical choice advised by pulp and paper industry experts. The time span between each control move was selected to be 0.5 h, bearing in mind the large retention times inside the reactor.

Four different controllers were implemented in each case study: The first one was the proposed adaptive RBF–MPC scheme, whereas for comparison purposes we also tested an adaptive linear MPC controller, a static RBF–MPC controller, and a static linear MPC controller. The following procedure was executed to generate the models used by the four controllers:

A set of 200 training data was collected by exciting the digester with a random sequence of input temperatures that produced kappa number values between 20 and 30. The input and output training data were scaled so that their values were

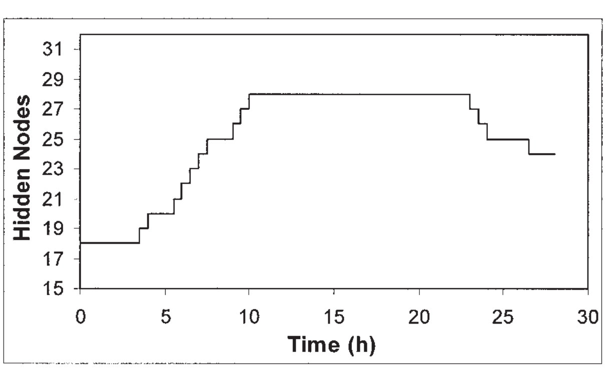


Figure 9. Case I: development of hidden layer structure with time.

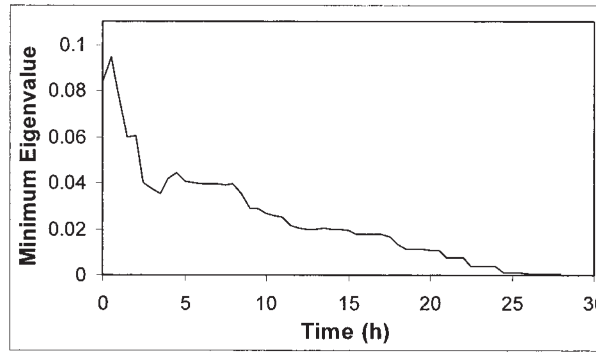


Figure 10. Case I: minimum eigenvalue of the matrix $\mathbf{R}^T(k) \cdot \mathbf{R}(k) - \rho_1\mathbf{I}$ with time.

in the same order of magnitude. Afterward, the nonadaptive fuzzy means algorithm was used to train offline a dynamic model that correlates the kappa number with the trim and quench flow temperatures. To correctly define the number of past inputs used by this model, the large dead time between the two manipulated variables and the kappa number should be taken into account. Figure 4 depicts the kappa number responses after applying step changes of 3 degrees to the manipulated variables T_{trim} and T_{quench}. It can be seen that there is a 2-h dead time, whereas the reactor reaches a steady state at about 4.5 h after the steps have occurred. Thus, the effective time span where changes in the output response are observed is between 2 and 4.5 h after an input change has occurred and a satisfactory RBF dynamic model based on past values of the manipulated variables can be expressed as follows

$$Kappa(k) = NN[T_{trim}(k-4), \ldots, T_{trim}(k-9), T_{quench}(k-4), \ldots, T_{quench}(k-9)] \quad (23)$$

The produced RBF model consisted of 18 hidden node centers and it was used throughout the simulations by the static RBF–MPC controller and only as an initial model by the adaptive RBF–MPC scheme.

To test the linear MPC approaches, a linear FIR model was developed based on the same set of 200 training data and using the same past values of the manipulated variables. The RBF network and the linear model were evaluated using a validation set consisting of 100 input–output examples in the same operating region. The absolute values of the prediction errors for

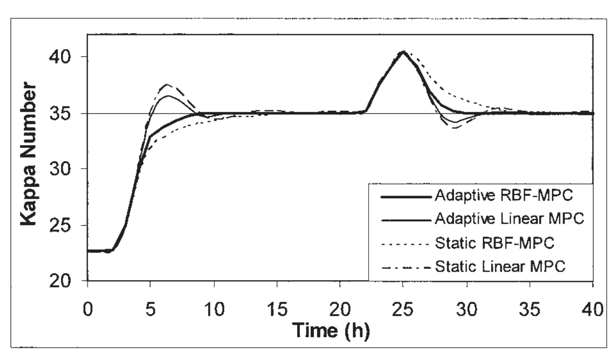


Figure 11. Case II: responses for the four controllers.

