10th International Research/Expert Conference
”Trends in the Development of Machinery and Associated Technology”
TMT 2006, Barcelona-Lloret de Mar, Spain, 11-15 September, 2006

NEURAL NETWORKS IN MODEL PREDICTIVE CONTROL

David Samek
Faculty of Technology, Tomas Bata University in Zlin
Nam. T. G. Masaryka 275, 76272, Zlin
Czech Republic

ABSTRACT
The contribution is aimed at predictive control of nonlinear processes with the help of artificial neural networks as the predictor. Since this methodology is relatively wide, paper only concentrates on the prediction via artificial neural networks. Special attention is paid to the usage of offline-learnt predictor based on multilayer feed forward neural network. The proposed method is tested in simulations on a nonlinear system.
Keywords: artificial neural networks, prediction, predictive control

1. INTRODUCTION
Model predictive controllers (MPC) make use of a model of the plant to be controlled in order to predict the future output values. The manipulated signal is obtained by optimizing a performance criterion that penalizes the deviation from a desired future trajectory, the control effort and other undesired phenomena. A key feature of the MPC schemes (linear and nonlinear) is the necessity of a reliable model of the plant being controlled. However, the methodology of MPC for linear plants has to be changed when coping with nonlinear behaviour, because the superposition principle does not hold [1]. A primary advantage to the approach is the explicit handling of constraints. In addition, the formulation for multivariable processes with time-delays is straightforward.
Classic approach in prediction is based on a set upping of mathematical model of process. The deterministic model is usually concerned, but in case of stochastic process it is necessary to use the stochastic model. Many predictive control techniques are established on this approach mainly based on the assumption that the process to be controlled can be regarded as linear and that its model is available a priori.
The disadvantage of this approach is the necessity to know process, its parameters and to describe them. Moreover, difficulties rise during the prediction of strongly nonlinear processes and eventual control performance is not satisfactory.
Artificial neural networks (ANNs) offer interesting possibility for modelling and predicting any nonlinear process without a priori knowledge. ANNs can be regarded as nonlinear black box models. The most used type of ANN for predictive control is multilayer feed forward neural network [2, 3].

2. MODEL PREDICTIVE CONTROL USING ANN
These methods are very similar to the classical MPC. Of course, there is just one important difference in model which is realized by ANN. This model predicts future performance of controlled process as a response to potential control actions. Optimization algorithm calculates sequence of control actions which minimise error between predicted output signal and reference trajectory according to the objective function. The first task of MPC using artificial neural networks is to obtain appropriate model of process to be predicted. Then this model is used by controller during optimization of control actions for the future outputs prediction. Whereas, the cost function J could be generally expressed:

$$J = \sum_{i=N_1}^{N_2} \lambda(i) \cdot \left[y_r(k+i) - \hat{y}(k+i) \right]^2 + \sum_{i=1}^{N_u} \rho(i) \cdot \left[u_t(k+i-1) - u_t(k+i-2) \right]^2 \qquad (1)$$

where λ and ρ is penalty for the first and the second sum respectively, N_1 a N_2 determine interval for the prediction error minimisation, horizon N_u determines number of steps to control effort minimisation, y_r is reference signal, $\hat{y}$ is prediction of output signal and u_t is tentative control signal.

There is usually assumed:

$$N_2 \geq N_u \tag{2}$$

and

$$\Delta u(k+i) = 0 \quad for \quad i \in \langle N_u, N_2 - 1 \rangle \tag{3}$$

For processes with non-minimum phase it is useful to set $N_1>1$, because first N_1 elements are ignored. However usually is used $N_1=1$ and $N_2=N_u$. Generally λ and ρ depends on i, but in practice they are often assumed as constant or exponential, example for λ:

$$\lambda(i) = \alpha^{N_2 - i} \tag{4}$$

If $\alpha \in \left(0,1\right\rangle$, the errors farthest from the current instant k are the most penalized. On the other hand, for $\alpha > 1$ the first errors are more penalized. Sometimes only ρ is used for determining the ratio between the summations (λ=1 for all i).

One of the most important advantages of MPC is that if the future reference signal is known a priori, the system is able to react before the change has effectively been made, thus avoiding the effects of delay in process responses. This condition is fulfilled true in many applications such as robotics, servos and batch processes. Nevertheless, most of the methods use just approximation of reference trajectory by first order system [1] or constant value of y_r is assumed along the prediction horizon [4].

3. ARTIFICIAL NEURAL NETWORKS USED FOR PREDICTION

Prediction by ANN is based on ability of some artificial neural networks to model certain (controlled) process and modelling is nothing more than approximation of certain function (input-output function). There are many types of ANN's suitable for modelling and control. However, because our simulations utilised only the multilayer feed forward neural networks and because limited space, this chapter is focused on them.

