A PRIORITIZED MULTIOBJECTIVE MPC CONFIGURATION USING ADAPTIVE RBF NETWORKS AND EVOLUTIONARY COMPUTATION

Eleni Aggelogiannaki, Haralambos Sarimveis[1], Alex Alexandridis

School of Chemical Engineering, NTUA, 9 Heroon Polytechniou str. Zografou Campus, 15780 Athens, Greece

Abstract: In this work a prioritized multiobjective model predictive control configuration for nonlinear processes is proposed. The process is modeled by an adaptive radial basis function neural network so that modifications through time can be identified. The different control targets are formulated in a multiobjective optimization problem which is solved using a prioritized evolutionary algorithm. The request for adequate information in order to adapt the dynamics of the model is considered as the top priority objective. The algorithm is tested through the control of a pH reactor and the results are in favor of the proposed methodology.

Keywords: Model Based Control, Radial Base Function Networks, Adaptation, Multiobjective Optimization, Heuristic Searches

1. INTRODUCTION

During the last two decades, Model Predictive Control (MPC) has found great acceptance from both the academic and the industrial communities. Its success as control methodology is mainly due to its capability to incorporate all kinds of constraints, handle multi-input multi-output (MIMO) systems and be applied to processes where fundamental equations are not easily obtained (Qin and Badgwell, 2003; Camacho and Bordons, 1999). It is an optimal control method that selects a finite number of future moves in order to minimize an objective function, which considers both the deviation of the controlled variables from their set points and the control energy. Even though the theory of MPC is well matured, a number of difficulties arise, which mainly include the identification of the process and the on-line solution of the optimization problem that is formulated (Morari and Lee, 1999).

More precisely, the implementation of such a control method requires a model capable of describing the dynamics of the real process adequately. There is a great variety of models that have been incorporated in MPC methodologies and since the method is strictly associated with the use of computers, the most applicable are discrete time models. The wide spread of the MPC methodology is highly associated with its capability to handle non linear process models such as neural networks (Hussain, 1999; Henson, 1998). Lennox, *et al.* (2001) showed that utilization of neural networks instead of other models improved the performance of process control systems in a number of industrial applications. Among other neural network architectures, radial basis functions networks (RBF) seem to be quite popular for system modeling and control because of their relatively simple structure and their fast learning algorithms (Pottmann and Seborg, 1997).

Another significant issue that concerns the selection of the appropriate model is the time dependency of the process. Quite often, adaptation of the process model is necessary in order to preserve the accuracy of the model over time. However, in order to implement a successful adaptation in a closed loop operation, sufficient information should be available. A commonly used methodology to achieve this, is to

[1] Author to whom all correspondence should be addressed: hsarimv@chemeng.ntua.gr

introduce a persistent excitation constraint in the MPC configuration (Genceli and Nikolaou, 1996).

It is obvious that such a requirement is contrary to the control target to drive the manipulated variables to a steady state. However, this is not a unique case in control applications. Very often multiple and competitive targets should be addressed simultaneously. The most common method to confront such difficulties is to weight each control goal in a single objective function. The main disadvantage of this method is that the closed loop performance depends on the successful selection of the weights, which are supposed to assign to each objective the proper importance. Consequently, a time consuming procedure is required to select those weights. To avoid it, multiobjective configurations of MPC have been tested by Tyler and Morari (1999), who used integer variables to prioritize the different objectives and by Kerrigan and Maciejowski (2002), who recommended the formulation of a hierarchy of the objectives depending on their importance. The concept was also exploited by Aggelogiannaki, *et al.*, (2004), who considered adaptation of a linear finite impulse response (FIR) process model as a first priority objective.

