Brazilian Journal
of Chemical
Engineering

ISSN 0104-6632
Printed in Brazil
www.abeq.org.br/bjche

Vol. 33, No. 01, pp. 177 - 190, January - March, 2016
dx.doi.org/10.1590/0104-6632.20160331s00003508

PREDICTIVE CONTROL OF A BATCH POLYMERIZATION SYSTEM USING A FEEDFORWARD NEURAL NETWORK WITH ONLINE ADAPTATION BY GENETIC ALGORITHM

A. Cancelier[1], C. A. Claumann[2], A. Bolzan[2] and R. A. F. Machado[2*]

[1]Departament of Chemical Engineering, Federal University of Santa Maria, UFSM,
Av. Roraima 1000, Bairro Camobi, CEP: 97.105-900, Santa Maria - RS, Brazil.
E-mail: adriano.cancelier@ufsm.br
[2]Program in Chemical Engineering, Dept of Chemical Engineering,
Federal University of Santa Catarina, UFSC, Trindade Campus, Trindade,
CEP: 88040-900, P.O. Box: 476, Florianópolis - SC, Brazil.
Phone: + 55 (48) 3721-2431
[*]E-mail: ricardo.machado@ufsc.br
E-mail: caio@enq.ufsc.br; abolzan@enq.ufsc.br

(Submitted: May 16, 2014 ; Revised: March 15, 2015 ; Accepted: March 17, 2015)

Abstract - This study used a predictive controller based on an empirical nonlinear model comprising a three-layer feedforward neural network for temperature control of the suspension polymerization process. In addition to the offline training technique, an algorithm was also analyzed for online adaptation of its parameters. For the offline training, the network was statically trained and the genetic algorithm technique was used in combination with the least squares method. For online training, the network was trained on a recurring basis and only the technique of genetic algorithms was used. In this case, only the weights and bias of the output layer neuron were modified, starting from the parameters obtained from the offline training. From the experimental results obtained in a pilot plant, a good performance was observed for the proposed control system, with superior performance for the control algorithm with online adaptation of the model, particularly with respect to the presence of off-set for the case of the fixed parameters model.
Keywords: Predictive control; Neural network; Genetic algorithm; Suspension polymerization.

INTRODUCTION

Control systems based on models have proven to be efficient in chemical processes, especially in cases with strong interactions between input and output, high dead time, and physical constraints on the variables (García *et al.*, 1989). For the purpose of application of control strategies, mathematical models of processes can be presented in different formats, provided they can be used to obtain useful, significant and reliable predictions of the process behavior.

It is now well established that phenomenological models typically provide a more accurate description of the process, especially for extrapolation, and empirical models are easier to obtain and manipulate during online applications in real time, especially when obtaining experimental data is facilitated (Vieira *et al.*, 2003; Cubillos *et al.*, 2007; Janakiraman *et al.*, 2013). For this reason, some applications require an optimization/adaptation of the model developed and eventually the use of hybrid structures, which take into account empirical knowledge plus phenomeno-

*To whom correspondence should be addressed logical knowledge, may be considered. In order to overcome difficulties in developing mechanistic models, empirical models based on neural networks are being used for the purpose of modeling and process optimization, as well as for the construction of control strategies (Zhang, 2003; Ławryńczuk, 2013). Especially when the system treated has strong nonlinearities, neural networks have been widely applied for identification and modeling (Ng and Hussain, 2004; Qiao and Han, 2012).

Empirical modeling based on Artificial Neural Networks (ANNs) has been widely used in chemical processes, especially for identifying systems or using predictive control techniques. In this context, the models based on ANNs are presented as a powerful tool for modeling static and dynamic systems, with large nonlinearities and high dead-time, mainly due to two fundamental qualities: rapid adaptability and intrinsic approach (De Souza Jr *et al.*, 1996; Krothapally and Palanki, 1997; Yu and Yu, 2003; Zhang, 2003; Ng and Hussain, 2004; Hosen *et al.*, 2011).

Polymer production has grown considerably, finding applications in several areas, from the simple use in the manufacture of packaging and utensils to specific applications, such as engineering polymers, polymers with textile properties, and polymers with optical properties, among others. Depending on the application, the polymer needs to meet specific quality standards, among which is the molecular weight distribution, whose variation directly affects its characteristics, such as mechanical, thermal and flow properties (Takamatsu *et al.*, 1988; Crowley and Choi, 1998).

Considering that polymer properties such as molecular weight, polydispersity index, and morphological characteristics are not easy to be obtained online in a polymerization system, models for estimation of these properties are required for the implementation of efficient control and monitoring systems (Prasad *et al.*, 2002; Vieira *et al.*, 2003; Bindlish and Rawlings, 2003; Santos *et al.*, 2008). For polymerization chain reactions, temperature and initial initiator concentration has high influence on the reaction kinetics and the polymer molecular weight distribution, with a direct effect on polymer properties (Sacks *et al.*, 1973; Erdogan *et al.*, 2002; Hosen and Hussain, 2012).

Thus, considering that changes in the operating conditions of the polymerization system have a significant influence on the polymer final properties, different control techniques have been developed for this application. Given the characteristics of these processes, usually conducted in batch systems, classic controllers for fixed parameters generally do not produce satisfactory results. For this reason, research is focused on the development of techniques based on linear and nonlinear predictive control, adaptive algorithms and, more recently, control algorithms using rule-based computer techniques or expert systems. Efficient controllers, together with optimization algorithms, provide an important tool to determine the operating conditions necessary to produce the polymer with the desired characteristics, and to adjust them during the polymerization time (Alvarez and Odloak, 2012; Hosen and Hussain, 2012).

Given the polymerization reactions characteristics (large nonlinearities, constraints on their operational variables, multiple stationary states for continuous systems and lack of a stationary state for batch systems), adaptive control techniques have been developed and applied mainly for predictive control where empirical models are used. Among these are models based on neural networks trained offline, because these networks are developed with experimental data obtained under certain operational conditions of the system, sometimes in open loop, which are rarely repeated during the closed loop control. Deviations that occur in the system, which are not predicted in the model, cause various control problems, such as high overshoot, offset, and others. In this case, adaptive techniques can be used to adjust the weights of the network and prevent such deviations from occurring (Zeybec *et al.*, 2003; Ng and Hussain; 2004; Marcolla *et al.*, 2009, Hosen *et al.*, 2011).

The goal of this study was to develop an algorithm for the temperature control of styrene suspension polymerization in a jacketed batch reactor. The control was performed by manipulating two variables, the steam flow, used for heating the reactor, and the water flow, for cooling. To this end, a predictive controller using an empirical model of the process has been developed. As an empirical model, a feedforward neural network was used with online adaptation of the parameters (weights and bias) by means of an optimization system based on the genetic algorithm technique.

METHODOLOGY

Experimental Unit

The experiments were conducted in the Laboratory of Process Control at the Department of Chemical Engineering and Food Engineering at the Federal University of Santa Catarina (EQA/CTC/UFSC). The reaction pilot unit (Figure 1) consisted of a stainless steel jacketed reactor (Suprilab Ltda), with a capacity of 5 liters and maximum pressure of 15 kgf/cm^2, equipped with a stirring system with a double turbine impeller, centered and which extends to the base of the reactor. The thermal exchanges necessary to the reactor are performed by a plate heat exchanger with cross currents, Alfa Laval brand, and two pneumatic valves of equal percentage of air-open/fail-close, brand Badge Meter Inc. Also part of the system is an Ecil brand type *J* thermocouple in conjunction with an amplifier/transmitter, as well as a reservoir of nitrogen gas. Steam is supplied by a boiler with electric heating, SIMILI brand, type SIM-HE, with a steam production capacity of 100 kg/h and pressure of 8.4 kg/cm^2.

