Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Intelligent Modelling of MIMO Nonlinear Dynamic Process Plants for Predictive Control Purposes

Morteza Mohammadzaheri*, Ley Chen**

** Mechanical Engineering School (Room 206), NTCE campus of the University of Adelaide Australia (Tel: 08 83033156; e-mail: morteza@ mecheng.adelaide.edu.au).*
*** Mechanical Engineering School, University of Adelaide,(e-mail: lei.chen@mecheng.adelaide.edu.au)*

Abstract: In this research, input/output data of a MIMO nonlinear system are used to create intelligent models. Multi layer perceprtrons and neuro-fuzzy networks are utilized for this purpose. For the purpose that these models suit predictive control in their best, a variety of subtle points should be considered. Recurrent models and subtractive clustering are used in this research, and a pre-processing is exerted on the columns of raw data. Then the prepared data are used to train models. A reliable checking process is also offered. A Catalytic Continuous Stirred Tank Reactor is used as case study. A computer model is used to gather input/data rather than a real one. Finally, the simulation is successfully performed to indicate the capabilities of intelligent modeling methods as well as the importance of the points offered through this paper.

1. INTRODUCTION

In feedback control, as the most common type of control, the control command is generated using the error which has already occurred, whereas, in predictive control the predicted error (which is going to occur) is utilized in the determination of control command to avoid the error before appearing (Camacho and Bordons,2004). Predictive control was initially introduced as the classical model predictive control which needs a linear state space model of system (Camacho and Bordons,2004, Bemporad et al, 2007). However, the nonlinearity of many systems is not negligible; so, linear state space models can not represent such models properly. In such occasions, approximate fully (Li and Christofides,2007, Feng et al,2007, Nagy et al,2007) or piecewise (Cervantes et al,2003, Magni and Scattolini,2007) linear models may be used. But, in general, nonlinear models are needed to predict the output(s) of nonlinear system for control purposes. There are some physics-based methods which define the model of some systems entirely (Holenda et al,2007) or partially (Harnischmscher and Marquardt,2007) (the structure of model). Artificial neural networks (Aggelogiannaki et al,2007, Seyab and Cao,2007 , Mohammadzaheri and Chen,2007, Demuth et al,2007) and fuzzy inference systems (neuro-fuzzy networks) (Barros and Dexter,2007, Karer et al,2007, Na and Upsdhyaya, 2007, Ghaffari et al,2007) also can model the systems. These methods are categorized as intelligent modelling methods. These models should be trained using input-output data after initial design.

In this paper, the case study (for intelligent modelling) is a Catalytic Continuous Stirred Tank Reactor (CSTR) which a computer model of this system is available (Demuth et al,2007), the output-input data of this model are used instead of experimental data. The studied CSTR is a dynamic nonlinear MIMO system. The purpose of this research is to indicate the capabilities of intelligent modelling for nonlinear predictive purposes, and offering some helpful subtle points for such applications of intelligent systems. The offered methodology can be used if a precise mathematical model is not available through classical methods (e.g. mass-energy equilibrium equations).

2. INTELLIGENT MODELING OF DYNAMIC SYSTEMS

Intelligent systems which are capable to be trained by input-output data can be properly utilized in the modelling of industrial process plants. There are three main structures for intelligent models; perceptron neural networks (Haykin,1999), neuro-fuzzy networks (Jang et al,2006) and radial basis function networks (Jang et al,2006). The two first structures are used in this research. RBF networks are powerful modelling tools, but in practice, these neural networks need a lot more neurons in comparison two other models (Demuth et al, 2007). As a result, RBFNs are rarely practically applicable in the modelling of complex nonlinear MIMO systems.

In total, in the I/O data based modelling, a system is basically defined based on the signal(s) (e.g. temperature, pressure or …) which is important to be predicted or estimated. This/these signal(s) is/are called the output(s). All other signal(s) which influence on the output(s) are considered as the input(s). Almost all industrial process plants (including chemical plants) are dynamic systems. In a dynamic system, the current value of the system's output(s) not only is/are affected by the inputs of the system, but also is affected by the value(s) of the system's output(s) at previous instants. In order to model dynamic systems, dynamic or recurrent models are needed.

