Nonlinear modelling of a reactor-exchanger by using NARX neural networks
Proceedings of European Congress of Chemical Engineering (ECCE-6)
Copenhagen, 16-20 September 2007

Nonlinear modeling of a reactor-exchanger by using NARX neural networks

Y. Chetouani

Department of Chemical Engineering, University of Rouen, Rue Lavoisier, 76821, Mont Saint Aignan Cedex, France. Tel: (0033)235528466, Fax: (0033)235147191, Yahya.Chetouani@univ-rouen.fr

Abstract

The main aim of this paper is to establish a reliable model of process behaviour under its normal operating conditions. The use of this model should reflect the true behaviour of the process and allow distinguishing a normal mode from an abnormal one. In order to obtain this reliable model for the process dynamics, the black-box identification by means of a NARX (Nonlinear Auto-Regressive with eXogenous input) model has been chosen in this study. It is based on the neural network approach. This paper shows the choice and the performance of the neural network in the training and test phases. An analysis of the inputs number, hidden neurons and their influence on the behaviour of the neural predictor is carried out. Three statistical criteria; Aikeke's Information Criterion (AIC), Rissanen's Minimum Description Length (MDL) and Bayesian Information Criterion (BIC) are used for the validation of the experimental data. A reactor-exchanger is used to illustrate the proposed ideas concerning the dynamics modeling. The model is implemented by training a Multi-Layer Perceptron (MLP) Artificial neural network with input-output experimental data. Satisfactory agreement between identified and experimental data is found and results show that the model successfully predicts the evolution of the outlet temperature of the process.

Keywords: reliability, process safety, modelling, neural network, NARX, nonlinear identification

1. Introduction

Process development and continuous request for productivity led to an increasing complexity of industrial units. In chemical industries, it is absolutely necessary to control the process and any drift or anomaly must be detected as soon as possible in order to prevent risks and accidents. Moreover, detecting a fault appearance on-line is justified by the need to solve effectively the problems within a short time (Villermaux, 1996; Chetouani 2007; Chetouani, 2006). The intrinsic highly nonlinear behaviour in the industrial process, especially when a chemical reaction is used, poses a major problem for the formulation of good predictions and the design of reliable control systems (Cammarata et al., 2002). Due to the relevant number of degree of freedom, to the nonlinear coupling of different phenomena and to the processes complexity, the mathematical modeling of the process is computationally heavy and may produce an unsatisfactory correspondence between experimental and simulated data. Similar problems arise also from the uncertainty for the parameters of the process, such as the reaction rate, activation energy, reaction enthalpy, heat transfer coefficient, and their unpredictable variations. In fact, note that most of the chemical and thermo-physical variables both strongly depend and influence instantaneously the temperature of the reaction mass (Chetouani, 2006). One way of addressing this problem is the use of a reliable model for the on-line prediction of the system dynamic evolution. However, designing empirical models like the black-box models is unavoidable (Leontaritis et al., 1985). Various techniques of the processes identification were already proposed. Many researchers employed the neural network to solve several nonlinear complex problems. Engell et al. (Engell et al., 2003) discussed general aspects of the control of reactive separation processes. They used a semi-batch reactive distillation process. A comparison was carried out between conventional control structures and model-based predictive control by using a neural net plant model. Savkovic (Savkovic, 1996) used a neural network for product composition control of a distillation plant. The neural network controller design is based on the process inverse dynamic modeling. The back-propagation algorithm is applied to dynamic nonlinear relationship between product composition and reflux flow rate. The obtained results illustrate the feasibility of using neural net for learning nonlinear dynamic model distillation column from plant input-output data and control. Assaf et al. (Assaf et al., 1996) modeled an ethylene oxidation fixed-bed reactor by a phenomenological model. They compared the results given by this model and those given by the neural model for possible thermal runaway situations of highly exothermic process. The final objective is to build a reliable inference alarm algorithm for fast detection and prevention of this situation. Nanayakkara et al. (Nanayakkara et al., 2002) presented a novel neural network to control an ammonia refrigerant evaporator. The objective is to control evaporator heat flow rate and secondary fluid outlet temperature while keeping the degree of refrigerant superheat at the evaporator outlet by manipulating refrigerant and evaporator secondary fluid flow rates.