The heat exchange system is started by activating the centrifugal pump (6), which changes the reactor jacket pressure to approximately 2 kgf.cm^{-2}. Valve 8 is responsible for controlling the steam flow in the plate heat exchanger (2), through which the hot current for heating the reactor is generated (1). Valve 7 is held closed and only the hot current flows in the reactor, receiving more energy at each passage through the heat exchanger, reaching in this way high temperatures in a relatively short time interval. Opening valve 7 the pressure reduces in the reactor jacket, as well as in the whole circulation line, and then the cold circulation starts.

Chemical Reaction

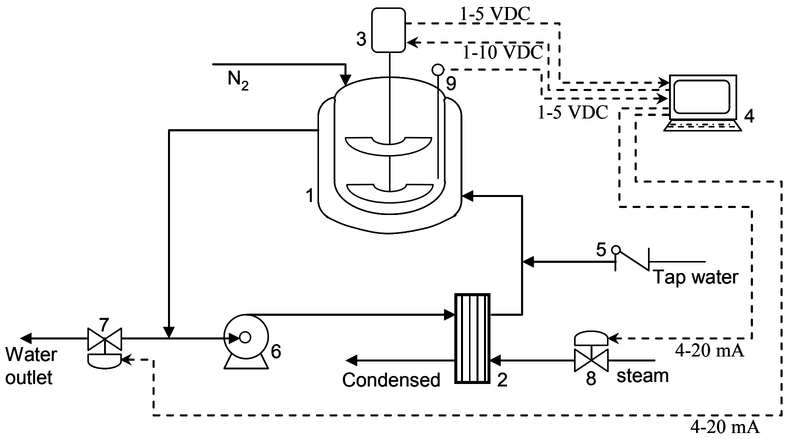
In every reaction, 1.2 L of monomer was used as the dispersed phase and 2.8 L was used as the continuous medium (distilled water). Poly(vinylpyrrolidone), (PVP K-90, Graft Corp.) was used as a suspension agent at a concentration of 1.0 g.L^{-1} in relation to the continuous phase. As initiator, benzoyl peroxide - BPO (Sigma-Aldrich) was used, at concentrations in a range between 1.262x10^{-3}gmol$_{\mathrm{BPO}}$.gmol$^{-1}_{\mathrm{Styrene}}$ and 1.893x10^{-3} gmol$_{\mathrm{BPO}}$.gmol$^{-1}_{\mathrm{Styrene}}$. The styrene (monomer) was supplied by Termotécnica Ltda. All reagents used in the experiments were used as received. The stirring frequency was kept constant during the reaction time, at 300 rpm.

Process Empirical Model (FANN)

For the batch styrene polymerization system representation/model, a Feedforward Artificial Neural Network (FANN) was used, fully interconnected with three layers whose mapping between the output and the inputs can be described by Equation (1),

$$y_p(k+1) = f\begin{bmatrix} y(k), y(k-1); \\ U_1(k-d_1), U_2(k-d_2) \end{bmatrix} \qquad (1)$$

where *f* denotes a nonlinear mapping considering all operations performed by the network in the input data. $U_1(k\text{-}d_1)$ and $U_2(k\text{-}d_2)$ are, respectively, entries 1 and 2 of the process, with d_1 and d_2 representing the dead-time regarding each input variable. *y(k)* and *y*(*k-1*) are, respectively, the actual values of the outputs, measures of the process at instants *k* and *k*-1 (Multiple Input / Single Output). $y_p(k+1)$ is the value predicted by the neural network at instant *k*+1. It is worth mentioning in Equation (1) the distinction between the predicted value, y_p, for which the subscript *p* was used, and the measured values, without any identification in particular.



1. Jacketed reactor,
2. Plate heat exchanger,
3. Agitation system (motor, frequency controller and digital tachometer),
4. Computer equipped with AD/DA data acquisition board,
5. Retention valve,
6. Centrifugal pump,
7. Pneumatic valve: reactor jacket circulation water outlet flow control,
8. Pneumatic valve: steam flow control,
9. Temperature sensor.

Figure 1: Scheme of the polymerization pilot unit.

The neural network input variables were based on the usual methodology of identification systems used in process control. Thus, the input variables were chosen from identification tests by disturbances in the cold water and steam introduced into the system without reaction. Variables $y(k)$ and $y(k\text{-}1)$ were chosen by experimental observations that the system has underdamped behavior. In this case, at least two delays are necessary for correct system identification: the dead time between temperature measurement and the changes caused by the cold water and vapor.

In the polymerization system studied in this research work, the output variable of interest is the reaction temperature, which is controlled by manipulating the steam flow control valve (8-Figure 1), U_1, and the jacket hot water discharge control and cold water supply (7-Figure 1) valves, U_2. The system is subject to restrictions on the opening valves, with 1 volt indicating that it is completely closed and 5 volts indicating that is 100% open. Restrictions in temperature obey the physical limits of the system. Thus, the FANN used has 3 layers with 4 neurons in the input layer, 5 neurons in the intermediate layer and one neuron in the output layer, all fully interconnected. It is noteworthy that the architecture was defined by evaluating the behavior of the objective function (quadratic average error) for a training and testing group. The network with fewer neurons in the middle layer was adopted, which allowed the best approach, both for training and for testing (offline evaluation of the FANN from a set of test data and online evaluation of the controller using the FANN obtained in training). The activation function used for the intermediate layer neurons is the hyperbolic tangent function, and for the output layer a linear function was used. Thus, the weights of the output layer can be estimated using the least squares method, as described in the next section.

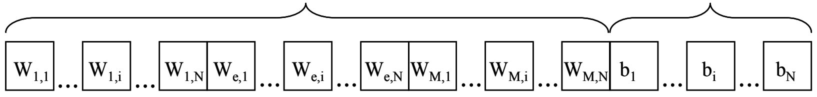
The results presented in the FANN Training Offline section show that only one hidden layer was enough to identify the system, showing that there is no benefit to increasing the number of hidden layers due to the possibility of overfitting. Additionally, it is desirable to minimize the number of weights to be adapted in real time.

FANN Offline Training

For FANN training, a genetic algorithm (GA) with real coding was used in conjunction with the least squares method. Thus, the result is a hybrid algorithm between a stochastic and a deterministic method, respectively. Both are based originally on the functions available in the MatLab application® Toolbox (Houck *et al.*, 2003). In this case, the weights and bias of neurons in the hidden layer were determined using the GA, while the weights and bias of the output layer neurons were, in sequence, determined by the least squares method, in a serial configuration. The advantage of this configuration is to decrease the size of the problem by weights with GA estimation, coupled with the fact that there is an optimum solution in terms of minimizing the sum of the quadratic error obtained directly by the least squares method. This methodology provided better results than the isolated application of the genetic algorithm.

The definition of the GA operators followed that presented by Claumann (1999) for training a FANN with similar architecture. Coding of FANN weights in the shape of a chromosome is shown in Figure 2. Therefore, the number of parameters to be determined by GA, which corresponds to the number of genes in the chromosome, is identical to the total weight plus the number of bias of the FANN intermediate layer. For training, 300 generations were used for a population of 150 individuals and a search interval of the weights and bias of the intermediate layer of -6 to +6.

To perform offline FANN training, a data set must initially be acquired. These were obtained with the open loop system and the reactor containing only water, without polymerization reaction, for a sampling time of 10 s. The data are presented in Figure 3. In the situation with no reaction it is possible to apply arbitrary disturbances to maximize the amount of information obtained in identification of the process (Fernandes and Lona, 2005). However, during the reaction, the temperature profile should be followed and therefore sharp changes of temperature, according to the variations of the control actions, are not allowed under penalty of the polymer produced not achieving the desired properties and, in the worst case, destabilization of the reaction.