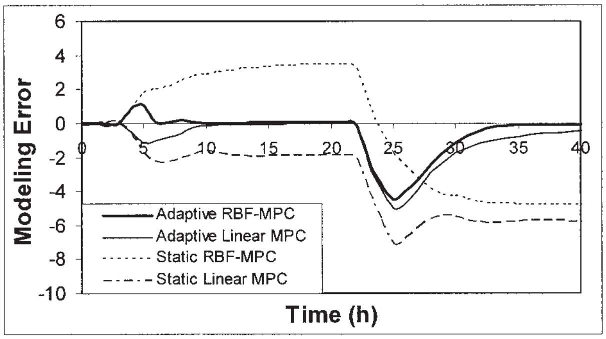


Figure 12. Case II: modeling errors for the four controllers.

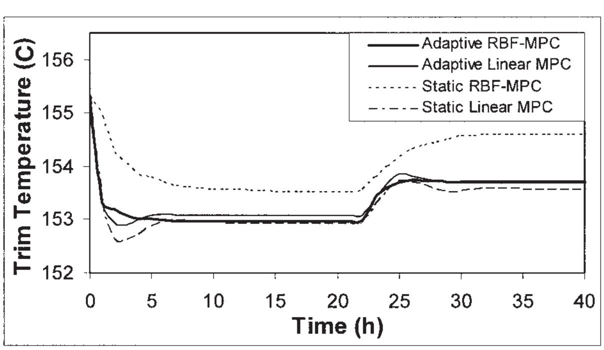
the two models are shown in Figure 5. The corresponding SSEs are equal to 1.33 for the RBF network and 3.46 for the linear model, showing that there is a significant advantage of the RBF model, as far as prediction accuracy is concerned.

The four MPC configurations were used in two test cases that are described next. As far as the proposed MPC configuration is concerned, at each time instance the optimal control moves were first calculated by solving the minimization problem defined by Eqs. 13–18 and Eq. 22, followed by the RBF model adaptation using the fuzzy means algorithm. The adaptive linear scheme solved a similar minimization problem, where in Eqs. 14 and 15 the neural network predictions were substituted by the linear model predictions and in Eq. 22 the matrix $\mathbf{R}(k)$ was appropriately modified to contain current and past values of the manipulated variables over the moving time window. Adaptation of the linear model was performed at each time point using the RLS algorithm. To implement the static RBF–MPC and linear MPC schemes we excluded the PE condition (Eq. 22) from the list of constraints that describe their adaptive counterparts. For all four control schemes and both cases that are presented next, the parameters depicted in Table 2 were used, whereas the additional parameters used by the adaptive schemes are shown in Table 3. Because the scaled input and output values are of the same order of magnitude, the objective function weights were set to 1. The prediction horizon was selected to be twice the number of past inputs used by the model (including the dead time) plus 1 as a compromise between the benefits offered by a long prediction horizon and the effort to reduce the size of the optimization problem. The control horizon was selected smaller than the prediction horizon following the standard MPC approach.

Case I: Change in the Operating Region. In this case study the control objective was to drive the digester to a new operating region, different from the one considered during the training phase. As soon as the simulation started, a step change in the kappa number set point from 22.6 to 45 was implemented. The new set-point value was clearly beyond the range covered by the training data. The results of the simulation are summarized in Figures 6–8, where the responses of the controllers, the modeling errors, and the control moves are shown, respectively. The time needed for the solution of the online optimization problem formulated for the two nonlinear MPC configurations at each time instance was approximately 12 s on a Pentium IV 2.2-GHz processor running MATLAB 6.5. This is perfectly accepted, taking into account that there is a 0.5-h period between two successive control moves. It should be noted, however, that when increasing the value of the parameter ρ_1 the required computational time increases substantially.