The purpose of this identification is to establish a reliable model of the dynamic behaviour of a process as a reactor-exchanger. This reliable model enables to reproduce the process dynamics under different operating conditions in a normal mode. We are interested in the anomaly detection module intended to supervise the functioning state of the system (Chetouani, 2007). The former has to generate on-line information concerning the state of the automated system. This state is characterized not only by control and measurement variables (temperature, rate, etc.), but also by the general behaviour of the process and its history, showing in time whether the behaviour of the system is normal or presents drifts. In the context of numerical control, fault detection and isolation (FDI) proves a vital complement to the adaptive means of dealing with instationarities in nonlinear highly unsteady-state systems. Under normal conditions, the fault detection module allows all information to be processed and managed in direct liaison with its general behaviour. In other case, it detects any anomaly and alerts the operator by setting on the appropriate alarms.

The main aim of this research is to obtain a powerful model of reference allowing to reproducing the process dynamics in normal mode. The present study focuses on the development, and implementation of a NARX neural model for the one-step ahead forecasting of the reactor-exchanger dynamics. The performance of this stochastic model was then evaluated using the performance criteria. Results show that the NARX neural model is representative for the dynamic behaviour of the nonlinear process. Experiments were performed in a reactor-exchanger and experimental data were used both to define and to validate the model. The identification procedure, the experimental set-up and prediction results are described in the following sections.

2. Input-output modeling approach

Modeling is an essential precursor in the parameter estimation process. Identification strategies of various kinds by means of input–output measurements are commonly used in many situations in which it is not necessary to achieve a deep mathematical knowledge of the system under study, but it is sufficient to predict the system evolution (Fung et al., 2003; Mu et al., 2005). This is often the case in control applications, where satisfactory predictions of the system that are to be controlled and sufficient robustness to parameter uncertainty are the only requirements. In chemical systems, parameter variations and uncertainty play a fundamental role on the system dynamics and are very difficult to be accurately modeled (Cammarata et al., 2002). Therefore, the identification approach based on input-output measurements can be applied.

In order to provide a closer approximation to the actual process is some situations, a nonlinear NARX model is employed (Qin et al., 1996; Previdi, 2002), which is identified by means of Artificial Neural Networks (ANN). The NAX model was obtained by using Multi-Layer Perceptron (MLP) Artificial neural networks (Chen et al., 1989; Narendra et al., 1990) to describe accurately the process behaviour. This approach allows bypassing the exact determination of model parameters and of their unpredictable variations, as well as the achievement of deep physical knowledge of the process and of its governing equations.

The nonlinear model of a finite dimensional system (Ljung, 1999) with order (n_y, n_u) and scalar variables y and u are defined by:

$$y(t) = \phi(y(t-1),..., y(t-n_y), u(t-1),...,u(t-n_u)) \tag{1}$$

where *y(k)* is the Auto-Regressive (AR) variable or system output; *u(k)* is the eXogenous (X) variable or system input. n_y and n_u are the AR and X orders, respectively. ϕ is a nonlinear function.

This neural network (1) consists in highly interconnected layers of neuron-like nodes. It has an input and an output layer and any optional layers that are included between these are termed hidden layers. Figure 1 shows typical feed-forward network architecture with one hidden layer. The term 'feed-forward' means that the connections between nodes only allows signals to be sent to the next layer of nodes and not to the previous (Warnes et al., 1996).