Weights
Bias
$W_{1,1}$ … $W_{1,i}$ … $W_{1,N}$ $W_{e,1}$ … $W_{e,i}$ … $W_{e,N}$ $W_{M,1}$ … $W_{M,i}$ … $W_{M,N}$ b_1 … b_i … b_N

N: Number of neurons in the hidden layer
M: Number of inputs

Figure 2: Coding of the weights of the FANN intermediate layer in the form of a chromosome.

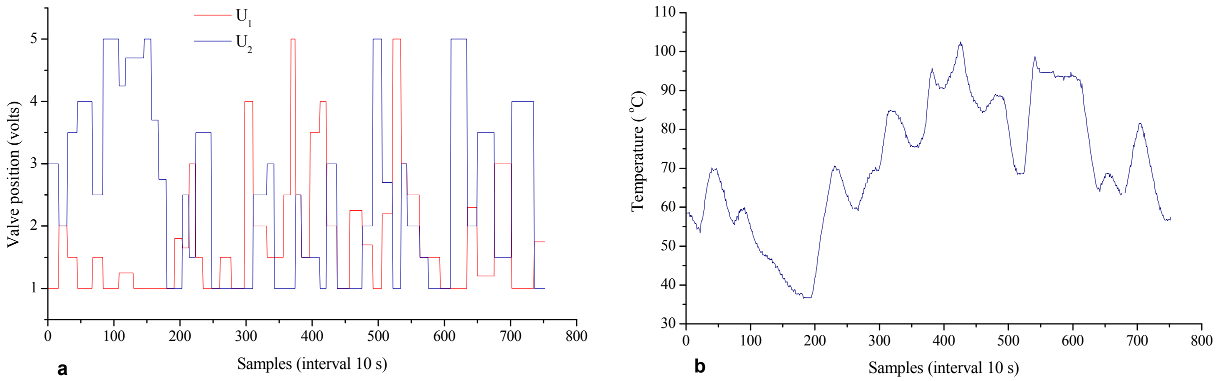


Figure 3: Standards for FANN offline training. (a) Step disturbances applied to the control valves. (b) Response of the reactor temperature to the disturbances applied.

Next, the collected data must be converted into patterns for network training. Table 1 shows the formation of a pattern for the network inputs, considering the temporal displacement of the components to a dead time d_1 of 4 sampling times and d_2 of 3 sampling time, for the prediction output variable (in this case the system temperature) at (k+1). The dead time was identified experimentally.

Table 1: Formation of the network input patterns for training.

Instant	$y(k+1)$ Target	$y(k)$	$y(k-1)$	$U_1(k)$	$U_2(k)$
1	$y(1)$	$y(1)$	$y(1)$	$U_1(1)$	$U_2(1)$
2	$y(2)$	$y(2)$	$y(2)$	$U_1(2)$	$U_2(2)$
3	$y(3)$	$y(3)$	$y(3)$	$U_1(3)$	$U_2(3)$
4	$y(4)$	$y(4)$	$y(4)$	$U_1(4)$	$U_2(4)$
5	$y(5)$	$y(5)$	$y(5)$	$U_1(5)$	$U_2(5)$
6	$y(6)$	$y(6)$	$y(6)$	$U_1(6)$	$U_2(6)$

Figure 4a shows the comparison between the actual patterns used for network training and predicted values. The result for the application of this network to the test patterns group is presented in Figure 4b. It is noted that the methodology achieved excellent performance across the working range of the data used for training and testing, and represented well the system dynamics.

Comparing the experimental patterns and those obtained by the network, after training, there was a quadratic average error of 2.8945x10^{-4} and a coefficient of determination of 0.99934. For the group of testing data, these values were 4.1815x10^{-4} and 0.99916, respectively. The behavior of the results presented in Figure 4 also suggests that there was no overfitting.

Online Adaptation of FANN Weights

The data used as standards for FANN offline training, for experimental simplicity, were obtained for the system with no chemical reaction using only water as the fluid to fill the reactor volume. Logically, this model can present deviations if used to represent the system dynamics while conducting the polymerization.

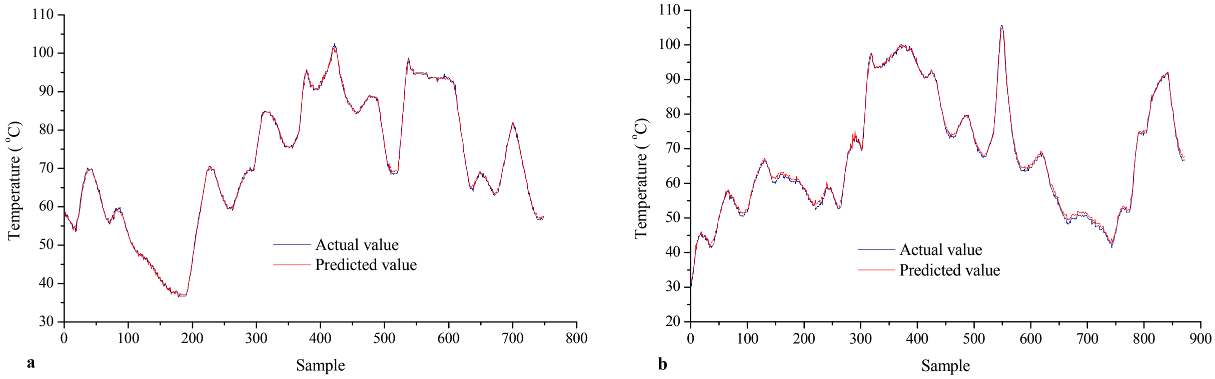


Figure 4: Comparison between actual and predicted values by the network after training. (a) Group of data used in training. (b) Group of test data.

The dynamic modification occurs mainly for three reasons: the reaction ΔH (for the styrene polymerization ΔH_r = -70 kJ mol^{-1} (Chen, 2000; Billmeyer, 1984), that depends on the propagation rate), auto acceleration of the reaction due to the gel effect (Huang and Lee, 1989), and the change in the heat capacity of the reaction medium since, instead of just water, there are polymer/monomer particles in suspension with variation of this ratio during the progress of the reaction.

Considering the network used as a prediction model for the predictive controller, the performance will be better if trained on a recurring basis, since is how it is used in the algorithm control. For the online FANN training, GA with real coding was used, due to its applicability in nonlinear problems and ease of generalization. The use of GA parameter values must be such as to maintain the diversity of the population in each generation and to not result in a large computational effort (such as a very large population, for example), given the limitation of the sampling time used (10 seconds). Accordingly, the parameters should be adjusted so as to allow the adaptation of the model for each sampling time, without the system becoming unstable.

The formation of the chromosome follows the same pattern shown in Figure 2, in this case the weights and bias of the output layer neuron. Only these parameters will be modified by assuming that the deviations that occur in relation to the model obtained by offline training do not occur in such a pronounced manner at short time intervals. Therefore, it is logical to conclude that there is no need to adapt all FANN weights, and that there are no significant changes in relation to these two consecutive sampling times. In addition, the values obtained by the offline training can be used as a starting point for the beginning of the adaptation. Thus, there are a total of 6 parameters for optimization, a much lower number in relation to the total number of parameters of the network (31), and demanding less effort from computational processing.