As mentioned before, a key issue in the implementation of nonlinear MPC configurations is the efficiency of the algorithm which is used to solve the on-line optimization problem. Among other methodologies, the requirement of efficient optimization algorithms in control engineering has been confronted by using evolutionary computation. Evolutionary algorithms are search methods that have borrowed their principles from natural selection. Fleming and Purshouse (2002), review the applications of those algorithms in control engineering, while a number of publications recommend the implementation of such algorithms in MPC configurations (Sarimveis and Bafas, 2003; Potočnik and Grabec, 2002; Martinez, *et al.*, 1998). During the last two decades a number of evolutionary multiobjective optimization methods based on the idea of genetic algorithms and simulated annealing (Zitzler, *et al.*, 2000; Deb, *et al.*, 2000; Suman, 2004) have been developed. A significant advantage of those algorithms is that they can be used to optimize the different objectives simultaneously. In this way, a set of optimal solutions, known as the Pareto optimal set, can be obtained. Nevertheless, those algorithms ignore any knowledge relative to the importance of each objective and intend to find solutions that maintain the diversity and distribution of the optimal set. Consequently, those methodologies are more appropriate for design problems and not for on line optimization where a unique solution must be supplied the process.

Following the issues mentioned above, a new nonlinear MPC configuration is proposed in this work, which is based on an RBF neural network dynamic model of the real process. The weights that multiply the outputs of the hidden layer are adapted over time to follow the changes of the process. Then, a hierarchy of the control targets is formulated and optimized with a simulated annealing based multiobjective evolutionary algorithm. The top priority objective is the requirement of the persistent excitation as far as the outputs of the hidden layer are concerned, so that the successful adaptation of the weights is guaranteed. The target of this specially designed evolutionary algorithm is to produce a single solution and not the entire optimal Pareto front.

The rest of the article is formulated as follows. In section 2 the adaptive RBF-MPC configuration is described in more details. In section 3 the prioritized multiobjective simulated annealing based algorithm is presented. The overall methodology is tested through the control of a pH reactor in section 4. Finally, the conclusions derived from this work are outlined in the last section.

2. THE PROPOSED ADAPTIVE RBF-MPC CONFIGURATION

2.1 MPC and identification.

A typical MPC methodology based on an FIR process model predicts the value of the controlled variables over a finite horizon (*ph*) using *n* previous values of the manipulated variables. The optimization problem solved at each time point is of the following form:

$$\min_{\mathbf{u}(t|t)\ldots\mathbf{u}(t+ch-1|t)} \sum_{i=1}^{ph}\left\|\mathbf{W}\left(\hat{\mathbf{y}}(t+i|t)-\mathbf{y}^{sp}\right)\right\|_2^2+\sum_{i=0}^{ch-1}\left\|\mathbf{R}\Delta\mathbf{u}(t+i|t)\right\|_2^2 \quad (1)$$

$$\mathbf{u}_{\min} \le \mathbf{u}(t+i\,|\,t) \le \mathbf{u}_{\max}, \quad i=0,\ldots,ch-1 \quad (2)$$

$$\hat{\mathbf{y}}(t+i\,|\,t)=\sum_{j=1}^{n}\mathbf{G}_j(t\,|\,t)\cdot\mathbf{u}(t+i-j\,|\,t) + \mathbf{d}(t\,|\,t),\ i=1,\ldots,ph \quad (3)$$

where $\hat{\mathbf{y}}$ is the vector of the predicted values of the *cv* controlled variables at each future time point $t+i$, $\mathbf{u}$ is the vector of the *mv* manipulated variables, $\mathbf{G}_j$ $j=1,\ldots,n$ are the $cv \times mv$ FIR model coefficients, *ch* is the control horizon, $\mathbf{d}$ is the estimated disturbance at time point *t*, $\mathbf{W}$ and $\mathbf{R}$ are weight matrices. The optimization problem (1) is solved subject to Eq. (2) that poses upper and lower limits on the input variables. In order to simplify the problem, output or input move constraints are not considered in the MPC formulation. To satisfy the persistent excitation of the input variables and provide to the control scheme the capability of closed loop model adaptation, an additional constraint has been added (Genceli and Nikolaou, 1996; Åström and Wittenmark, 1995). This constraint requires the information matrix (that is formulated at each future time point $t+i$, $i=1,\ldots,ph$, as a function of the *m* last regression vectors $\boldsymbol{\varphi}(t)$) to be well conditioned (Eqs. (4), (5)):