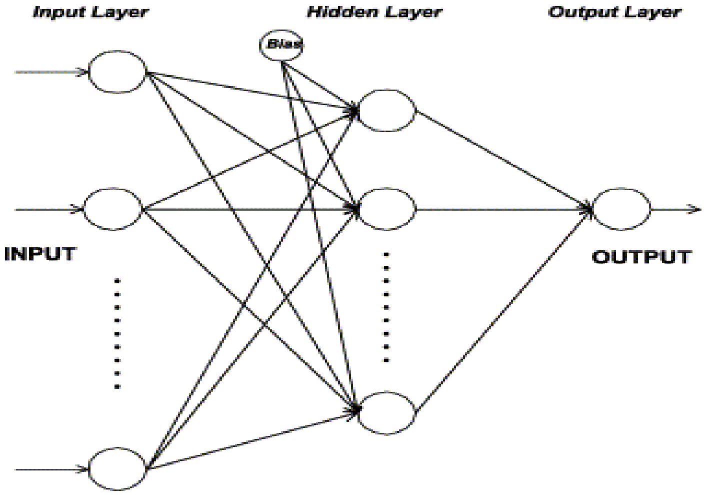

Figure 1. Feed-forward network for prediction

The number of nodes in a hidden layer is determined by the user and can vary from zero to any finite number. The number of nodes in the input and output layers are determined by the number of inputs and by the output variables, respectively. This structure is based on a result by Cybenko (Cybenko, 1989) who proved that a neural network with one hidden layer of sigmoid or hyperbolic tangent units and an output layer of linear units is capable of approximating any continuous function.

$$f(z) = 1/(1+e^{-z}) \tag{2}$$

where z is the sum of the weighted inputs and bias term. The determination of these weights for the node connections allows the ANN to learn the information about the system to be modeled. The input data are presented to the network via the input layer. These data are propagated through the network to the output layer to obtain the network output. The network error is then determined by comparing the network output with the actual output. If the error is not smaller than a desired performance, the weights are adjusted and the raining data are presented to the network again to determine a new network error. One of the most well-known is the back-propagation algorithm (Rumelhart et al., 1986). In this algorithm, as with any other gradient approach, large values of learning rate will speed up the learning process, but lead to instability, and convergence can only be expected for small values of learning rate. The momentum factor is used to damp down oscillations in the learning process. The latter is repeated until the network error reaches the desired performance. In this case the network is then said to have converged and the last set of weights are retained as the network parameters.

2.1. Calculation of the NN output

The following steps explain the calculation of the NN output based on the input vector (Fung et al., 2003).

1. Assign $\hat{w}^T(k)$ to the input vector $x^{\mathrm{T}}(k)$ and apply it to the input units where $\hat{w}^T(k)$ is the regression vector given by the following equation:

$$\hat{w}^T(t) = [y(t-1),..., y(t-n_y), u(t-1),..., u(t-n_u)] \tag{3}$$

2. Calculate the input to the hidden layer units:

$$net_j^h(k)=\sum_{i=1}^{p} W_{ji}^h(k)x_i(k)+b_j^h \tag{4}$$

where p is the number of input nodes of the network, i.e. $p = n_y + n_u + n_b$; j is the jth hidden unit; W_{ji}^h is the connection weight between ith input unit and jth hidden unit; b_j^h is the bias term of the jth hidden unit.

3. Calculate the output from a node in the hidden layer:

$$z_j=f_j^h(net_j^h(k)) \tag{5}$$

where f_j^h is the sigmoid function defined by the equation (2).
4. Calculate the input to the output nodes:

$$net_l^q(k)=\sum_{j=1}^{h} W_{lj}^q(k)z_j(k) \tag{6}$$

where l is the lth output unit; $W_{lj}^q(k)$ is the connection weight between jth hidden unit and lth output unit.

5. Calculate the outputs from the output nodes:

$$\hat{v}_l(k)=f_l^q(net_l^q(k)) \tag{7}$$

where f_l^q is the linear activation function defined by:

$$f_l^q(net_l^q(k)) = net_l^q(k) \tag{8}$$

2. 2. Back-propagation training algorithm

The error function E is defined as:

$$E = \frac{1}{2}\sum_{l=1}^{q}(v_l(k)-\hat{v}_l(k))^2 \tag{9}$$

where q is the number of output units and $v_l(k)$ is the lth element of the output vector of the network. Within each time interval from k to k+1, the back-propagation (BP) algorithm tries to minimize the error for the output value as defined by E by adjusting the weights of the network connections, i.e. $W_{ji}{}^h$ and $W_{lj}{}^q$. The BP algorithm uses the following procedure (Eqs. 10, 11, 12, 13):

$$W_{ji}^h(k+1)=W_{ji}^h(k)+\alpha\Delta W_{ji}^h(k)-\eta\frac{\partial E}{\partial W_{ji}^h(k)} \tag{10}$$

$$W_{lj}^q(k+1)=W_{lj}^q(k)+\alpha\Delta W_{lj}^q(k)-\eta\frac{\partial E}{\partial W_{lj}^q(k)} \tag{11}$$