The network obtained by offline training is of the feedforward type, that is, corresponds to a predictor of a step-forward, as shown in Equation (1). In the online controller application, the FANN is transformed into a RNN (Recurrent Neural Network) due to the feedback of the output value, i.e., the network is retrained in real time and used as a predictor of multiple steps ahead. It is noteworthy that the number of neurons and hidden layers remains the same when comparing the FANN with RNN. The change was made in terms of neural connections, wherein the recurring training process enabled feedback of the output variable. Therefore, the weights of the output FANN layer, obtained by offline training, are used as starting values (initial condition) of the weights in same layer of the RNN. The weights of the hidden layer are kept fixed (even while applying online) in the RNN. For a data window of size J, the actual measured values $y(k\text{-}J)$ and $y(k\text{-}1\text{-}J)$ are applied to the network. This generates a sequence of predicted values until instant k, but not using as input for the neural network any actual temperature between instants k and k+1-J, as shown in the sequence of Equations (2) to (5).

$$y_p\left(k+1-J\right)=f\begin{bmatrix}y\left(k-J\right),y\left(k-1-J\right);\\ U_1\left(k-d_1-J\right),U_2\left(k-d_2-J\right)\end{bmatrix} \quad (2)$$

$$y_p\left(k+2-J\right)=f\begin{bmatrix}y_p\left(k+1-J\right),y\left(k-J\right);\\ U_1\left(k-d_1+1-J\right),\\ U_2\left(k-d_2+1-J\right)\end{bmatrix} \quad (3)$$

$$y_p\left(k+3-J\right)=f\begin{bmatrix}y_p\left(k+2-J\right),y_P\left(k+1-J\right);\\ U_1\left(k-d_1+2-J\right),\\ U_2\left(k-d_2+2-J\right)\end{bmatrix} \quad (4)$$

$$\vdots$$

$$\vdots$$

$$y_p\left(k\right)=f\begin{bmatrix}y_p\left(k-1\right),y_P\left(k-2\right);\\ U_1\left(k-d_1-1\right),U_2\left(k-d_2-1\right)\end{bmatrix} \quad (5)$$

The actual measured values (y (k+1-J) to y(k)) are used as targets in Equation (2) to (5) compared to their respective predicted values (y_p(k+1-J) to y_p(k)).

Figure 5 illustrates the representation of FANN and recurrent presentation of training standards, used for online adaptation of weights, where J is the size of the window of points (number of patterns) used for training, obtained from the history of the ongoing process. In the system studied, a window with 40 patterns was used, i.e., considering the current instant k, the (40 + $max(d_1, d_2)$) past points were stored for training. The network evaluation is obtained by the sum of the quadratic error, according to Equation (6).

$$SEMQ=\sum_{i=1}^{J}\left(y_{real,i}-y_{pred,i}\right)^2 \quad (6)$$

where y_{real} is the real output of the process, y_{pred} is the value predicted by the model and J is the size of the standard window.

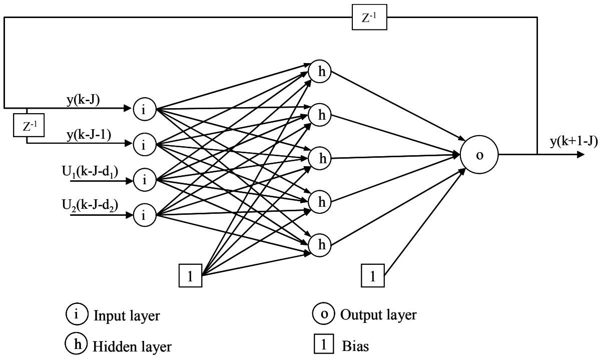


Figure 5: FANN with the representation used in the recurrent training.

One factor that should be taken into consideration when working with online optimization is the processing time, since it must be less than the sampling time used, because there is still the demand of time for the determination and implementation of control actions. One way to reduce the time required for the optimization is by decreasing the search space of the weights (their variation range). In this study, the search space is given at each new interval, always starting from the parameters used in the past interval. The limits definition that will determine the variation range of the weights and *bias* of the FANN is given by Equation (7):

$$\begin{cases} L_{Sup,i} = \gamma_i + \delta \\ L_{Inf,i} = \gamma_i - \delta \end{cases} \tag{7}$$

where $L_{Sup,i}$ and $L_{Inf,i}$ denote the upper and lower limits of each parameter (weights and *bias*), respectively, to define the search space for the optimization by GA. The variable γ_i defines the values of each parameter i used in the past interval. The value of δ will define the size of the search space. In this study the value is $\delta = 0.15$. This procedure makes the adaptive process faster, so that the study does not start from random values from a wide range to proceed to optimization, but from values that are, somehow, close to optimal.

The adjustment of FANN weights for each sampling was chosen so that small variations of the parameters for each application of the algorithm would be needed. Leaving more interval time, the parameters correction would be accentuated, especially during the gel effect, which could destabilize the control.

The GA encoding used was the same as for the offline training of the network. For the online application, the algorithm used was coded in the *Object Pascal* language, programmed in the *Borland Delphi®* compiler. A population of 100 individuals was used for the GA and, according to the capacity of the computer installed in the plant for data acquisition and control strategy implementation, 140 generations were used without affecting the processing of other activities of the controller in the time available.

As it is not possible to ensure that the network obtained by adjusting parameters, in each new interval for 140 generations, is always better than what was implemented in a past instant, or even that which was trained offline, the algorithm presented by Marcolla *et al.* (2009) was used in order to avoid, at any given moment, the use of parameters that may lead to a worse performance for the model. This could cause divergences in the solution, since the parameters are adapted from the values obtained in a previous sampling, making the control system unstable. The algorithm was developed by taking as a principle that, under no circumstances, a model with worse adjustment than that made by the offline training could be used.

All analysis and considerations are based on the window of patterns used for training/adaptation. The criterion to decide which was the best amongst the three models was based on the quadratic average error sum, Equation (6), for the set of standards considered in the window. The model with the lowest sum of the quadratic average error was used in the control law as a prediction model.

Predictive Controller Based on a Nonlinear Model

A predictive controller that uses a feedforward neural network as an empirical model for a batch polymerization process was developed in this work. For the system studied (MISO), two variables must be determined, the opening of the steam flow control valve (U_1) and the opening of the valve which allows the cooling water in (U_2) as a function of a desired value for the temperature (set point). Another factor to be considered is the steam consumption. In any industrial process, the consumption of steam must be minimized because it results in large energy savings and does not require a larger generation device. Based on these premises, Equation (8) is proposed as an objective function for optimization/determination of control actions, which was proposed with reference to the study by Özan *et al.* (1998).

$$J\left(N_1,N_2,N_u\right)=E\left\{\begin{array}{l}\sum_{j=N_1}^{N_2}\left(y\left(k+j\right)-w\left(k+j\right)\right)^2+\lambda_1\sum_{j=1}^{N_u}\left(\Delta U_1\left(k+j-1\right)\right)^2+\\+\lambda_2\sum_{j=1}^{N_u}\left(\Delta U_2\left(k+j-1\right)\right)^2+\lambda_C\sum_{j=1}^{N_u}\left(\left(U_1-1\right)\left(k+j-1\right)\right)^2\end{array}\right\} \tag{8}$$

The signal sent to the valve control can change between 1 to 5 V, corresponding to the states of completely closed and completely open, respectively. As previously described, GA was used for real-time optimization of the objective function. In this case, the definition of the variable search range is required and was described by the following restrictions: $U_{1,min} \leq U_1 \leq U_{1,max}$; $U_{2,min} \leq U_2 \leq U_{2,max}$, where $U_{1,min}=U_{2,min}=$ 1 v and $U_{1,max}=U_{2,max}=$ 5 v.

