Robust Control of Continuous Polymerization Reactor by Dynamically Constructed Recurrent Fuzzy Neural Network

Y. Frayman and L. Wang
Deakin University, School of Computing and Mathematics,
662 Blackburn Road, Clayton, Victoria 3168, Australia
E-mail: yfraym@deakin.edu.au, lwang@deakin.edu.au

ABSTRACT

Improvements in control of the reactor temperature and the number-average molecular weight of the polymer products have been motivated by increasing environmental, operational and energy restrictions. Conventional control systems for continuous stirred tank polymerization reactor are based on the single-input-single-output design principle. The polymerization process, however, is a typical multivariable system with a highly nonlinear and highly interacting behavior. Consequently, a new design strategy is necessary to overcome the problem of mutual interactions and is indispensable for the improvement of polymerization reactor control. In order to solve this problem, we propose to use a recurrent fuzzy neural network dynamically constructed from the process input-output data and a reference model for direct adaptive control. The effectiveness of our approach is demonstrated by simulation results and is compared with the performance of conventional PI controllers. Exploiting the advantage of multivariable nonlinear control, our dynamically constructed FNN significantly increases the control robustness to disturbances and noise compared to linear PI controllers.

Keywords: direct adaptive control, recurrent fuzzy neural networks, dynamic network construction, continuous stirred tank polymerization reactor control

1. Introduction

Chemical processes are multivariable in nature, and exhibit strongly nonlinear and time varying behavior. There are also high mutual interactions between process variables. Conventional control of chemical processes uses simple linear or linearized models to approximate the process behavior. For multi-input-multi-output (MIMO) processes it is very difficult to derive accurate models, due to complex nonlinear relationships among variables, time dependent changes in model parameters, and difficulties in accounting for some physical phenomena. Thus, the severe nonlinearity and complexity of the process results in large robustness margins, and in some cases, extremely poor performance.

It is therefore necessary to develop solid control methodologies that are capable of coping with both nonlinearities and interactions, as well as time varying processes with a strong influence of disturbances on the process behavior.

In addition, the nonlinear control schemes that employ more realistic and more complex nonlinear process descriptions require process models in the form of nonlinear differential equations [6]. This limits its industrial application, since such first principles models are not readily available in industrial practice due to a chronic lack of detailed and extensive process knowledge required for the development of these models.

We are especially interested in control of the nonlinear MIMO processes in the presence of disturbances, such as the continuous stirred tank polymerization reactor, with a minimum amount of human intervention. Therefore, it is desirable to integrate an intelligent component to increase control system flexibility, e.g., to extract relations from the process, and to change relations to improve control performance. Adaptive intelligent controllers using process input-output data with both structural and parameter tuning, such as fuzzy neural networks (FNNs), would fulfill the above objectives.

Most of the existing research in neural or fuzzy neural control has been concentrated on indirect control schemes where the neural network or the fuzzy neural network is used to identify the process and a controller is subsequently synthesized from this model [14]. We follow an alternative approach of direct model reference adaptive control (MRAC) where a fuzzy neural network is the controller and no model of the process is required. This method has already been successfully applied to torsional vibration control of tandem cold rolling mill spindles [8], and to backup rolls eccentricity and thickness control in cold rolling mills [9], [10].

Similarly to neural networks, FNNs require two type of tuning: structural and parameter tuning. Structural tuning concerns with the structure of the rules: the number of input-output variables, partition of each variable universe of discourse, the number of

rules, and the logical operation to compose rules. In other existing recurrent FNN approaches (e.g., [11], [12]) these selections are carried out on a trial-and-error basis. Parameter tuning concerns with adjustments of the position and the shape of membership functions. While most of existing efforts are concentrated on parameter tuning, insufficient efforts have been made concerning structural tuning, i.e., to find the simplest network structure capable of achieving an optimal performance.

A direct MRAC based on our dynamically constructed recurrent FNN offers a method for automatic discovery of an efficient controller. Such an approach is able to achieve good robustness in time varying environments through continuous adaptation. Local learning of our FNN is different from general and indirect methods commonly used in neural control in that our FNN controller learns from a direct evaluation of accuracy with respect to the outputs of the process rather than from the inputs and the outputs of the controller.

The overall network is a nonlinear function approximator which is linear in parameters with each combination of input node, rule node and output node acts as a linear approximator. A major advantage of approximators that are linear in parameters is that for square error types of cost functions, as the one we use, there is a unique global minimum [7].