$$\rho_1 \mathbf{I} < \sum_{k=0}^{m-1} \lambda^k \boldsymbol{\varphi}(t+i-k \mid t)\ \boldsymbol{\varphi}^T(t+i-k \mid t) \quad < \rho_2 \mathbf{I} \quad (4)$$

$$\boldsymbol{\varphi}^T(t+i-k \mid t) = \left[\mathbf{u}^T(t+i-k-1 \mid t) \ldots \mathbf{u}^T(t+i-k-n \mid t)\right] \quad (5)$$

where ρ_1, ρ_2 are positive real numbers, $\mathbf{I}$ is the $n \times n$ identity matrix, λ is the forgetting factor and $\mathbf{A} > \mathbf{B}$ means that the square matrix $\mathbf{A}-\mathbf{B}$ is positive definite.

2.2 Adaptive RBF neural network.

Radial basis function networks are simple in structure neural networks that consist of three layers, namely the input, the hidden and the output layer. Development of an RBF network based on input-output data includes the computation of the number of nodes in the hidden layer and the respective centers and the calculation of the output weights, so that the deviation between the predicted and the real values of the output variables over a set of training data is minimized. The method utilized to train neural networks in this work is based on a fuzzy partition of the input space and is described in details in Sarimveis *et al*. (2002). Due to the special RBF architecture, although the relationship between the input and the output variables in the produced model is nonlinear, the output of the model is a linear combination of the responses of the hidden nodes. Thus, the RBF network model can easily correct itself over time by adapting the output weights, using a linear adaptation technique such as the recursive least squares (RLS) with forgetting factor method (Åström and Wittenmark, 1995), that is adopted in this work. To incorporate a proper closed loop adaptation of the RBF network in an MPC configuration, the implementation of a persistent excitation constraint similar to the one introduced in the previous section, can also guarantee the collection of adequate on-line information. However, the regression vector now contains the outputs of the hidden layer and Eqs. (4), (5) should be converted as follows:

$$\sum_{k=0}^{m-1} \lambda^k \boldsymbol{\psi}(t+i-k \mid t)\ \boldsymbol{\psi}^T(t+i-k \mid t) \quad > \rho_1 \mathbf{I} \qquad (6)$$

$$\boldsymbol{\psi}^T(t+i-k \mid t) = \left[x_1\ (t+i-k \mid t)\ x_2(t+1-k \mid t) \ldots x_L(t+1-k \mid t)\right] \qquad (7)$$

where L is the number of hidden nodes and x_i is the response of the ith hidden node. The upper bound of Eq. (4) is satisfied anyway since there are upper limits for the values of the input variables and sequentially for the outputs of the hidden layer. The request to keep the outputs of the hidden layer excited is applied only on the next time step of the prediction horizon, in order to reduce the computational burden of the algorithm. That means that we apply Eq. (6) only for i=1.

2.3 The proposed prioritized MPC configuration.

In this subsection the prioritized MPC configuration is presented in details. The proposed formulation aims at the simultaneous identification of the process changes through the adaptation procedure, and the improvement of the closed loop response. Furthermore the tuning effort required for the typical design of such a controller is reduced. In order to achieve this, a hierarchy of the different control goals is formulated instead of weighting them in a single objective. This hierarchy is of descending order so that the high priority objectives are optimized first.

The first modification of the proposed configuration concerns the persistent excitation. The idea of relaxing the hard constraint of keeping inputs excited firstly suggested by Genceli and Nikolaou (1996) is further exploited. In this work the constraint is transformed to an optimization problem by introducing an additional variable μ in the persistent excitation equation (6), which is modified as follows:

$$\mathbf{M} = \sum_{k=0}^{m-1} \lambda^k \boldsymbol{\psi}(t+1-k \mid t)\ \boldsymbol{\psi}^T(t+1-k \mid t) - (\rho_1 - \mu) \cdot \mathbf{I} > \mathbf{0} \qquad (8)$$