The value w represents the reference trajectory for the process, determined by Equation (9), according to Camacho and Bordons (1998):

$$\begin{gathered}w\left(t+k\right)=\alpha w\left(t+k-1\right)+\left(1-\alpha\right)r\left(t+k\right)\\k=1\ldots N\end{gathered} \tag{9}$$

where α is parameter with values between 0 and 1. Lower values of α result in a faster transition to the reference, r, whereas higher values cause the transition to slow down.

The goal, when minimizing Equation (8), is to cause the future output $y(k+j)$ to follow the reference $w(k+j)$ and, at the same time, minimize the control efforts $\Delta U_i\left(k+j-1\right)$. Observing Equation (8), it is clear that it has some degrees of freedom (N_1, N_2, N_u, λ_1, λ_2, and λ_C) that can be modified to obtain the desired behavior of the controlled system. N_1 and N_2 indicate the time desired for the output to follow the reference. If N_1 is set to a high value, it means that the errors in the first instants are not important.

The coefficient λ_C determines the importance that opening the valve monitored by action U_1, in this case the steam flow control valve, has on minimizing the cost function. The higher the value of λ_C, the more importance steam consumption will have for the objective function; in other words, high λ_C values have a tendency to lower steam consumption. One should, however, assure that the term governing the consumption does not have a much greater importance in relation to other terms in the equation, which would make the control inefficient. Coefficients λ_1 and λ_2 penalize sudden variations in the control actions.

It should be noted that the coefficient λ_C does not have financial cost units and therefore the values of J obtained from Equation (8) do not directly represent a financial value, but obviously the minimization of Equation (8) will result in a minimization of operating costs.

The experiments for determining the parameters of the controller were conducted on the system in the absence of polymerization. In this case, the reactor was filled with water and controller performance was evaluated in terms of water temperature behavior against disturbances. After a sequence of tests, the following values were reached for the parameters: α=0.6; λ_1=λ_2=0.02 and λ_C=0.01. The prediction horizon, N, used in all experiments was 10 for sampling time, or 100 s for a control horizon, N_U, of an interval.

For the horizon prediction value obtained from the identification of the system without reaction, the reactor has a dynamics similar to a heating tank. In this case, a step disturbance was applied to the system and the evolution of the temperature registered over time. Considering the system dynamics to be first order, the time constant was obtained. In this work, the prediction horizon was defined as numerically equal to the time constant determined.

The values N_1 and N_2 were determined from Equations (10) and (11), respectively (Camacho and Bordons, 1998). Thus, the values used were 4 and 15 intervals, respectively.

$$N_1=\min\left(d_1,d_2\right)+1 \tag{10}$$

$$N_2=\max\left(d_1,d_2\right)+1+N \tag{11}$$

The simplified block diagram that shows the predictive control system used for the process of suspension polymerization of styrene is presented in Figure 6.

RESULTS AND DISCUSSION

As the first step of the experiments, involving the analysis of the control system, the parameters of the objective function have to be obtained and the analysis of their effects on the controller performance evaluated. Their values were determined from empirical assessments of the behavior of the process response to disturbances in the set point of the step and ramp type, with an analysis of the controller acting as a servo. Because for much of the reaction time the process temperature should be kept fixed at a given value, one should also evaluate the performance of the controller acting as a regulator.

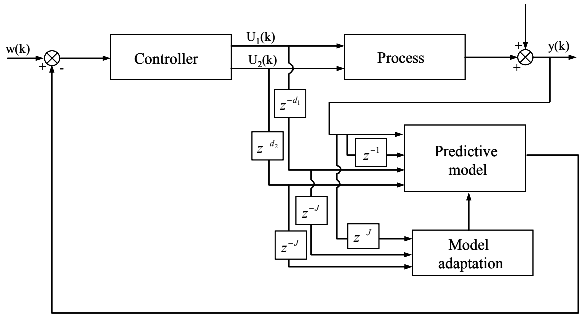


Figure 6: Simplified block diagram of the predictive control system.

In this work we sought to emphasize that, even when a nonlinear controller is used, there is a need for online adaptation of an empirical model that was adjusted from data obtained for a system where no variations occur in the dynamic function of time processing. In this case, the offline FANN was trained to the system containing only water. However, styrene polymerization in terms of energy is characterized by presenting a time variant behavior. Thus, it is expected that the FANN with offline training needs to adapt its weights so that the controller can achieve better performance. Other time-varying influences were presented in the section above on Online Adaptation of FANN weights. In addition, the FANN is transformed into a RNN (Recurrent Neural Network) due to the feedback of the output value, i.e., the network is retrained in real time and used as a predictor of multiple steps ahead (the prediction horizon).

Figure 7 shows the results of the temperature behavior and control actions for testing the controller in the system without reaction, using the parameter values shown in the previous chapter. It is noted that these results were the best among the various tests. The tests were performed using the algorithm for the adaptation of the model, so the behavior of the nonlinear predictive controller was evaluated using a recurrent feedforward network as an empirical model of the process.

Through an analysis of Figure 7, it can be seen that the control system performed well using the parameters determined, acting both as a servo and as a regulator. For both step and ramp disturbances in the set point, there are transitions in which the reference was followed with few deviations.

The following two experiments were conducted in order to provide a performance comparison between the controller based on the static network trained offline and the controller using the network with recurrent online adaptation. In tests, the styrene polymerization reaction with an initiator concentration of 1.578x10^{-3} $\text{gmol}_{\text{BPO}}.\text{gmol}^{-1}{}_{\text{Styrene}}$ was used. The other reaction conditions were presented in the previous sections. Figures 8 and 9 show the polymerization behavior for the control system using the fixed parameter model and recurrent parameter adaptation, respectively.

By comparing Figures 8 and 9, we can see that the adapted model exhibited performance superior to the model without adaptation. In Figure 8 it can be seen that the controller had a good performance in the initial periods of the reaction, the first 2 hours, but was not satisfactory for the remainder of the batch. The occurrence of a high offset in some sections (3 to 4 °C), denounces unforeseen disturbances that occurred in the model for the reaction. Oscillations that occurred in the steam pressure supplied by the boiler were apparently lower than required for

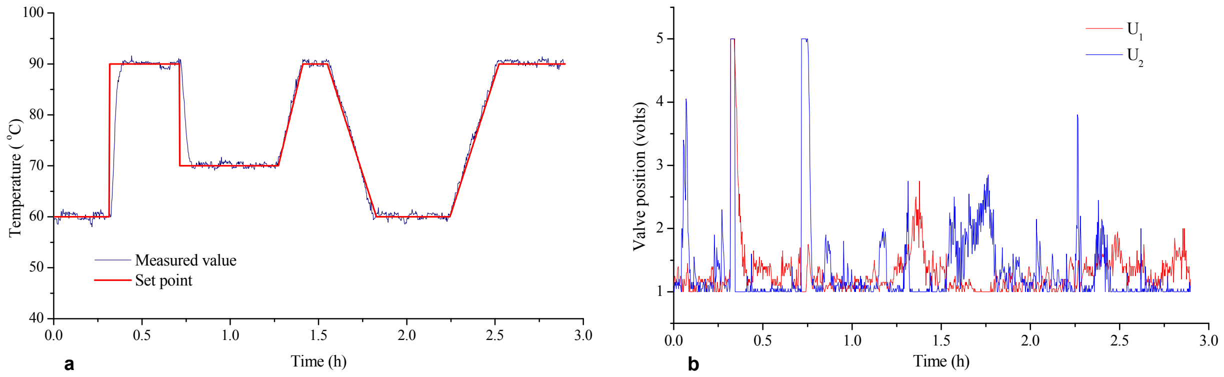


Figure 7: Performance test of the control system for the process without reaction. (a) Temperature behavior. (b) Control actions behavior.