The paper is structured as follows. Section 2 describes the dynamically constructed recurrent FNN, its learning algorithm, and the overall control system structure. Section 3 describes the problem to be studied - continuous stirred tank polymerization reactor control. Section 4 presents the simulation results using the dynamically constructed recurrent FNN controller and PI controllers to control of the reactor temperature and the number-average molecular weight of the polymer product. Finally, section 5 summarizes the main findings of this paper.

2. Dynamically Constructed FNN for Direct Adaptive Multivariable Control

A general MIMO nonlinear processes can be represented by the following state-space description

$$\vec{x}(k+1) = \vec{f}[\vec{x}(k), \vec{u}(k), \vec{d}(k)] \ , \qquad (1)$$

$$\vec{y}(k) = \vec{g}[\vec{x}(k), n(k)] \ . \qquad (2)$$

where $\vec{u}$ $(u_1, u_2, .., u_m)$, $\vec{y}$ $(y_1, y_2, .., y_m)$ and $\vec{d}$ $(d_1, d_2, .., d_p)$ are the process input, output and disturbance input vectors, respectively, $\vec{x}$ $(x_1, x_2, .., x_n)$ is the process state vector, and $\vec{n}$ $(n_1, n_2, .., n_m)$ is the measurement noise vector. Given that the vector maps $\vec{f}$ and $\vec{g}$ are unknown, a dynamically constructed recurrent FNN can be used to control the process by assuming only that the states and corresponding outputs of the process are measurable.

A reference model with input x and desired output $\vec{y}_d$ is used to designate the desired performance. The desired output response $\vec{y}_d$ is obtained from the output of the reference model subject to disturbance signals $\vec{d}$. The parameters of the model are determined by the desired performance in terms of overshoot, settling time, and steady-state error.

The learning algorithm is designed to obtain the correct control signals $\vec{u}$ corresponding to the desired outputs $\vec{y}_d$. The learning error $\vec{\varepsilon}$ defined as the difference between the desired responses $\vec{y}_d$ and the measured process outputs $\vec{y}$, is used as a learning criterion. The learning error $\vec{\varepsilon}$ asymptotically approaches zero or a pre-specified small value > 0 as the iteration number k increases. The objective is to minimize an square error cost function.

The structure of our FNN is shown in Fig. 1 [9]. The network has an input layer, a input membership functions layer, a rule layer, and an output layer.

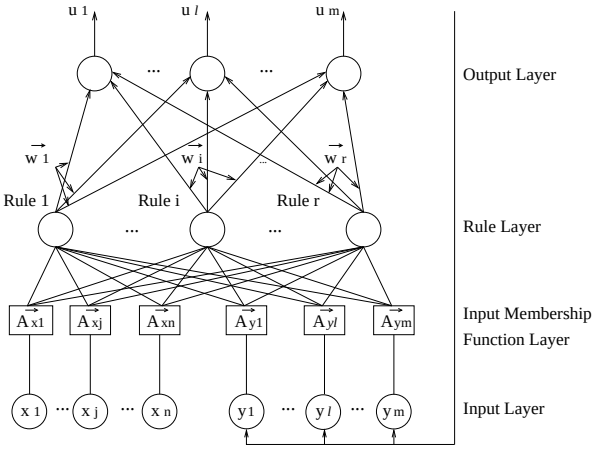

Fig. 1: The structure of our fuzzy neural network.

The input nodes represent input variables consisting of the current network inputs (process states) and the previous outputs of the process, rather than the network. This recurrent structure provides the possibility to include temporal information, i.e. the network produces dynamic input-output mapping, in contrast to static feedforward networks. The recurrency also speeds up the convergence of the network. An important feature of the recurrent structure of our FNN is that a convergence to a stable solution is guaranteed, as there is no feedback path between network outputs and inputs [15].

The input membership function nodes generate input membership functions for numerical inputs. We use piecewise-linear triangular membership functions. This type of membership functions is simple to implement and is computationally efficient. The leftmost and rightmost membership functions are shouldered. Each input node is connected to all membership function nodes for this input. The input membership functions act as fuzzy weights between the input layer and the rule layer.