After modelling of a dynamic system, in the discrete domain, output value(s) at previous instants are used as the inputs of the model to predict/estimate the current value of the output. For a single input-single output dynamic system, with input of u and output of y, a nonlinear recurrent model can be described by following equation:

$$y_s(k+1) = f(u(k-nu+1), \ldots, u(k), y(k-ny+1), \ldots, y(k)) \qquad (1)$$

r =the order of the model= $max(nu, ny)$

y_s is the estimated output and f represents the mathematical model. Usually $ny \geq nu$, therefore, the order of the model is often considered as the number of delayed outputs which are used for the estimation/prediction.

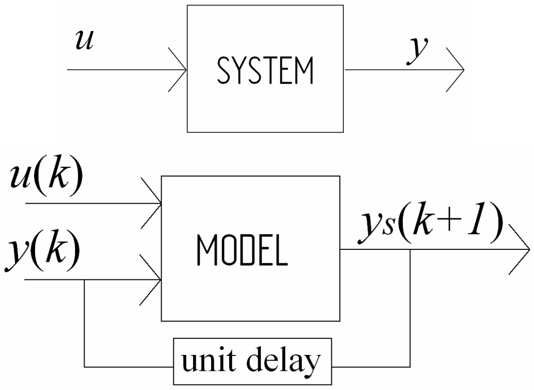

Fig.1: A scheme of a first order SISO dynamic system and its recurrent model, u and y are input and output

Recorded/sensed data of input/output signals are usually available in the form of numeric arrays. These data should be arranged properly to be used in the training/modelling. To achieve the model f (shown in (1)) through training the sensed/recorded raw data of this SISO system is prepared as below:

$$raw\,data = \left[\begin{array}{c||c} \overbrace{}^{input} & \overbrace{}^{output} \\ u_1 & y_1 \\ \vdots & \vdots \\ u_k & y_k \end{array}\right] \qquad (2)$$

$$prepared\,data =$$

$$\left[\begin{array}{ccc|ccc||c} \multicolumn{6}{c||}{\overbrace{\qquad\qquad\qquad\qquad\qquad\qquad}^{input}} & \overbrace{\quad}^{output} \\ u_1 & \cdots & u_{nu} & y_1 & \cdots & y_{ny} & y_{ny+1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ u_{k-r} & \cdots & u_{k+nu-r-1} & y_{k-r} & \cdots & y_{k+ny-r-1} & y_{k+ny-r} \end{array}\right] \qquad (3)$$

Fixed value input signals are not used in the modelling and the achieved model is valid while these signals keep fixed. In such occasions, fixed value inputs are considered as the parameters of the system (instead of input signals).

In the modelling, using input-output data (training), If some of inputs of the model have values with higher magnitudes, their relevant connection weights /activation functions may influence on the output more than other connection weights/activation functions. Consequently, the training procedure particularly tends to adjust the connection weights/activation functions regarding inputs with the higher numeric values, and some parameters may not be subject to sufficient modification. In order to avoid the aforementioned problem, the numeric arrays, representing the signals, should have the magnitudes close to each other. A suitable way to guarantee this matter is the normalization of the data arrays so that the total sum of squared values of each error is of a definite value.

3. CHECKING THE MODEL. A CRITICAL HINT

After the completion of modelling, the achieved model should be checked. It is clear that checking data should be different from training data. The model should be able to predict the behaviour of system for many future instants; therefore, in checking the model, after the very first instants, previously estimated values of output should be used as the delayed output(s) which are the inputs of recurrent model. Consider (1):

$$y_s(k+1) = f(u(k-nu+1), \ldots, u(k), y(k-ny+1), \ldots, y(k)) \qquad (1)$$

This equation is used at the first estimation; at this step all the values of y, in the past instants, are available. Whereas, for the prediction of the output at one instant later $y_s(k+2)$, $y(k+1)$ is needed , but there is no recorded data for this value, so the estimated value at the first step is used for the prediction or :

$$y_s(k+2) = f(u(k-nu+2), \ldots, u(k+1), y(k-ny+2), \ldots, y_s(k+1)) \qquad (4)$$

After ny instants, all the delayed outputs are estimated ones:

$$y_s(k+ny+1) = f(u(k-nu+ny+1), \ldots, u(k+ny), y_s(k+1), \ldots, y_s(k+ny)) \qquad (5)$$

Since, in checking, both input and output data are available for the designer/analyser, sometimes the checking data are mistakenly arranged same as training data (see (3)) and the prepared input data are given to the model (see (1)) and the estimated data are used to assess the accuracy of model. Using this method, a very high accuracy is usually observed which is absolutely unreliable.