The value of variable μ is minimized with respect to the next future values of the manipulated variables subject to the constraints of the initial optimization problem (Eq. (2)) and the modified persistent excitation constraint (Eq. (8)). This optimization problem is regarded as the top priority one and is solved first:

$$\mu^* = \min_{\mathbf{u}(t|t),\ldots,\mathbf{u}(t+ch-1),\mu} f_1, \qquad f_1 = \mu \qquad (9)$$

This modification avoids infeasibility issues related to the choice of parameter ρ_1 that may arise when the requirement is expressed as a hard constraint. The proposal of Genceli and Nikolaou (1996) to add the parameter μ in the objective function of Eq. (1) can also avoid infeasibilities but has the drawback that an extra tuning effort is necessary to assign an appropriate weight to this control target.

Then the objective function of Eq. (1) is separated in cv new functions, one for each controlled variable. These objective functions weigh the deviation of the particular variable from its set point value and the control energy of the manipulated variables:

$$\min_{\mathbf{u}(t|t),\ldots,\mathbf{u}(t+ch-1|t)} f_{j+1} \qquad (10)$$

$$f_{j+1} = \sum_{i=1}^{ph} \left(\hat{y}_j(t+i \mid t) - y_j^{sp} \right)^2 + \sum_{i=0}^{ch-1} \left\| \mathbf{R}_j \Delta \mathbf{u}(t+i \mid t) \right\|_2^2 \quad (11)$$

for j=1,…, cv , where the order depends on the importance of each variable. Each optimization problem is solved subject to constraint (Eq. (2)), but should also satisfy the persistent excitation criterion (Eq. (8)) for $\mu=\mu^*$ and should not deteriorate previous in rank objectives. For each controlled variable j, one (if the system is square, cv=mv) or a group of manipulated variables can be associated based on the experience on the process. Tuning of the move suppression weights $\mathbf{R}_j$ is performed by assigning large weights to the inputs associated to the next in rank controlled variables. In this way control energy is preserved for all the controlled variables.

3. MULTIOBJECTIVE OPTIMIZATION ALGORITHM BASED ON THE PRINCIPLES OF SIMULATED ANNEALING

The multiobjective MPC configuration presented in the previous section requires the calculation of a unique solution at each time step and not the entire Pareto front. This section describes the new stochastic algorithm that is proposed to approximate this unique solution of the multiobjective optimization problem. The algorithm is a random search methodology based on the simulated annealing principle that also considers the known hierarchy of objectives (Eq. (12)).

$$\mathbf{f}(\mathbf{z}) = \left[f_1(\mathbf{z}),\ldots,f_{cv+1}(\mathbf{z})\right]^T, \quad \mathbf{z} = \left[\mathbf{u}^T(t\,|\,t),\ldots,\mathbf{u}^T(t+ch-1\,|\,t),\ \mu\right]^T \tag{12}$$

To improve its performance the optimal value found at each time step is used as the initial point for the next optimization problem. Also, a non-uniform mutation operation is applied to accelerate the convergence. The parameters of the algorithm are the initial temperatures corresponding to the different objectives $T_{in,j}$, the number of iterations *maxiter* and the rate that the temperatures change r_b, which is positive and less than 1. Finally, an additional penalty parameter b is used to incorporate the persistent excitation criterion in the first objective. Thus, the top priority objective is augmented according to the following equation:

$$F_1(\mathbf{z}) = f_1(\mathbf{z}) + b\cdot\max\left\{0,-g(\mathbf{z})\right\} \tag{13}$$

where $g(\mathbf{z})$ is the minimum eigenvalue of the symmetric matrix $\mathbf{M}$ of Eq. (8).

$$g(\mathbf{z}) = \min\left(eig(\mathbf{M})\right) \tag{14}$$

The detailed steps of the algorithm are given below.

1) Define the parameters of the algorithm and the search space. Set $T_j=T_{in,j}$.

2) Calculate the values of all objective functions for the initial solution $\mathbf{z}_1$. Set $iter=1$.