3.1. Multilayer feed forward neural networks

Multilayer feed forward neural networks (MFFNN) were derived by generalization from Rosenblatt's perceptron, thus they are often called multilayer perceptrons (MLP). These networks are trained (= learnt) by supervised learning algorithms, whereas the basic method is backpropagation algorithm. Therefore, sometimes MFFNNs are also called backpropagation networks.

In the MFFNN the information flows between the neurons only in the forward direction i.e. towards the output end. Neurons of each layer can have inputs from any neurons of the earlier layer. Each neuron is characterized by the generally nonlinear transfer function S and by the threshold value b. The neuron sums the weighted inputs and the threshold, and passes the result through its characteristic transfer function. The transfer function is usually same for all neurons from the layer.

Weights are commonly labelled $w_{\text{number of layer}}$(source neuron, target neuron), thresholds likewise. Values can be arranged into matrixes and the function of the three-layer neural network can be written:

$$\boldsymbol{y}_{out} = S_3\left(\boldsymbol{b}_3 + \boldsymbol{W}_3 \cdot \boldsymbol{x}_2\right) \tag{5}$$

$$\boldsymbol{x}_2 = S_2\left(\boldsymbol{b}_2 + \boldsymbol{W}_2 \cdot \boldsymbol{x}_1\right) \tag{6}$$

$$\boldsymbol{x}_1 = S_1\left(\boldsymbol{b}_1 + \boldsymbol{W}_1 \cdot \boldsymbol{u}_{in}\right) \tag{7}$$

where $\boldsymbol{y}_{out}$ is output vector of the MFFNN, S_i is transfer function of the i-th layer, $\boldsymbol{b}_i$ is bias vector of the i-th layer, $\boldsymbol{W}_i$ is weighting matrix of the i-th layer, $\boldsymbol{x}_i$ is output vector of the i-th layer and $\boldsymbol{u}_{in}$ is input vector of the MFFNN.

4. METHODOLOGY

Let's consider SISO nonlinear system to be controlled which is shown at Figure 1 and consist two connected ball tanks for liquid.

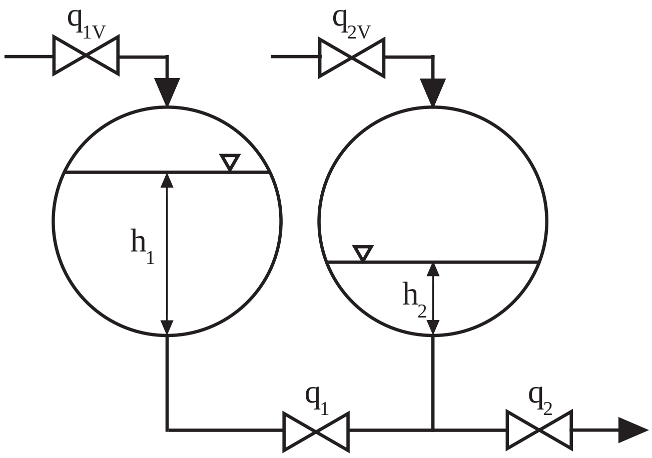

Figure 1. Two connected tanks for liquid

Mathematical model of this system can be written in applying usual simplifications by these differential equations:

$$\pi h_1(d_1 - h_1)\frac{dh_1}{dt} + q_1 = q_{1v} \quad (8)$$

$$\pi h_2(d_2 - h_2)\frac{dh_2}{dt} - q_1 + q_2 = q_{2v} \quad (9)$$

where d_j are diameters of tanks, h_j are liquid levels in tanks, q_j are output volume rates of flow and q_{jv} are input volume rates of flow, (for j = 1, 2). The output volume rates of flow depend on the liquid levels:

$$q_1 = k_1\sqrt{|h_1 - h_2|} \quad (\text{if} \quad h_1 - h_2 < 0 \quad \text{then} \quad q_1 = -q_1) \quad (10)$$

$$q_2 = k_2\sqrt{h_2} \quad (11)$$

where k_1 and k_2 are constants representing pipeline properties.

The initial conditions of the equations (8) and (9) are $h_1(0)$ = 1.5m^3/min and $h_2(0)$ = 1.3m^3/min. The parameters of the system model were d_1=d_2=2m, k_1 = 0.85m$^{2.5}$/min, k_2 = 0.5m$^{2.5}$/min, $q_{1v}(0)$ = 0.38m^3/min and q_{2v} = 0.19m^3/min=constant. We have regarded this SISO system output y as h_1 and control action u as q_{1v} which was constrained from 0.1 to 1m^3/min.