As an example, consider the system of fig.1, and assume a series of 500 recorded data sets are available to check the model. The estimation/prediction is started at r+1th instant (r=1). At very first estimation, the recorded output is used:

$$y_s(2) = f(y(1), u(1)). \qquad (6)$$

Other recorded values of output ($y(k)$, k=2,…,500) are only used to specify the accuracy. In order to predict $y_s(3)$, $y_s(2)$ should be used (not y(2)):

$$y_s(3) = f(y_s(2), u(2)). \qquad (7)$$

There are software-packages usable for design and training of intelligent models for predictive control, the checking accuracy offered by these softwares are sometimes calculated without the consideration of the recurrence of the model , that is recorded outputs , in checking data, are used as the delayed inputs instead of previously estimated outputs, or only (1) is used for checking. So unordinary high accuracies offered by such softwares should be checked by the designer.

4. NONLINEAR PREDICTIVE CONTROL

In nonlinear predictive controller, usable in the predictive control of nonlinear systems, nonlinear models are needed to predict the behaviour of nonlinear systems. In a typical nonlinear predictive control in discrete domain, at the instant of *k*, the output of system is known ($y(k)$), and the tentative control command (u') is calculated to be applied as the control command at the same instant of *k*. An optimisation method defines the control command (represented by u') based on minimising a performance function involving the predicted error (8) is a typical performance function(represented by J), which is usually used in neuro-predictive control.

$$J(k)=\sum_{i=1}^{N}[y_s(k+i)-y_d]^2+\rho[u'(k)-u(k-1)]^2. \quad (8)$$

y_s and y_d are the estimated and desired outputs of the system respectively, and u' and u are tentative and actual control inputs. Additionally, ρ is a factor defining the importance of the constancy of control input.
As the first stage of the definition of tentative control command, the performance function (J) should be calculated. To do so, the output values should be predicted for *N* future instants (see (8)), so the nonlinear model should be used *N* times. *N* is named the horizon of prediction. The predicted output value of any stage of prediction is applied as one of the inputs for the next prediction stage. Figure 2 shows how predicted values of output are achieved, for the system shown in Fig.1 and the prediction horizon of 4.

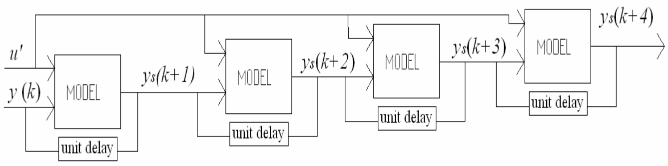

Fig.2: Prediction of output values with the horizon of 4

Using nonlinear model, predicted output values of system ($y_s(k+i)$, $i=1\sim N$) are known. Using previous and current values of the output of system (y) , previous values of control input (u) , tentative control input (u') and ρ the value of performance function (J) is available.
If current output and previous output/input of the system (as the recorded data) and u' are known, all other arguments of J will be definitely known (see fig.2). So these arguments can not be subject to modification by optimization algorithms. However u' can be changed arbitrarily freely from the recorded input/output data and this change affects other arguments of J , then the performance function itself. Therefore, in the optimisation for control purposes, it can be assumed:

$$J=J(u'). \quad (9)$$

Finding u' so as to minimise the performance function is the last stage at nonlinear predictive control.

5. CASE STUDY

The case study is a Catalytic Continuous Stirred Tank Reactor (CSTR). A diagram of the process is shown in the following figure:

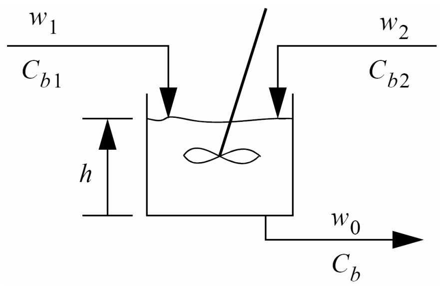

Fig.3: A schematic of the studied CSTR(Demuth et al,2007)