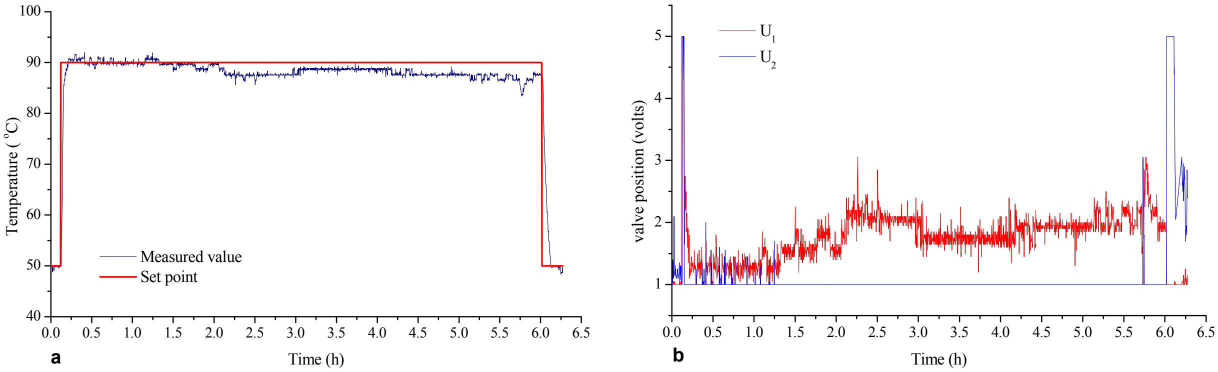


Figure 8: Controller performance analysis for the system with the static network trained offline. (a) Temperature behavior. (b) Control actions behavior.

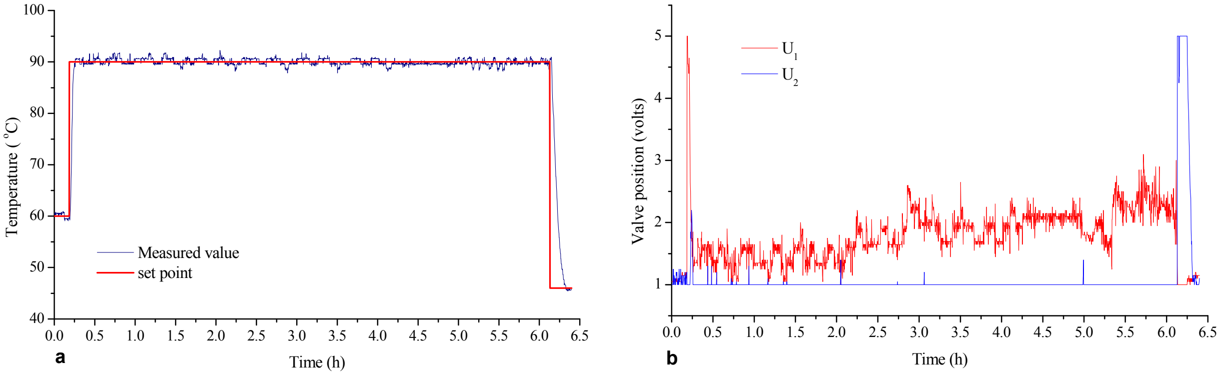


Figure 9: Controller performance analysis for the system with the recurrent network adapted online. (a) Temperature behavior. (b) Control actions behavior.

obtaining of standards to offline training. Because temperature has an influence on the polymerization kinetics and, therefore, on the final properties of the polymer, deviations such as these for long reaction periods are undesirable (Sacks *et al.*, 1973; Erdogan *et al.*, 2002; Hosen and Hussain, 2012). These deviations are not observed in Figure 9, showing that possible changes in the dynamics of the process during the reaction were identified by online training. Furthermore, from the behavior shown in Figure 9, it is clear that the control actions were not longer oscillatory depending on model adaptation, that is, the controller with the adaptive model did not lose robustness with respect to the fixed parameters model and, additionally, managed to eliminate the offset.

One can also evaluate the controller performance by relations based on integral error criteria (Table 2): the IAE (Integral of the absolute value off the error), ISE (Integral of the squared error) e ITAE (Integral of the time-weighted absolute error) (Seborg *et al.*, 1989). For the same time interval, the results show that the controller using a neural network trained online performed better, complementing the discussions already presented.

Table 2: Performance criteria for the models used by the controller.

Model used by the controller	Control performance criteria		
	IAE	ISE	ITAE
Network with offline training (Figure 8)	12.819	103.753	49.716
Network with online training (Figure 9)	4.266	26.638	9.849

The test shown in Figure 9 was also monitored for the time required for data acquisition, training the neural network and optimizing the controller objective function. It was verified that, on average, the time spent was 6 s (on a 500 MHz PC-Intel Pentium III) and never exceeded 8 s. Thus, on average, 60% of the IOS time available for sampling was used. Therefore, there was no problem regarding the time available for implementation of control actions.

Figure 10 shows calculated values for the model parameters (a) and values of the objective function for the original and adapted network (b) throughout the reaction period shown in Figure 9. As presented, only the weights and bias of the neuron of the output layer of the network are adapted, and the others are held fixed at the values determined by offline training. Values of w_i represent the weights of the neuron of the output layer connected to each neuron in the middle layer.

It is noted that, at certain moments, the parameter variation is higher, probably because the greatest deviations occurred between the model prediction and the actual process variable. These deviations are caused by disturbances that occur during the reaction period. In the case of Figure 10b, it is quite clear that, in all intervals and throughout the reaction period, the value of the objective function (sum of quadratic error) is much lower than for the network with adapted parameters and that the values of the adapted network are very close to zero, the optimal value for the function. It is also important to note that the value of the objective function has less oscillation for the adapted network than for the network with fixed parameters.

A comparison between values predicted from the network with adapted parameters and real values for the process temperature is showed in Figure 11. The comparison was always made for instant $\{k+\min(d_1,d_2)\}$, considering k as the interval in which the calculated control actions are implemented. For the process studied, $min(d_1,d_2)=3$, and thus comparisons were made for (k+3). By analyzing Figure 9 it can be concluded that there was an excellent representation of the process by the empirical model with a deviation not exceeding ±0.2 °C during the reaction period, an error of approximately 0.3% when considering the controller range.

The last two tests were conducted in order to test the control strategy with online adaptation of network weights in reactions, when the initiator concentration was changed, the transitional form between set points occurred at the start of the reactor and even the inclusion of recovered material (GPPS - General Purpose Polystyrene) in the reaction mass. These test reactions used 10% GPPS in the reaction load. The other reaction conditions were the same as those presented in the previous section.

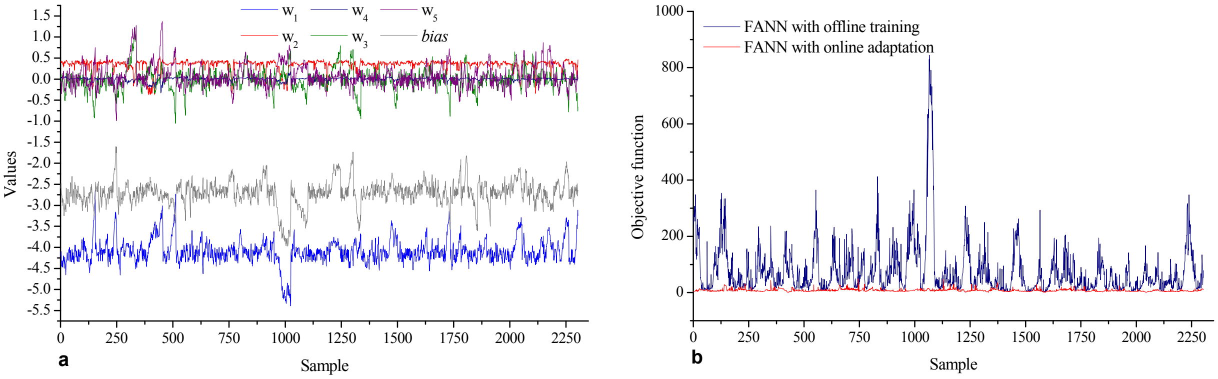


Figure 10: (a) Calculated FANN parameters. (b) Comparison between the values of the objective function for the original and adapted network.