As can be seen from Figure 7, the change in the operating region generates a large error for the four models. The static RBF model exhibits the largest modeling error, obviously stemming from its inability to extrapolate outside the training region. The static linear model also has a significant error, which is attributed to the nonlinear dynamics of the process. Contrary to the static schemes, the two adaptive models manage to eliminate the error after a few time steps. It should be noted, however, that the proposed approach eliminates the modeling error in a shorter time period. Figure 9 depicts the development of the hidden layer for the adaptive neural network, where the addition of 10 new nodes can be observed. These new nodes enable the adaptive RBF model to more accurately approximate the new operating region. Starting at time step 23, a number of nodes describing the old operating region are deleted as a result of long periods of inactivity. Figure 10 plots the minimum eigenvalue of the matrix $\mathbf{R}^T(k) \cdot \mathbf{R}(k) - \rho_1\mathbf{I}$ as a function of time. Because this remains positive during the simulation, the matrix is always positive definite and, thus, the persistent excitation condition is satisfied.

With respect to the responses of the four controllers, the adaptive RBF–MPC one is superior, given that the static nonlinear MPC is rather sluggish, whereas both linear controllers are more aggressive and generate oscillations. Note also that the adaptive RBF controller makes smoother moves to reach the set point, as can be seen in Figures 8a and 8b.



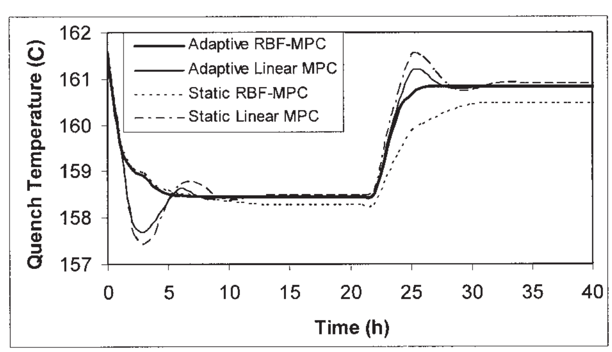


Figure 13. (a) Case II: control moves for T_{trim}; (b) Case II: control moves for T_{quench}.

Case II: Change in the Operating Region and Disturbance Rejection. During the actual operation of a digester, the reactor goes through a series of disturbances and set-point changes, which create system-model mismatches. The target of this case study is to simulate more realistically the real industrial operation of the digester, by introducing a change in the operating region followed by a disturbance in the quality of the raw material entering the reactor.

The four controllers used the same models with the previous case. Then the set point was changed to 35, which implies a smaller change in the operating region compared to that introduced in case I. After the controllers reached the new set point at time point 20, a disturbance was introduced, by changing the parameters that define the type of wood entering the top of the digester to the values of the second column of Table 1.

Figures 11–13 show the results. In this case, modeling error is generated by both the change in the operating region and the disturbance. As can be seen in Figure 14, which depicts the development of the hidden layer with time, the adaptive RBF model adds five RBF centers to deal with the change in the operating region. During the rejection of the disturbance, two hidden nodes are eliminated as a result of a long period of inactivity. Figure 15 shows the minimum eigenvalue of the matrix $\mathbf{R}^T(k) \cdot \mathbf{R}(k) - \rho_1\mathbf{I}$ as a function of time that, similarly to the previous case, always remains positive, thus guaranteeing that the PE condition is satisfied.

The response of the proposed adaptive RBF controller is better compared to that of the other three MPC schemes. It is also important that the adaptive RBF controller achieves its goal while using less-aggressive control actions compared to those of the linear ones. Finally, the proposed MPC configuration manages to eliminate the modeling error faster than the adaptive linear MPC, especially during the rejection of the disturbance.

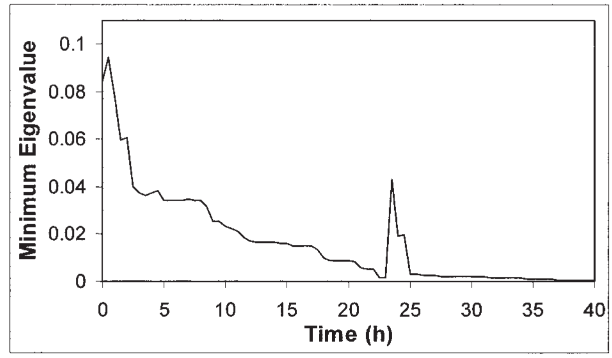


Figure 15. Case II: minimum eigenvalue of the matrix $\mathbf{R}^T(k) \cdot \mathbf{R}(k) - \rho_1\mathbf{I}$ with time.