where η and α are the learning rate and the momentum factor, respectively; $\Delta W_{ji}{}^h$ and $\Delta W_{lj}{}^q$ are the amounts of the previous weight changes; $\partial E/\partial W_{ji}{}^h(k)$ and $\partial E/\partial W_{lj}{}^q(k)$ are given by:

$$\frac{\partial E}{\partial W_{ji}^h(k)}=-\left[z_j(k)(1-z_j(k))x_i(k)\right]\sum_{l=1}^{q}\left[(v_l(k)-\hat{v}_l(k))\hat{v}_l(k)W_{lj}^h(k)\right] \tag{12}$$

$$\frac{\partial E}{\partial W_{lj}^q(k)}=-(v_l(k)-\hat{v}_l(k))z_j(k) \tag{13}$$

The implementation of the NN for forecasting is as follows:

1. Initialize the weights using small random values and set the learning rate and momentum factor for the NN.
2. Apply the input vector given by Eq. 3 to the input units.
3. Calculate the forecast value of the error using the data available at (k−1)th sample (Eqs. 3, 4 5, 6, 7, 8).
4. Calculate the error between the forecast value and the measured value.

5. Propagate the error backwards to update the weights (Eqs. 10, 11, 12, 13).
6. Go back to step 2.

For weights initialization, the Nguyen-widrow initialization method (Nguyen et al., 1990) is best suited for the use with the sigmoid/linear network which is often used for function approximation. The used programming language is Matlab 7.0.4.

3. Experimental results

3.1. Experimental device

The reactor-exchanger is a glass-jacketed reactor with a tangential input for heat transfer fluid. It is equipped with an electrical calibration heating and an input system. It is equipped also with Pt100 temperature probes. The heating–cooling system, which uses a single heat transfer fluid, works within the temperature range between −15 and +200 C. Supervision software allows the fitting of the parameters and their instruction value. It displays and stores data during the experiment as well as for its further exploitation. The input of the reactor-exchanger *u(t)* represents the heat transfer fluid temperature allowing the heating-cooling of the water. *y(t)* represents the outlet temperature of the reactor-exchanger. The process is excited by an input signal, very rich in frequencies and amplitudes, in order to have a data set suitable for the estimation procedure. The sampling period is fixed to 2 seconds. Before starting the estimation of parameters, the database is divided into two separated sets. The first set is used for the estimation of parameters and the second one for the model validation. The first set is sufficiently informative and covering the whole spectrum. The second set contains sufficient elements to make the validation as credible as possible.

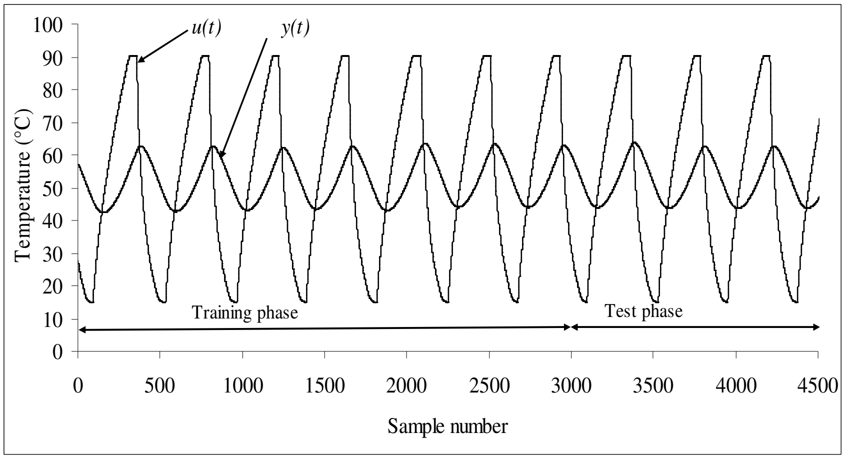

Figure 2. Evolution of the inlet and the outlet temperature

3.2. Establishment of NARX models

To establish a suitable NARX model order for a particular system, neural networks of increasing model order can be trained and their performance on the training data compared using the loss function (or mean squared error), LF. This function is expressed by the following equation:

$$LF = \frac{1}{N}\sum_{i=1}^{N}\varepsilon^{2}(t) \qquad (14)$$

where $\varepsilon\,(t)=y(t)-\hat{y}(t)$ represents the prediction error and N is the data length. The choice of the hidden nodes is carried out between 1 and 15 nodes. In fact, the minimal number of inputs is avoided to ensure the model flexibility. Also, the maximum number of inputs is excluded to avoid the over-fitting. The training on the database gives the evolution of the loss function.