All simulations have been done in Matlab 6.1 using Neural Network Toolbox and Simulink. Adopting all above mentioned assumptions, we can simplify the equation (1) to the version which is used in this paper:

$$J = \sum_{i=N_1}^{N_2}\left[y_r(k+i) - \hat{y}(k+i)\right]^2 + \rho\cdot\sum_{i=1}^{N_u}\left[u_t(k+i-1) - u_t(k+i-2)\right]^2 \quad (12)$$

The off-line trained 3-layer MFFNN model with architecture 10-25-5-1 (into ANN's input comes $u(k)$, $u(k\text{-}1)$, …, $u(k\text{-}4)$, $y(k)$, $y(k\text{-}1)$, …, $y(k\text{-}4)$) has been created as the ANN predictor. The hyperbolic tangent sigmoid was used as the transfer function of all neurons. At first, the controlled process was initialized by identification signal and process output was saved. Training data then consisted pairs of network input (values $u(k)$, … and $y(k)$ …) - required network output ($y(k+1)$). As the learning method was applied Levenberg-Marquardt algorithm.

The proper selection of training data (in this case it means selection of identification signal) is fundamental. If an ANN is trained for improper data, which poorly describes solved problem, the ANN wouldn't work properly.

5. SIMULATIONS AND RESULTS

There were studied two cases without and with immeasurable noise. Without influence of noise there was set ρ=0.8 and N_1=1 at equation (12) so that prediction interval starts at instant k+1. Selection of N_2 and N_u is rather intuitive; the control is unsatisfactory for too small values of these parameters. On the other hand, computational demands considerably grow up for too long prediction horizon. After some experiments N_2=20 a N_u=5 were applied.

In the case of noise influence, there was set ρ=1.3 to decrease an overshoot of system output. Start-point of prediction interval remains same (N_1=1). With the aim of reducing oscillations of control signal and minimize error N_2=5 a N_u=2 were applied.

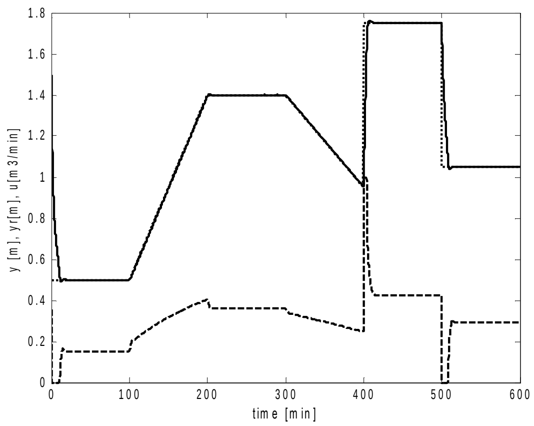

Figure 2. Response of the two connected ball tank system and control action (dotted line – reference signal, solid line - system output, dashed line - control signal)

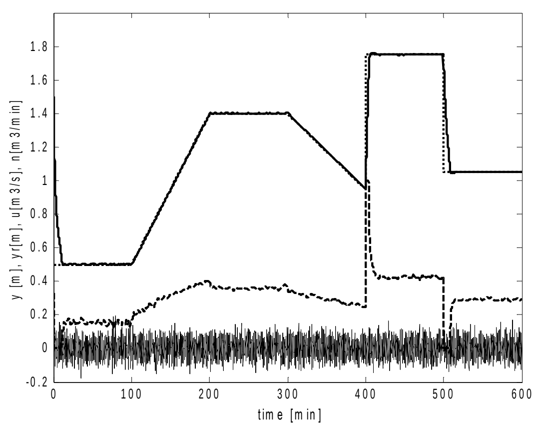

Figure 3. Response of the two connected ball tank system, control action and noise (dotted line – reference signal, solid thick line - system output, dashed line - control signal, solid thin line - noise)

6. CONCLUSION

This paper has given a brief introduction to the use of artificial neural networks in predictive control. We have selected one type of many different approaches, which has used multilayer feed forward ANN. We have also demonstrated the capabilities of this network for predictive control by presented simulation of nonlinear SISO system control. This approach has many advantages, for instance the elimination of problems with linearization, the possibility to design ANN to predict multiply predictions at a time and, of course, the opportunity to create MIMO predictor if they needed. Unfortunately, it has also some disadvantages, such as the off-line modelling and the higher computational requirements. Future work will be devoted to development of on-line ANN model.

7. ACKNOLEDGEMENT

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic under grant MSM 7088352102. This support is very gratefully acknowledged.

8. REFERENCES

[1] Camacho E.F., Bordons A.C.: Model Predictive Control in the Process Industry, Springer. New York, 1995. ISBN 978354019924.

[2] Kanjilal P.P.: Adaptive prediction and predictive control, Peter Perengrinus Ltd., London, 1995. ISBN 0863411932.

[3] Hagan M., Demuth H., Jesus O. D.: An Introduction to the Use of Neural Networks in Control Systems. International Journal of Robust and Nonlinear Control, vol. 12, no. 11, p. 959-985, ISSN 1049-8923, John Wiley & Sons, Inc., 2002.

[4] Bobal V., Bohm J., Fessl, J. Machacek, J.: Digital Self-tuning Controllers: Algorithms, Implementation and Applications, Springer. London, 2005. ISBN 1-85233-980-2.