3) If $iter>maxiter$ terminate the algorithm. The optimal solution is $\mathbf{z}_{maxiter}$.

4) Produce a new random solution according to the non-uniform mutation:
   FOR $l=1$ to $mv\cdot ch+1$
   Generate a random number $0\leq rand \leq 1$
   Generate a random binary number *bin*
   IF $bin=0$
   $$z_{\text{new}}(l)=z_{iter}(l)+(z_{\max}(l)-z_{iter}(l))\cdot rand\cdot e^{-2\cdot iter/maxiter} \tag{15}$$
   ELSE IF $bin=1$
   $$z_{\text{new}}(l)=z_{iter}(l)+(z_{iter}(l)-z_{\max}(l))\cdot rand\cdot e^{-2\cdot iter/maxiter} \tag{16}$$
   END
   END

5) Calculate the probabilities of the new solution to survive:

$$DF_1 = F_1(\mathbf{z}_{new}) - F_1(\mathbf{z}_{iter}) \tag{17}$$

$$DF_j = f_j(\mathbf{z}_{new}) - f_j(\mathbf{z}_{iter}), \quad j=2,\ldots,cv+1 \tag{18}$$

$$pr_j = \exp\left[-DF_j / T_j\right], \quad j=1,\ldots,cv+1 \tag{19}$$

6) Produce $cv+1$ random numbers r_j, $j=1,\ldots,cv+1$ between 0 and 1

   IF
   $$\left\{\begin{matrix}(DF_1<0 \text{ OR } r_1 < pr_1) \text{ AND} \\ (DF_2 < 0 \text{ OR } r_2 < pr_2) \text{ AND} \\ \vdots \\ (DF_{cv+1} < 0 \text{ OR } r_{cv+1} < pr_{cv+1})\end{matrix}\right\}$$
   THEN $\mathbf{z}_{iter+1}=\mathbf{z}_{new}$
   ELSE $\mathbf{z}_{iter+1}=\mathbf{z}_{iter}$
   END

7) Set $T_j = T_j\cdot r_b$, $j=1,\ldots,cv+1$ and $iter=iter+1$

8) Return to step 3.

Remark. In order to introduce to the algorithm the prioritization of the control targets, a different initial temperature is used in the simulated annealing concept for each objective. If the temperatures are given in a ascending order then the probabilities of accepting a worse than the current solution are higher for the objectives of less priority.

4. APPLICATION

4.1 Description of the process.

The proposed MPC configuration is tested through a pH reactor. This process is highly nonlinear and is often considered as a benchmark problem for nonlinear control methodologies. (Krishnapura and Jutan, 2000; Nie, *et al.*, 1996). A stream of acetic acid of flow rate F_1 and a stream of sodium hydroxide base of flow rate F_2 enter a continuous stirred tank reactor, as it is showed in Fig. 1. The concentrations of the reactants in the inlet streams are C_1 and C_2 respectively. The controlled variable of the process is the pH of the outlet stream and the manipulated variable is the base stream flow, F_2. The dynamic behavior of the process is described by the material balances as they are given by Eqs. (20), (21).

$$V\frac{dw_1}{dt} = F_1C_1 - (F_1 + F_2)w_1 \qquad (20)$$

$$V\frac{dw_2}{dt} = F_2C_2 - (F_1 + F_2)w_2 \qquad (21)$$

where V is the volume of the mixture inside the reactor and is supposed to remain steady and w_1 and w_2 are the concentrations of the acid and the base respectively in the outlet stream. The equilibrium relationships that hold for acetic acid and water associate the value of pH with the concentrations of the reactants according to the following equation:

$$w_2 + 10^{-\mathrm{pH}} - 10^{\mathrm{pH}-14} - \frac{w_1}{1+10^{\mathrm{p}K_a - \mathrm{pH}}} = 0 \qquad (22)$$

where pK_a=-log$_{10}K_a$ and K_a is the equilibrium constant for acetic acid. The process parameter values can be found in Krishnapura and Jutan (2000). The nonlinear character of the process is apparent, when the steady state behavior is studied (Fig. 2).