Two flows of liquid enter the reactor with the concentration of $C_{b1}=24.9$ (kg/ m^3) and $C_{b2}=0.1$ (kg/ m^3). The flow rate of input flows are named w_1 and w_2 .The reactor outlets another flow of liquid with the concentration of C_b and the flow rate of w_0 . Another important variable is the height of liquid in the reactor (*h*).
A simplified mathematical model of system, achieved by mass equilibrium equations, is:

$$\frac{dh(t)}{dt}=w_1(t)+w_2(t)-0.2\sqrt{h(t)} \quad (10)$$

$$\frac{dC_b(t)}{dt}=(C_{b1}-C_b(t))\frac{w_1(t)}{h(t)}-(C_{b2}-C_b(t))\frac{w_2(t)}{h(t)}-\frac{k_1C_b(t)}{(1+k_2C_b(t))^2} \quad (11)$$

If the concentration of outlet flow and the height of liquid are considered as the outputs ($w_0=0.2\sqrt{h}$), the total system can be shown as below:

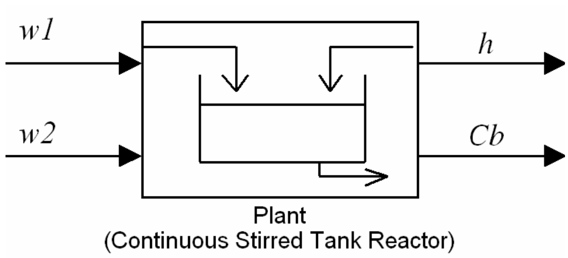

Fig.4: the studied CSTR as a MIMO system

6. INTELLIGENT MODELING OF CASE STUDY

In this research, the modelling is performed particularly for the purpose of predictive control. The case study can potentially have two control inputs, w_1 and w_2 . Inasmuch as in predictive control one control command is often used, one of these potential control inputs can be fixed. So, the flow rate of the second input flow (with the concentration of 24.9 kg/m^3) is set to the constant value of 0.1 $\frac{m^3}{s}$. As a result, this value is not considered in modelling as an input signal anymore (see sec.3). The order of two is considered for the model, and all values *nu* and *ny* (see (1)) are set to two. So such a model is made for the system:

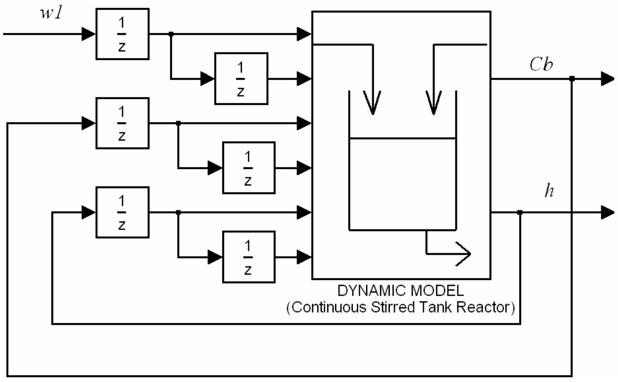

Fig.5: Dynamic model of CSTR, when the flow rate of an input flows is fixed

This model (presented in fig.5 and (12)) is used to return the first estimated value of C_b :

$$[\hat{C}_b(k+1), \hat{h}(k+1)] = F[w_1(k-1), w_1(k), C_b(k-1), C_b(k), h(k-1), h(k)] \quad (12)$$

Or

$$\hat{C}_b(k+1) = F_1[w_1(k-1), w_1(k), C_b(k-1), C_b(k), h(k-1), h(k)] \quad (13)$$

$$\hat{h}(k+1) = F_2[w_1(k-1), w_1(k), C_b(k-1), C_b(k), h(k-1), h(k)] \quad (14)$$

After very first instants of prediction (see (5) and (12)):

$$[\hat{C}_b(k+1), \hat{h}(k+1)] = F[w_1(k-1), w_1(k), \hat{C}_b(k-1), \hat{C}_b(k), \hat{h}(k-1), \hat{h}(k)] \quad (15)$$

or

$$\hat{C}_b(k+1) = F_1[w_1(k-1), w_1(k), \hat{C}_b(k-1), \hat{C}_b(k), \hat{h}(k-1), \hat{h}(k)] \quad (16)$$

$$\hat{h}(k+1) = F_2[w_1(k-1), w_1(k), \hat{C}_b(k-1), \hat{C}_b(k), \hat{h}(k-1), \hat{h}(k)] \quad (17)$$

where variables with hat are estimated/predicted ones.