Conclusions

This work presents a framework for integrating adaptive RBF models within the MPC structure. The RBF models are adapted online using the adaptive fuzzy means algorithm, which is able to update both the structure and the weights of an RBF network with time. In the proposed adaptive MPC configuration, at each time step an optimization problem is formulated that calculates the optimal values of the manipulated variables. Then the fuzzy means algorithm is used to adapt the RBF model. Special care has been taken to overcome the problem of simultaneous process control and identification by integrating the PE criterion in the optimization problem. This guarantees that there is always sufficient information to successfully adapt the weights between the hidden and the output layer.

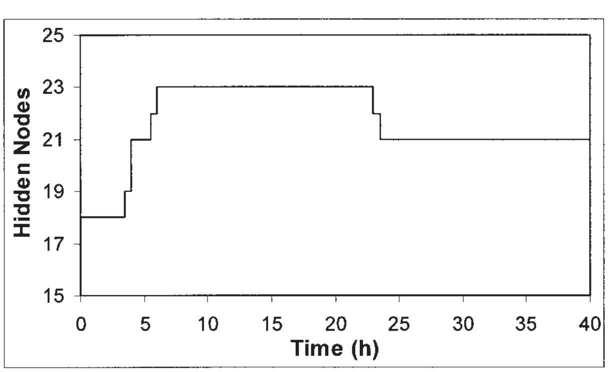


Figure 14. Case II: development of hidden layer structure with time.

The advantages of this algorithm are:

- It is suitable for nonlinear processes.
- It can take into account variations that might occur in the process with time.
- Using the PE criterion, it guarantees that no ill-conditioning problems are encountered in the calculation of the connection weights.

A possible shortcoming of the algorithm is the complexity of the nonlinear optimization problem that is formulated in each time step. Although a typical SQP algorithm solves the problem as long as small levels of excitation are applied, the authors are considering the development of a specialized algorithm to accelerate the solution of the nonlinear optimization problem.

The performance of the controller was tested in a case study, where the objective was to control a continuous digester, which is a complex nonlinear reactor used in the pulp and paper industry. Three more control schemes were implemented for comparison purposes, one based on a static RBF–MPC configuration and two based on adaptive and static linear MPC. The simulations involved changes in the operating region of the digester as well as rejection of disturbances caused by variations in the raw materials. It was shown that the proposed scheme, enhanced by its ability to eliminate the modeling errors, provides better control performance while at the same time it is more economical with respect to the control actions.

Literature Cited

1. Qin SJ, Badgwell TA. A survey of industrial model predictive control technology. *Control Eng Prac*. 2003;11:733-764.
2. Cutler CR, Ramaker DL. Dynamic matrix control—A computer control algorithm. Proc of the Joint Automatic Control Conference (JACC), San Francisco, CA; 1980.
3. Santos LO, Afonso PAFNA, Castro JAAM, Oliveira NMC, Biegler LT. On-line implementation of nonlinear MPC: An experimental case study. *Control Eng Prac*. 2001;9:847-857.
4. Genceli H, Nikolaou M. Design of robust constrained model-predictive controllers with Volterra series. *AIChE J*. 1995;41:2098-2107.
5. Maner BR, Doyle FJ III, Ogunnaike BA, Pearson RK. Nonlinear model predictive control of a simulated multivariable polymerization