4.2 Identification of the process using neural network.

For the development of the RBF network 3000 input-output data points were generated, by assigning random value between 300 and 530 l/min to the input variable. 2500 pairs were used to train the network while the rest of them were used for validation. The differential equations were solved using the ode45 function of Matlab. Data were collected using a sampling interval of 1 min. The resulting network consisted of L=51 nodes in the hidden layer. The weights were subsequently adapted on line using the described method.

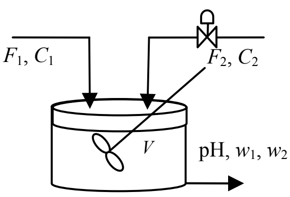

Fig. 1. The continuous stirred tank pH reactor.

4.3 Application of the proposed MPC configuration-Results

The proposed algorithm is compared with a conventional MPC algorithm that uses an FIR linear model to predict the future values of pH. The parameters of both controllers are summarized in Table 1, where R, W refer to the scaled values of the input and output variables. The parameters of the optimization algorithm can be found in Table 2.

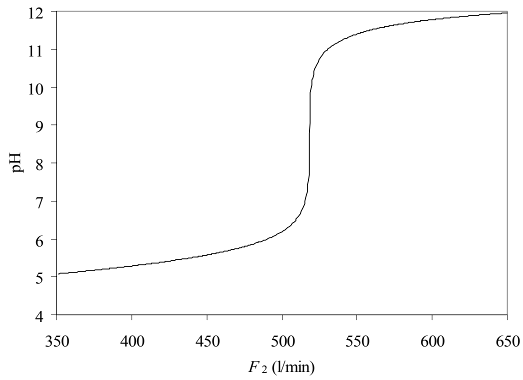

Fig. 2. Steady state pH curve by changing the flow rate of the base stream.

Table 1 Parameters of the two MPC configurations

Parameter	Proposed MPC	Conventional MPC
ph	15	15
ch	5	5
n	10	10
m	60	-
λ	1	-
ρ_1	10^{-5}	-
R	0.1	0.1
W	-	1
u_{min}	460	460
u_{max}	515	515
μ_{min}	0	-
μ_{max}	10^{-5}	-

Table 2 Multiobjective optimization algorithm parameters

Parameter	Value
T_{in}	$\begin{bmatrix}10^4 & 10^{10}\end{bmatrix}^T$
r_b	0.99
$maxiter$	3500
Cv	1
mv	1
B	10

To evaluate the proposed configuration a rapid change of a process parameter is enforced. Thus, at time point 10 the acid equilibrium constant is reduced from 1.76 10^{-5} to 10^{-6}. The set point of the process is pH=7. The responses of both control methodologies are depicted in Fig. 3. It is obvious that the implementation of a linear non adaptive model is not adequate to follow such changes. On the contrary the non linear adaptive controller manages to reject the disturbance and drive the system to pH=7 within 20 time steps. The requirement of the persistent excitation is also satisfied as Fig. 3 denotes, and an unstable behavior of pH is avoided. The computational time for solving the multiobjective optimization problem using Matlab 6.5 in a Pentium IV 1400 MHz machine was on average 12s which is much less that the sampling time of 1 min that was utilized.

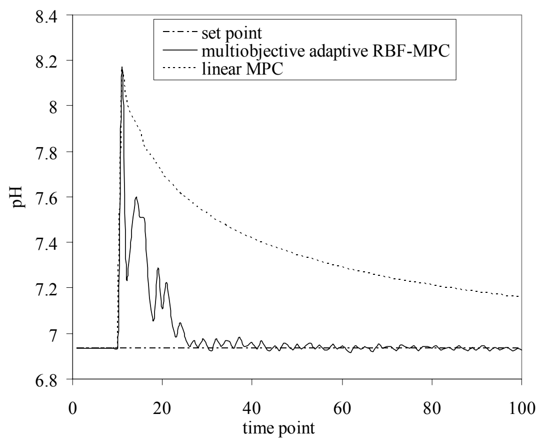

Fig. 3. Response of the two MPC configurations in a rapid change of the acid equilibrium constant K_a.