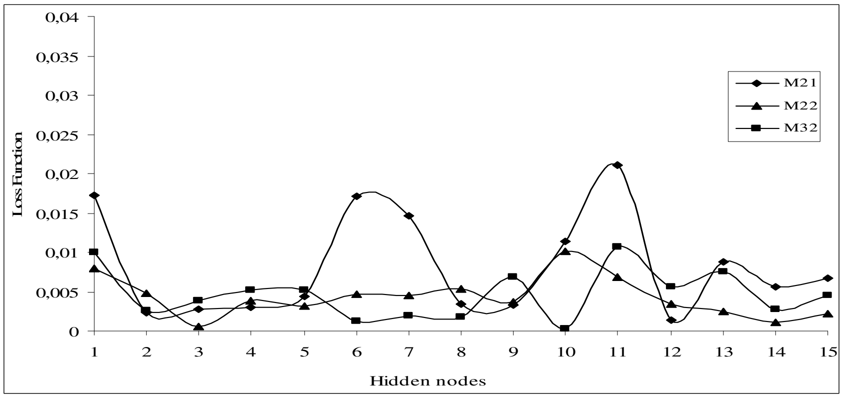

Figure 3. Evolution of the loss function for low complexity models

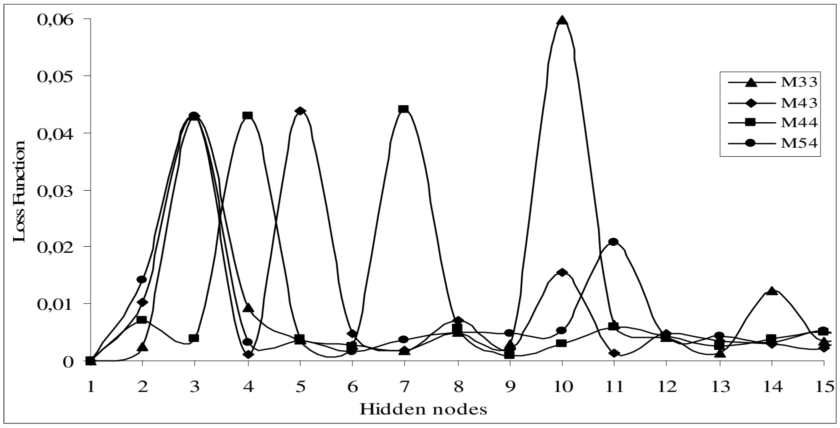

Figure 4. Evolution of the loss function for high complexity models

For showing well the minimum of the *LF* for each model according to the number of hidden nodes, we separate the *LF* evolution in two different figures. Figures 3 and 4 show the *LF* evolution according to the structure of the neural model. One indicates by M*ny.nu.nh* a neural model of which the input layer is made up of *ny* outputs, *nu* inputs and *nh* hidden nodes. These figures show the *LF* on the same training data for different neural network models according the hidden nodes. The M3.2.10 model exhibits the lowest *LF;* however, this model may not be the best choice, because there is a trade off between the model complexity (i.e. size) and accuracy. A small decrease in the *LF* may be rejected if it is at the expense of enlarging the model size. Thus, the decision procedure for selecting a parsimonious model using the *LF* is to decide for each increase in model order whether any reductions in the *LF* are worth the expense of a larger model. The difficult trade off between model accuracy and complexity can be clarified by using model parsimony indices from linear estimation theory, such as Aikeke's Information Criterion (AIC), Rissanen's Minimum Description Length (MDL) and Bayesian Information Criterion (BIC). The validation phase thus makes it possible to distinguish the model describing correctly the dynamic behaviour of the process. These statistical criteria are defined as follows:

$$AIC = ln\left(\frac{N}{2} LF \right) + \frac{2n_w}{N} \tag{15}$$

$$MDL = ln\left(\frac{N}{2} LF \right) + \frac{2n_w \, ln(N)}{N} \tag{16}$$

$$BIC = ln\left(\frac{N}{2} LF \right) + \frac{n_w \, ln(N)}{N} \tag{17}$$

where n_w is the number of model parameters (weights in a neural network).