Some times, influenced by system shown in fig.4, one of the outputs is ignored mistakenly. For instance, it is assumed that:

$$\hat{C}_b(k+1) = F[w_1(k-1), w_1(k), \hat{C}_b(k-1), \hat{C}_b(k)] \quad (18)$$

It should be noted; the system is dynamic and even if *h* is not important for the designer as an output to control, it can not be ignored in modelling because of its effect (as the representative of liquid volume) on the value of C_b . Such a mistake exists in some software packages usable in nonlinear predictive control, the results of the usage of such incomplete models are also shown in this paper.

After the definition of model's order, the training data should be normalized and arranged. 8000 set of data (including w_1, *h* and C_b) with time interval of 0.2 second are utilized in training. These normalized data are arranged as (19).

Although only one predicted value is often used in predictive control, both outputs should be estimated, because system is dynamic and the outputs are coupled. Therefore, the estimated values of both outputs are needed to predict any of them for a period of time in the future. A four-layer recurrent perceptron and a couple of recurrent neuro-fuzzy networks are trained using the prepared data (see (19)).

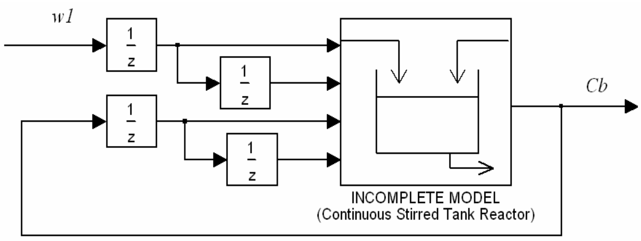

Fig.6: Incomplete dynamic model of CSTR, without height consideration

$$\left[\begin{array}{cccccc||cc}
\multicolumn{6}{c||}{\overbrace{\hphantom{w_1(7998)\;w_1(7999)\;C_b(7998)\;C_b(7999)\;h(7998)\;h(7999)}}^{input}} & \multicolumn{2}{c}{\overbrace{\hphantom{C_b(8000)\;h(8000)}}^{output}} \\
w_1(1) & w_1(2) & C_b(1) & C_b(2) & h(1) & h(2) & C_b(3) & h(3) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
w_1(7998) & w_1(7999) & C_b(7998) & C_b(7999) & h(7998) & h(7999) & C_b(8000) & h(8000)
\end{array}\right] \quad (19)$$

The input layer of the utilized perceptron has six neurons (equal to input signals). This ANN has one nonlinear (with sigmoid activation functions) and one linear (with linear activation functions with slope of 1) hidden layer. Both hidden layers have 13 neurons. The output layer also has two neurons with linear activation functions with slope of 1. Linear hidden layers may seem useless at the first glance, because a linear combination of the outputs of nonlinear hidden layer is generated at the output layer; however, adding this layer improves the accuracy, in practice. It seems, a wider variety of adjusting parameters let the model be trained more successfully. The training method is Levenberg-Marquardt error back propagation .The (batch) training has been performed in 100 epochs and the performance function is sum of squared errors (MSE). According to (18), another neural network is also designed and trained as an incomplete neural model. This ANN has a hidden layer including nine nonlinear sigmoid neurons. This structure suites best with training/checking data among many different attempted structures. The training method is similar to the complete neural model.

For neuro-fuzzy modelling, adaptive neuro-fuzzy inference system (ANFIS) is used. But ANFIS can learn only single output systems. In order to solve this problem, two parallel ANFIS are made and trained. Any of them are relevant to one of outputs, (see (16) and (17)). In checking, both ANFIS models are used simultaneously and the output of any one is used as the inputs of both ANFIS models at the next instant.

If three fuzzy values is allocated to any of six inputs of fuzzy model, 3^6 (=729) fuzzy rules are needed for each ANFIS model. In order to avoid having such a big model, subtractive clustering is utilized. The parameters of subtractive clustering are shown in Table.1. After subtractive clustering, ANFIS1 and ANFIS2 are of 17 and 15 fuzzy rules respectively.