reactor using second order Volterra models. *Automatica*. 1996;32: 1285-1301.
6. Norquay SJ, Palazoglu A, Romagnoli JA. Nonlinear model predictive control of pH neutralization using Wiener models. Proc of 13th International Federation of Automatic Control (IFAC) World Congress, San Francisco, CA; 1996.
7. Patwardhan RS, Lakshminarayanan S, Shah SL. Constrained nonlinear MPC using Hammerstein and Wiener models: PLS framework. *AIChE J*. 1998;44:1611-1622.
8. Grosman B, Lewin DR. Automated nonlinear model predictive control using genetic programming. *Comput Chem Eng*. 2002;26:631-640.
9. Hussain MA. Review of the applications of neural networks in chemical process control—Simulation and online implementation. *Artif Intell Eng*. 1999;13:55-68.
10. Moody J, Darken C. Fast learning in networks of locally-tuned processing units. *Neural Comput*. 1989;1:281-294.
11. Pottmann M, Seborg DE. A nonlinear predictive control strategy based on radial basis function models. *Comput Chem Eng*. 1997;21:965-980.
12. Bhartiya S, Whiteley JR. Factorized approach to nonlinear MPC using a radial basis function model. *AIChE J*. 2001;47:358-368.
13. Bhartiya S, Whiteley JR. Benefits of factorized RBF-based NMPC. *Comput Chem Eng*. 2002;26:1185-1199.
14. Maiti SN, Saraf DN. Adaptive dynamic matrix control of a distillation column with closed loop online identification. *J Process Contr*. 1995; 5:315-327.
15. Dougherty D, Cooper D. A practical model adaptive strategy for single loop MPC. *Control Eng Prac*. 2003a;11:141-159.
16. Dougherty D, Cooper D. A practical model adaptive strategy for multivariable model predictive control. *Control Eng Prac*. 2003b;11: 649-664.
17. Genceli H, Nikolaou M. New approach to constrained predictive control with simultaneous model identification. *AIChE J*. 1996;42: 2857-2868.
18. Shouche M, Genceli H, Vuthandam P, Nikolaou M. Simultaneous constrained model predictive control and identification of DARX processes. *Automatica*. 1998;34:1521-1530.
19. Vuthandam P, Nikolaou M. Constrained MPCI: A weak persistent excitation approach. *AIChE J*. 1997;43:2279-2288.
20. Nikolaou M. NSF/NIST workshop, process measurement and control: Industry needs, 6–8 March 1998, Sheraton New Orleans Hotel, Workshop on identification and adaptive control. *Comput Chem Eng*. 1998; 23:217-227.
21. Morningred JD, Paden BE, Seborg DE, Mellichamp DA. An adaptive nonlinear predictive controller. *Chem Eng Sci*. 1992;47:755-762.
22. Tsai PF, Chu JZ, Jang SS, Shieh SS. Developing a robust model predictive control architecture through regional knowledge analysis of artificial neural networks. *J Process Contr*. 2003;13:423-435.
23. Alexandridis A, Sarimveis H, Bafas G. A new algorithm for online structure and parameter adaptation of RBF networks. *Neural Netw*. 2003;16:1003-1017.
24. Gullichsen J, Fogelholm CJ. *Chemical Pulping*. Atlanta, GA: TAPPI; 1999.
25. Wisnewski P, Doyle FJ III, Kayihan F. Fundamental continuous pulp digester model for simulation and control. *AIChE J*. 1997;43:3175-3192.
26. Sarimveis H, Alexandridis A, Tsekouras G, Bafas G. A fast and efficient algorithm for training radial basis function neural networks based on a fuzzy partition of the input space. *Ind Eng Chem Res*. 2002;41:751-759.
27. Åström, KJ, Wittenmark B. *Adaptive Control*. 2nd Edition. Reading, MA: Addison–Wesley; 1995.
28. Rawlings JB. Tutorial overview of model predictive control. *IEEE Contr Syst*. 2000;20:38-52.
29. Wisnewski PA, Doyle FJ III. Control structure selection and model predictive control of the Weyerhaeuser digester problem. *J Process Contr*. 1998;8:487-495.
30. Amirthalingam R, Lee JH. Subspace identification based inferential control applied to a continuous pulp digester. *J Process Contr*. 1999; 9:397-406.
31. Doyle III FJ, Kayihan F. Reaction profile control of the continuous pulp digester. *Chem Eng Sci*. 1999;54:2679-2688.
32. Bhartiya S, Dufour P, Doyle FJ III. Thermal-hydraulic modeling of a continuous pulp digester. Proc of Pulp Digester Modeling and Control Workshop, Annapolis, MD; 2001.

Manuscript received Aug. 7, 2003, and revision received Jan. 12, 2005.