Hence, the AIC, MDL and BIC are weighted functions of the *LF* which penalize for reductions in the prediction errors at the expense of increasing model complexity (i.e. model order and number of parameters). Strict application of these statistical criteria means that the model structure with the minimum AIC, MDL or BIC is selected as a parsimonious structure. However, in practice, engineering judgment may need to be exercised. Also, for showing well the evolution of AIC, MDL and BIC criteria according the *LF* minimum for each model, we plot the figure 5. The corresponding criteria are shown in figure 5.

A strict application of the indices would select the models M2.2.3. and M3.2.10 because they exhibit the lowest of three indices for all the model structures compared. Although, in this case, the AIC, MDL and BIC criteria do not provide a clear indication of a particular model, the interpretation of the these criteria results described does provide further support for the choice of a M3.2.10 model indicated by the *LF*. Based on engineering judgment, the model M2.2.3 would be preferred without significant loss of accuracy.

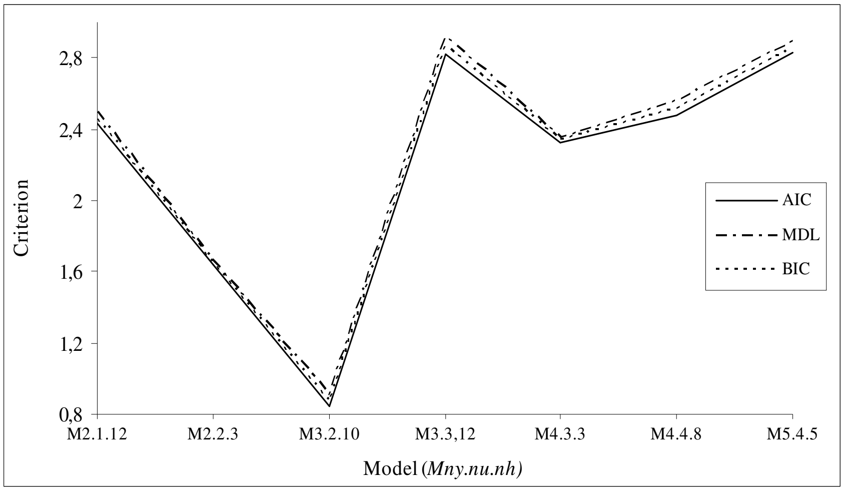

Figure 5. Evolution of the criteria for the LF minimum

3.3. Residual analysis

Once the training and the test of the NARX model have been completed, it should be ready to simulate the system dynamics. Model validation tests should be performed to validate the identified model. Billings et al. (Billings et al., 1986) proposed some correlations based model validity tests. In order to validate the identified model, it is necessary to evaluate the properties of the errors that affect the prediction of the outputs of the model, which can be defined as the differences between experimental and simulated time series. In general, the characteristics of the error are considered satisfactory when the error behaves as white noise, i.e. it has a zero mean and is not correlated (Cammarata et al., 2002; Billings et al., 1986). In fact, if both these conditions are satisfied, it means that the identified model has captured the deterministic part of the system dynamics, which is therefore accurately modeled. To this aim, it is necessary to verify that the auto-correlation function of the normalized error $\varepsilon(t)$, namely $\phi\varepsilon\varepsilon\ (\tau)$, assumes the values 1 for t=0 and 0 elsewhere; in other words, it is required that the function behaves as an impulse. This auto-correlation is defined as follows (Zhang et al., 1996; Billings et al., 1986):

$$\phi\varepsilon\varepsilon\ (\tau)=E(\varepsilon\ (t-\tau)\varepsilon\ (t)]=\delta(\tau) \qquad \forall\tau, \tag{18}$$

where ε is the model residual. $E(X)$ is the expected value of X, τ is the lag.