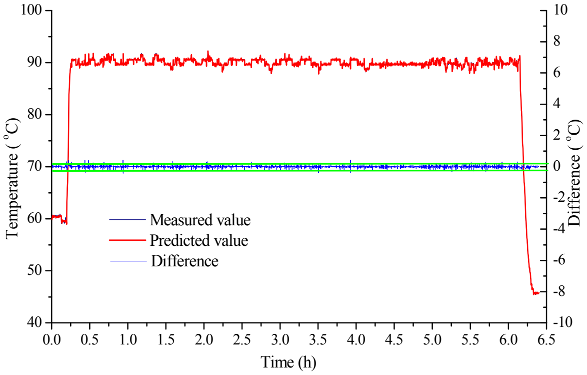


Figure 11: Comparison between the real process values and those provided by the network with *online* adaption of its parameters, for the instant (k+min(d_1,d_2)).

Figure 12 shows the results for the reaction started with a BPO concentration of 2.260x10^{-3} gmol$_{BPO}$.gmol^{-1}$_{Styrene}$. For this experiment, a transition ramp between 60 °C and 90 °C, with a rate of 1.2 °C/min, was used. Figure 13 shows the results for the reaction started with a BPO concentration of 2.713x10^{-3} gmol$_{BPO}$.gmol^{-1}$_{Styrene}$. For this experiment, a ramp transition between 60 °C and 90 °C, with a rate 0.45 °C/min, was used.

It can be verified that the performance of the controller was good, doing the ramp transition according to the desired reference and keeping the temperature at the *set point* during the polymerization period. This behavior is needed for the reaction to occur as planned and in accordance with the optimized operating conditions to obtain the desired molecular weight for the polystyrene produced. Reaction temperature and temperature profile used in the course of the reaction have a high influence on the molecular weight distribution (MWD) of the polymer produced. Several studies have explored these variables in order to produce a polymer with the desired quality in terms of MWD (Wu *et al.*, 1982; Özan *et al.*, 1998; Takamatsu *et al.*, 1988; Kiparissides *et al.*, 2002, Sheibat-Othmana, 2011, Zhang and Zhang, 2011). Thus, a controller that has the ability to maintain the temperature at the set point, in addition to allowing transitions between set points in different ways (step quickly and without the occurrence of overshoot and ramp with the desired variation range) ensures that the goal is reached.

Regarding the strategy of penalizing the steam consumption by the inclusion of another term in the objective function of the control law, Equation (8), Figures 7, 8, 9, 12 and 13 demonstrate that it was successful. In the results for the reaction system, it was observed that the valve used for cooling remains fully closed virtually throughout the reaction time. Thus, the need for steam was kept to a minimum to maintain the temperature at the desired set point (90 °C). If the split-range strategy were used, an intermediate value for opening the steam valve, U_1, would correspond to a complementary value for opening the valve, which allows for a cooling water input, U_2. Thus, to compensate for this action, a greater amount of steam would be required, increasing costs in terms of energy consumption.

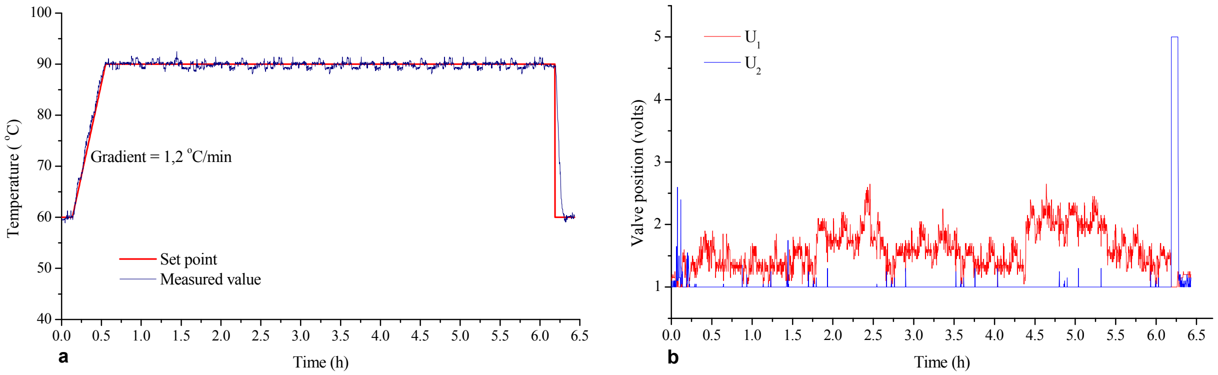


Figure 12: Performance of the controller for the reaction initiated with a BPO concentration of 2.260x10^{-3} gmol$_{BPO}$.gmol^{-1}$_{Styrene}$. (a) Temperature behavior. (b) Control actions behavior.

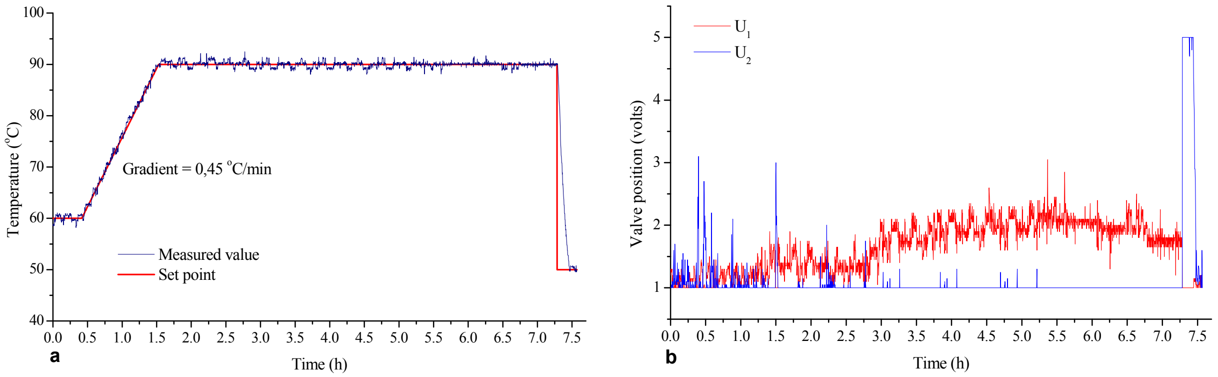


Figure 13: Performance of the controller for the reaction initiated with a BPO concentration of 2.713x10^{-3} gmol$_{BPO}$.gmol^{-1}$_{Styrene}$. (a) Temperature behavior. (b) Control actions behavior.

CONCLUSIONS

In this study, a predictive controller using a feedforward neural network with online adaptation of the parameters was used in the temperature control of a batch process for the styrene suspension polymerization reaction. The control algorithm with online adaptation of the empirical model showed good results in the control system studied. The results, especially in relation to the presence of off-set, were much better than those presented by the controller based on the fixed parameter model. The presence of random disturbances not identified by the patterns used in the offline training of the network could have been the main reason for this behavior. The online adaptation of the network weights allowed the prediction of future states with relative accuracy, which contributed to the good performance of the predictive controller.