CONCLUSIONS

A new MPC configuration was suggested for the control of highly nonlinear and relatively slow systems, which can also be time varying. Using an adaptive RBF network simultaneously with the multiobjective configuration, closed loop identification of the process is successfully performed, mainly due to the high priority that is assigned to the persistent excitation requirement. Furthermore, the capabilities of evolutionary algorithms concerning the solution of the formulated multiobjective optimization problem were illustrated. A new algorithm based on simulated annealing with different initial temperatures for each objective function was introduced in order to obtain an optimal solution that satisfies the known hierarchy of objectives.

REFERENCES

Aggelogiannaki E., H. Sarimveis and G. Bafas (2004). Multiobjective constrained MPC with simultaneous closed loop identification for MIMO processes. *Proceedings of the ESCAPE-6 Conference,* 523-528.

Åström K.J. and B. Wittenmark (1995). *Adaptive Control*. Addison-Wesley, Massachusetts.

Camacho E.F. and C. Bordons (1999). *Model Predictive Control*. Springer-Verlag, London.

Deb K., A. Pratap, S. Agarwal and T. Meyarivan (2000). A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II. *KanGAL Report No. 200001*, Kanpur.

Fleming P.J. and R.C. Purshouse (2002). Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice*, **10**, 1223-1241.

Genceli H. and M. Nikolaou (1996). New approach to constrained predictive control with simultaneous model identification. *AIChE Journal*, **42**, 2857-2868.

Henson M.A (1998). Nonlinear model predictive control: current status and future directions. *Computers & Chemical Engineering*, **23**, 187-202.

Hussain M. A. (1999). Review of the applications of neural networks in chemical process control-simulation and on line implementation. *Artificial Intelligence in Engineering,* **13**, 55-68.

Kerrigan E.C. and J.M. Maciejowski (2002). Designing model predictive controllers with prioritized constraints and objectives. *Proceedings of IEEE Conference on the Computer Aided Control System Design*, 33-38.

Krishnapura V.G. and A. Jutan (2000). A neural adaptive controller. *Chemical Engineering Science*, **55**, 3803-3812.

Lennox B., G. A. Montague, A. M. Frith, C. Gent and V. Bevan (2001). Industrial application of neural networks-an investigation. *Journal of Process Control*, **11**, 497-507.

Martinez M., J. S. Senent and X. Blasco (1998). Generalized predictive control using genetic algorithms. *Engineering Applications of Artificial Intelligence*, **11**, 355-367.

Morari M. and J. Lee (1999). Model predictive control: past, present and future. *Computers & Chemical Engineering*, **23**, 667-682.

Nie J., A.P. Loh and C.C. Hang (1996). Modeling pH neutralization processes using fuzzy-neural approaches. *Fuzzy Sets and Systems*, **78**, 5-22.

Potočnik P. and I. Grabec (2002). Nonlinear model predictive control of a cutting process. *Neurocomputing*, **43**, 107-126.

Pottmann M. and D. E. Seborg (1997). A nonlinear predictive control strategy based on radial basis function models. *Computers & Chemical Engineering*, **21**, 965-980.

Qin S.J. and T.A. Badgwell (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, **11**, 733-764.

Sarimveis H. and G. Bafas (2003). Fuzzy model predictive control of non-linear processes using genetic algorithms. *Fuzzy Sets and Systems*, **139**, 59-80.

Sarimveis H., A. Alexandridis, G. Tsekouras and G. Bafas (2002). A fast and efficient algorithm for training radial basis function neural networks based on a fuzzy partition of the input space. *Industrial Engineering Chemistry Research*, **41**, 751-759.

Suman B. (2004). Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem. *Computers & Chemical Engineering*, **28**, 1849-1871.

Tyler M.L. and M. Morari (1999). Propositional logic in control and monitoring problems. *Automatica*, **35**, 565-582.

Zitzler E., K. Deb and L. Thiele (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation,* **8**, 173-195.