This condition is, of course, ideal and in practice it is sufficient to verify that $\phi\varepsilon\varepsilon\ (\tau)$, remains in a confidence band usually fixed at the 95%, which means that $\phi\varepsilon\varepsilon\ (\tau)$ must remain inside the range $\pm\frac{1.96}{\sqrt{N}}$, with N the number of testing data on which $\phi\varepsilon\varepsilon\ (\tau)$ is calculated. Billings et al. (Billings et al., 1986) proposed also tests for looking into the cross-correlation among model residuals and inputs. This cross-correlation is defined by the following equation:

$$\phi\ u\varepsilon\ (\tau)=E(u\ (t-\tau)\varepsilon\ (t)]=0 \qquad \forall\tau, \tag{19}$$

To implement these tests (18, 19), u and ε are normalized to give zero mean sequences of unit variance. The sampled cross-validation function between two such data sequences *u(t)* and $\varepsilon\ (t)$ is then calculated as:

$$\phi u\varepsilon(\tau)=\frac{\sum_{t=1}^{N-\tau}u(t)\varepsilon\ (t+\tau\ )}{\left[\sum_{t=1}^{N}u^{2}(t)\sum_{t=1}^{N}\varepsilon\ ^{2}(t)\right]^{1/2}} \tag{20}$$

If the equations (17, 18) are satisfied then the model residuals are a random sequence and are not predictable from inputs and, hence, the model will be considered as adequate. These correlations based tests are used here to validate the neural network model. The results are presented in figure 6. In these plots, the dash dot lines are the 95% confidence bands.

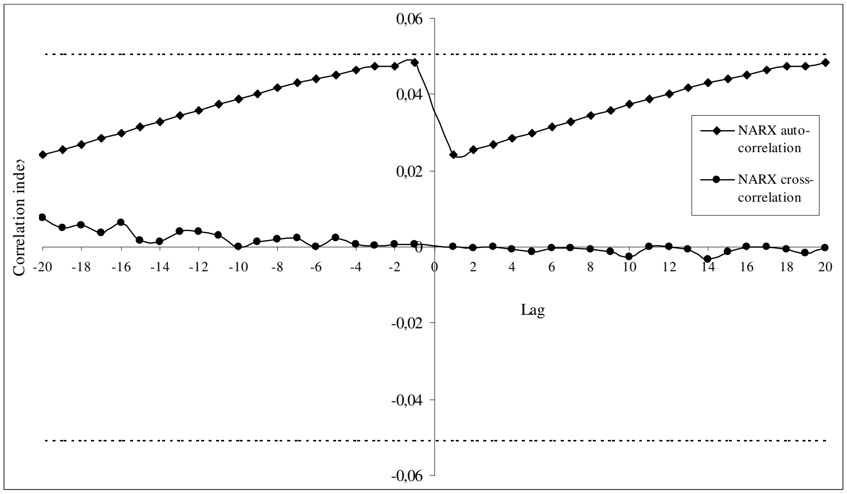

Figure 6. Results of model validation tests

The evolution of the cross-correlation of the NARX model is inside the 95% confidence bands. In addition, the NARX cross-correlation is low. This explains the independence of the residual signal from the input one. For the auto-correlation of the NARX neural model, all points are inside the 95% confidence bands. Therefore, this model is considered a reliable one for describing the dynamic behaviour of the process. This validation phase is used with the neural weights found in the training phase. There is a good agreement between the learned neural model and the experiment in the validation phase. This result is important because it shows the ability of the neural network with only one hidden layer to interpolate any nonlinear function (Cybenko, 1989). Figure 7 shows the difference between the experimental output and those simulated par the neural model M2.2.3.