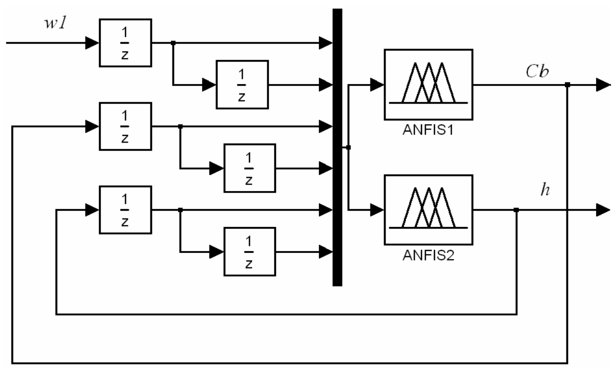

Fig.7: CSTR dynamic fuzzy model formed by two ANFISs

Table1: parameters of subtractive clustering, used in this research

parameter	value
Range of influence	0.5
Squash factor	1.1
Accept ratio	0.3
Reject ratio	0.1

After subtractive clustering, the model is trained only three epochs. It is observed, more rules and more training epochs does not have considerable positive effect on the accuracy of the model. In both Sugeno-type fuzzy inference systems; AND operator is used between different fuzzy values of antecedents; product is used as AND.
To be compared with complete fuzzy model, an incomplete fuzzy model is also designed according to (18). This fuzzy inference system has 11 rules derived from subtractive clustering.

7. SIMULATION RESULTS

In this research, two different series of checking data are used. Both series are entirely different from training data. Figure 8 shows the responses of actual system and different models for the first series of data. Outlet concentration (C_b) is considered as the main output of the system for control. It is observed, as stated in sec.8, as the height is ignored, the accuracy decreases significantly.
A criterion is defined for the predictive accuracy of models, namely PAN:

$$\text{PAN}=\sum_{i=1}^{N}|\hat{C}_b(i)-C_b(i)|\,. \tag{20}$$

Table 2 shows PA10 and PA30 (the sum of absolute error of prediction for 10 and 30 future instants or next 2 or 6 seconds), for two different series of checking data and four different models.

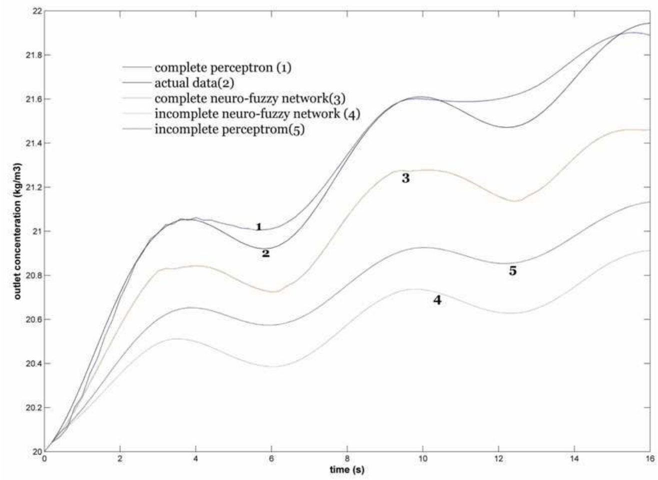

Fig.8: Actual and predicted data for the first series of checking data

Table2: Prediction accuracy for different trained models

Criterion	PA10 (kg/m^3)		PA30(kg/m^3)	
Checking data	1st series	2nd series	1st series	2nd series
Incomplete Neuro-fuzzy model (one ANFIS)	0.313	0.154	5.568	6.508
Incomplete Perceptron (single output)	0. 113	0.131	4.533	5.059
Complete Neuro-fuzzy model (two ANFIS)	0.060	0.036	0.752	0.658
Complete Perceptron (double output)	0.018	0.022	0.051	0.033

As previously stated in section4, in checking the previously estimated data should be used rather than the recorded outputs, otherwise, other mistakes may be hidden by false high accuracy of the model. For example, using (21) instead of (18):

$$\hat{C}_b(k+1)=F[w_1(k-1),w_1(k),C_b(k-1),C_b(k)], \tag{21}$$

leads a very high accuracy. So, the effect of height neglect is not observed. Table 3 shows the response of incomplete neural model, in case that Eq.(29) is exerted.

8. CONCLUSION

In this paper, modelling of dynamic systems for neuro-predictive purposes is studied. As a basis of nonlinear predictive control, accurate and reliable nonlinear models should be available to predict the systems' output(s) (section 5). Perceptrons and neuro-fuzzy networks (made by subtractive clustering) are utilized to perform this task, and a nonlinear Catalytic Continuous Stirred Tank Reactor (CSTR) is used as the case study. Training and checking data are not experimental and they are originally gathered using a computer model. This paper shows the capabilities of intelligent tools in the prediction of the behaviour of nonlinear MIMO process plants. Also, it is indicated that a variety of subtle points should be paid attention both in intelligent modelling and the validation of achieved models, especially for predictive control purposes, otherwise, the predicted values are not reliable and the predictive control system looses its efficiency. Some of the most important points in intelligent modelling for predictive control purposes are listed below:

1) Fixed inputs of systems can be ignored in modelling, providing that they remain fixed while the model is used for prediction (sections 2 and 6).
2) The magnitude of numbers which represent different signals (temperature, pressure or …) should be close to each other. Normalization of all the data columns is a suitable alternative to guarantee this matter (sections 2 and 6).
3) Dynamic systems (including all process plants) can be modelled only by recurrent models (sections 2 and 6).
4) In checking/validating of models, apart from very first instants, previously estimated values of system's outputs should be used as the input(s) of model, and the recorded data of output(s) should be utilized just for comparison (sections 3 and 7).
5) Even if only one of the outputs of the system is going to be predicted, the model should be able to estimate all the outputs (outputs of systems are usually coupled, sections 6 and 7).

These points can be useful only in case the inputs and outputs of systems are defined properly (sections 2 and 5), no matter which intelligent tool (sections 1 and 2) is utilized.

10. REFRENCES

Aggelogiannaki E., Sarimveis H., Dimitrios Koubogiannis, "Model predictive temperature control in long ducts by means of a neural network approximation tool", Applied Thermal Engineering 27 (2007) 2363–2369 .

Barros J., Dexter A., "On-line identification of computationally undemanding evolving fuzzy models", Fuzzy Sets and Systems 158 (2007) 1997 – 2012.

Bemporad A., Morari M., Ricker L., "Model Predictive Control2, User's Guide", Mathworks, March 2007.

Cervantes A., Agamennoni1 O., Figueroa J. , "A nonlinear model predictive control system based on Wiener piecewise linear models", Journal of Process Control 13 (2003) 655–666.

Demuth H., Beale M., Hagan M. "Neural Networks Toolbox 5, User's Guide", The MathWorks, March 2007, Online.

Feng L., Wang J., Poh E., "Improved robust model predictive control with structured uncertainty", Journal of Process Control 17 (2007) 683–688.

Fernández Camacho E., Bordons C., **"**Model Predictive Control **"** Springer**,** 2004

Ghaffari A., Mehrabian A., and Mohammad-Zaheri M., "Identification and Control of Power Plant De-Super Heater Using Soft Computing Techniques," Engineering Applications of artificial Intelligence, Special Issue in Applications of A.I. in Process Engineering, vol. 20, no. 2, March 2007, pp. 273-287.

Harnischmacher G., Marquardt, "A multi-variate Hammerstein model for processes with input directionality", Journal of Process Control 17 (2007) 539–550.

Haykin S., "Neural Networks a Comprehensive Foundation", 2nd edition, Prentice-Hall Inc. London, 1999.

Holenda B. , Domokos E., Redey A., Fazakas J., "Dissolved oxygen control of the activated sludge wastewater treatment process using model predictive control", Computers and Chemical Engineering (article in press) .

Jang J. , Sun C., Mizutani E. "Neuro-Fuzzy and Soft Computing". Prentice-Hall of India, New Delhi, 2006

Karer G., Music G., Skrjanc I., Zupancic B., "Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model", Nonlinear Analysis: Hybrid Systems (article in press).

Li M., Christofides P.," An input/output approach to the optimal transition control of a class of distributed chemical reactors", Chemical Engineering Science 62 (2007) 2979 – 2988 .

Magni L., R. Scattolini, "Tracking of non-square nonlinear continuous time systems with piecewise constant model predictive control", Journal of Process Control 17 (2007) 631–640.

Mohammadzaheri M., Chen L., "Design of an Intelligent Controller for a Model Helicopter Using Neuro-Predictive Method with Fuzzy Compensation", World Congress of Engineering, London, 2~4 July 2007.

Na M., Upadhyaya B., "Application of model predictive control strategy based on fuzzy identification to an SP-100 space reactor" Annals of Nuclear Energy 33 (2006) 1467–1478 .

Nagy Z., Mahnb B., Frankec R., Allgower F. , "Evaluation study of an efficient output feedback nonlinear model predictive control for temperature tracking in an industrial batch reactor", Control Engineering Practice 15 (2007) 839–850 .

Seyab R., Cao Y., "Differential Recurrent Neural Network based Predictive Control", Computers and Chemical Engineering (article in press) .