The controller implemented also showed a good performance in set-point changes, ensuring rapid transitions in step changes, with no occurrence of overshoot, in addition to maintaining the desired reference in the ramp transitions. Beyond the need to adjust the parameters, it was also observed that the recurrent network training output feedback performed well. For this task, the genetic algorithm technique proved to be very efficient and possible implement in control systems in real time. The versatility of the objective function of the predictive controller successfully enabled the inclusion of a steam consumption penalty term for temperature control of the styrene polymerization reaction. Thus, it was possible to maintain the system at the desired temperature with minimal steam consumption, which reduces the energy costs of the polymerization system.

REFERENCES

Alvarez, L. A. and Odloak, D., Optimization and control of a continuous polymerization reactor. Brazilian Journal of Chemical Engineering, 29(4), p. 807-820 (2012).

Billmeyer, F. W., Textbook of Polymer Science. John Willey and Sons Inc., 3rd Ed., New York (1984).

Bindlish, R. and Rawlings, J. B., Target linearization and model predictive control of polymerization processes. AIChE Journal, 49(11), p. 2885-2899 (2003).

Camacho, E. F. and Bordons, C., Model Predictive Control. Springer Publishers, London, (1998).

Chen, C., Continuous production of solid polystyrene in back-mixed and linear-flow reactors. Polymer Engineering and Science, 40(2), p. 441-464 (2000).

Claumann, C. A., Modelagem dinâmica e controle de processos não lineares: Uma aplicação de algoritmos genéticos para treinamento de redes neurais recorrentes. Masters Dissertation, Federal University of Santa Catarina, Florianópolis (1999). (In Portuguese).

Crowley, T. J. and Choi, K. Y., Experimental studies on optimal molecular weight distribution control in a batch-free radical polymerization process. Chemical Engineering Science, 53(15), p. 2769-2790 (1998).

Cubillos, F. A., Acuña, G., Lima, E. L., Real-time process optimization based on grey-box neural models. Brazilian Journal of Chemical Engineering, 24(3), p. 433-443 (2007).

De Souza Jr., M. B., Pinto, J. C. and Lima, E. L., Control of a chaotic polymerization reactor: A neural network based model predictive approach. Polymer Engineering and Science, 36(4), p. 448-457 (1996).

Erdogan, S., Alpbaz, M. and Karagöz, A. R., The effect of operational conditions on the performance of batch polymerization reactor control. Chemical Engineering Journal, 86, p. 259-268 (2002).

Fernandes, F. A. N. and Lona, L. M. F., Neural network applications in polymerization processes. Brazilian Journal of Chemical Engineering, 22(3), p. 401-418 (2005).

Hosen, M. A., Hussain, M. A. and Mjalli, F. S., Control of polystyrene batch reactors using neural network based model predictive control (NNMPC): An experimental investigation. Control Engineering Practice, 19, p. 454-467 (2011).

Hosen, M. A. and Hussain, M. A., Optimization and control of polystyrene batch reactor using hybrid based model. Computer Aided Chemical Engineering, 31, p. 760-764 (2012).

Houck, C. R., Joines, J. A. and Key, M. G., A genetic algorithm for function optimization: A Matlab implementation. Available in: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.22.4413> (Accessed in September 2011).

Huang, Y. J. and James, Lee, L., Optimization of diffusion-controlled free radical polymerizations in a distributed parameter system. Chemical Engineering Science, 44(2), p. 363-376 (1989).

Janakiraman, V. M., Nguyen, X. and Assanis, D., Nonlinear identification of a gasoline HCCI engine using neural networks coupled with principal component analysis. Applied Soft Computing, 13, p. 2375-2389 (2013).

Kiparissides, C., Seferlis, P., Mourikas, G. and Morris, A. J., On-line optimizing control of molecular weight properties in batch free-radical polymerization reactors. Industrial Engineering Chemistry Research, 41(24), p. 6120-6131 (2002).

Krothapally, M. and Palanki, S., A neural network strategy for batch process optimization. Computers Chemical Engineering, 21, Suppl., p. S463-S468 (1997).

Ławryńczuk, M., Explicit nonlinear predictive control algorithms with neural approximation. Neurocomputing: http://dx.doi.org/10.1016/j.neucom.2013.09. 002i (2013).

Marcolla, R. F., Machado, R. A. F., Cancelier, A., Claumann, C. A. and Bolzan, A., Modeling techniques and processes control application based on neural networks with on-line adjustment using genetic algorithms. Brazilian Journal of Chemical Engineering, 26(1), p. 113-126 (2009).

Ng, C. W. and Hussain, M. A., Hybrid neural network-prior knowledge model in temperature control of a semi-batch polymerization process. Chemical Engineering and Processing, 43, p. 559-570 (2004).

Özkan, G., Hapoglu, H. and Alpbaz, M., Generalized predictive control of optimal temperature profiles in a polystyrene polymerization reactor. Chemical Engineering and Processing, 37, p. 125-139 (1998).

Prasad, V., Schley, M., Russo, L. P. and Bequette, B. W., Product property and production rate control of styrene polymerization. Journal of Process Control, 12, p. 353-372 (2002).

Qiao, J. F. and Han, H. G., Identification and modeling of nonlinear dynamical systems using a novel self-organizing RBF-based approach. Automatica, 48 p. 1729-1734 (2012).

Sacks, M. E., Lee, Soo-II and Biesenberger, J. A., Effect of temperature variations on molecular weight distribution: Batch, chain addition polymerizations. Chemical Engineering Science, 28, p. 241-257 (1973).

Santos, J. C., Lopes, C. N., Reis, M. M., Giudici, R., Sayer, C., Machado, R. A. F. and Araújo, P. H. H., Comparison of techniques for the determination of conversion during suspension polymerization reactions. Brazilian Journal of Chemical Engineering, 25(2), p. 399-407 (2008).

Seborg, D. E., Edgar, T. F. and Mellichamp, D. A., Process Dynamics and Control. John Wiley & Sons, New York (1989).

Sheibat-Othmana, N., Othmana, S., Boyronb, O. and Alamirc, M., Multivariable control of the polymer molecular weight in emulsion polymerization processes. Journal of Process Control, 21, p. 861-873 (2011).

Takamatsu, T., Shioya, S. and Okada, Y., Molecular weight distribution control in a batch polymerization reactor. Industrial Eng. Chemistry Research, 27, p. 93-99 (1988).

Vieira, R. A. M., Embiruçu, M., Sayer, C., Pinto, J. C. and Lima, E. L., Control strategies for complex chemical processes. Applications in Polymerization Processes, Computers and Chemical Engineering, 27, p. 1307-1327 (2003).

Wu, G. Z. A., Denton, L. A. and Laurence, R. L., Batch polimerization of styrene: Optimal temperature histories. Polymer Engineering and Science, 22(1), p. 1-8 (1982).

Yu, D. W. and Yu, D. L., Neural network control of multivariable processes with a fast optimization algorithm. Neural Comput & Applic., 12, p. 185-189 (2003).

Zeybek, Z., Yüce, S., Hapoglu, H. and Alpbaz, M., Adaptive heuristic temperature control of a batch polymerization reactor. Chemical Engineering and Processing, 43, p. 911-920 (2003).

Zhang, J. A., Reliable neural network model based optimal control strategy for a batch polymerization reactor. American Chemical Society, p. A-I (2003).

Zhang, Y. and Zhang, P., Optimization of nonlinear process based on sequential extreme learning machine. Chemical Engineering Science, 66, p. 4702-4710 (2011).