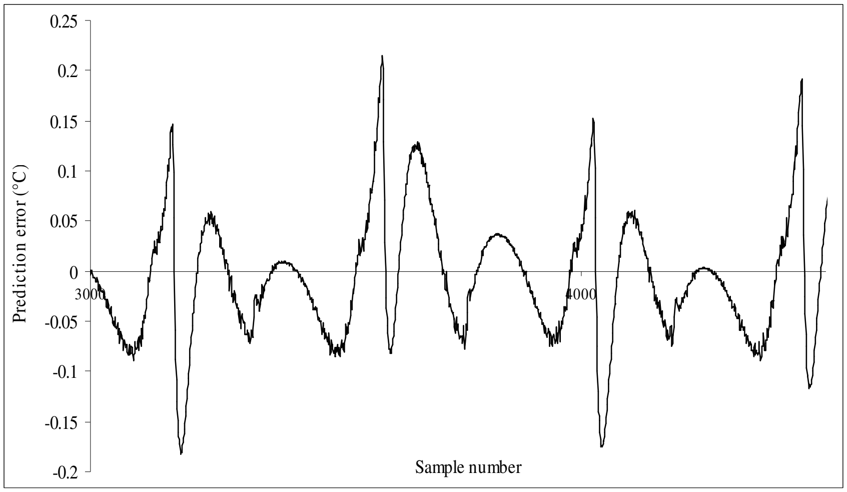

Figure 7. Prediction error of the output temperature

Analyzing this figure, it emerges that the NARX model M2.2.3 ensure satisfactory performances as it is indeed able to correctly identify the dynamics of the reactor-exchanger. The main advantage of the proposed neural approach consists in the natural ability of neural networks in modeling nonlinear dynamics in a fast and simple way and in the possibility to address the process to be modeled as an input-output black-box, with little or no mathematical information on the system.

4. Conclusion

This work aims to identify process dynamics by means of a NARX model. The identification of the system dynamics by means of input-output experimental measurements provides a useful solution for the formulation of a reliable model. This paper aimed at identifying the dynamics of a process like a reactor-exchanger in order to provide reliable predictions. The identification of the system was performed by means of the NARX approach implemented using a neural network. In this case, the results showed that the model is able to give satisfactory descriptions of the experimental data. Moreover, the developed neural model is used in a recursive scheme in order to test their ability to perform long-term predictions. Although the predictive capability of the models is limited to a few steps ahead and varies with the variable considered, the time period for which satisfactory predictions were achieved is sufficient for the implementation of the neural models in a more complex control scheme. Finally, the identified neural model will be useful as a reference one for the fault detection and the isolation (FDI) which can occur through the process dynamics.

References

Assaf, E.M., Giordano, R. C., Nascimento, C.A., (1996) *Chemical engineering science*, 3, 107-112.

Billings, S.A., Voon, W.S.F., (1986) *International journal of Control*, 44, 235–244.

Cammarata, L., Fichera, A., Pagano, A., (2002) *Applied Energy*, 72, 513-528.

Chen, S., Billings, S.A., (1989) *International Journal of Control,* 49, 1013–1032.

Chetouani, Y., (2006) *Process Safety and Environmental Protection*, 84, 27-32.

Chetouani, Y., (2007) *International Journal of Reliability, Quality and Safety Engineering*, 14, 65-80, 2007.

Cybenko, G., (1989) *Math. Control Signals Systems*, 4, 303–312.

Engell, S., Fernholz, G., (2003) *Chemical Engineering and Processing*, 42, 201-210.

Fung, E.H.K., Wong, Y.K., Ho, H.F., Mignolet M.P., (2003) *Applied Mathematical Modelling*, 27, 611-627.

Leontaritis, I.J., Billings, S.A., (1985) *Int. J. Control*, 41, 303–328.

Ljung, L., (1999) *System Identification, Theory for the User,* Prentice Hall, Englewood Cliffs, New Jersey.

Mu, J., Rees, D., Liu, G.P., (2005) *Control Engineering Practice*, 13, 1001-1015.

Nanayakkara, V.K., Ikegami, Y., Uehara, H., (2002) *International Journal of Refrigeration*, 25, 813-826.

Narendra, K.S., Parthasarathy, K., (1990) *IEEE Trans. on Neural Networks*, 1, 4–21.

Nguyen, D., Widrow, B., (1990) *International Joint Conference of neural networks*, 3, 21-26.

Previdi, F., (2002) *Control Engineering Practice*, 10, 91-99.

Qin, S.J., McAvoy, T.J., (1996), *Computers chem. Engng*, 20, 147-159.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., (1986) *Nature*, 323, 533–536.

Savkovic-Stevanovic, J., (1996) *Computers chem. Engng*, 20, 925-930.

Villermaux, J., (1996) *Proceedings of the Plenary Reading in Fifth World Congress of Chemical Engineering*, San Diego, CA.

Warnes, M.R., Glassey, J., Montague, G.A., Kara, B., (1996) *Process Biochemistry*, 31, 147-15.

Zhang, J., Morris, J., (1996) *Fuzzy Sets and Systems*, 79, 127-140.