Rule layer nodes represent fuzzy rules using the

following form for rule i:

$$\text{Rule} \quad i: \text{ IF } x_1 \text{ is } A^i_{x_1} \text{ and } ... \; x_n \text{ is } A^i_{x_n}$$
$$\text{and } y_1(k-1) \text{ is } A^i_{y_1} \text{ and } ... \; y_m(k-r) \text{ is } A^i_{y_m} \quad (3)$$
$$\text{THEN } u_1 = w^i_1 \; , \; ... \; , \; u_m = w^i_m \; ,$$

where w^i_l $(l = 1, 2, .., m)$ is a real number. A^i_q $(q = x_1, , .., x_n, y_1, , .., y_m)$ is the membership function of the antecedent part of rule i for input node q, k is the time, and z $(z = 1, 2, ..., r)$ is the delay. Each rule node is connected to all input membership function nodes and output nodes for this rule. The membership value μ_i of the premise of the ith rule, is calculated as fuzzy AND using the product operator

$$\mu_i = A^i_{x1}(x_1) \cdot .... \cdot A^i_{xn}(x_n) \cdot A^i_{y1}(y_1) \cdot .... \cdot A^i_{ym}(y_m) \; . \quad (4)$$

The use of the product operator for fuzzy AND produces a smoother control surface, in contrast to commonly used fuzzy *min* operator. Links between the rule layer, the output layer and the input membership functions are adaptive during learning.

In the output layer each node receives inputs from all rule nodes connected to this output node and produces the actual output of the FNN. Output u_l of the FNN is obtained using the weighted average

$$u_l = \frac{\sum_i \mu_i w^i_l}{\sum_i \mu_i} \; . \quad (5)$$

Use of the weighted average allow us to avoid problems with the commonly used center of area (COA) defuzzification that can produce unpredictable results.

A block diagram of the overall FNN control system is presented in Fig. 2 [9].

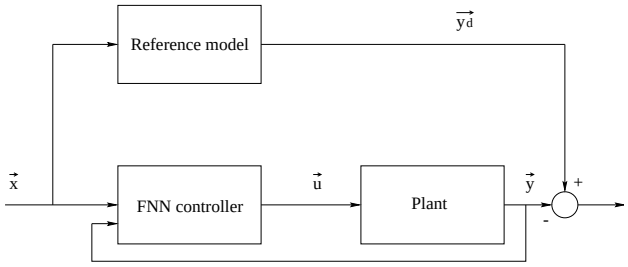

Fig. 2: Block diagram of the overall FNN control system.

The FNN structure generation and parameter tuning algorithms are as follows. We generate the training input-output data for the selected reference model. We need to specify the allowable error threshold ε and/or the maximum number of rules (rule nodes) N_i for learning to stop.

Initially we add two equally spaced fully overlapping shouldered input membership functions along the operating range of each input variable. In such a way these membership functions satisfy ϵ-completeness [9]. If the ϵ-completeness is not satisfied, there may be no rule applicable for a new data input. The initial rule layer is created using Eq. (3).

The network is trained using the following learning rules

$$w^i_l(k+1) = w^i_l(k) - \eta \frac{\partial \varepsilon_l}{\partial w^i_l} \; . \quad (6)$$

$$A^i_q(k+1) = A^i_q(k) - \eta \frac{\partial \varepsilon_l}{\partial A^i_q} \; , \quad (7)$$

where η is the learning rate.

The learning rate η varies to improve the speed of convergence, as well as the learning performance (accuracy).

If the degree of overlapping of membership functions is greater than a pre-specified threshold, we combine those membership functions. We use the following *fuzzy similarity measure* [5]

$$E(A_1, A_2) = \frac{M(A_1 \cap A_2)}{M(A_1 \cup A_2)} \; , \quad (8)$$

where $\cap$ and $\cup$ denote the intersection and the union of two fuzzy sets A_1 and A_2, respectively, and $E(A_1, A_2)$ is a degree of $A_1 = A_2$. $M(\cdot)$ is the size of a fuzzy set, and $0 \leq E(A_1, A_2) \leq 1$. We can thus reduce the size of the rule base, which is necessary in order to protect the network from the "curse of dimensionality".

If the accuracy of the FNN is satisfactory, the algorithm stops. Otherwise, if the number of rule nodes is less than the specified maximum, we add an additional membership function for all inputs at the point of the maximum system output error. In such a way we are able to reduce the error more efficiently. By firstly eliminating the errors whose deviation from the target values is the greatest, we can speed up the convergence of the network substantially. Next, the rule base is updated and the process is repeated until either we obtained a satisfactory performance, or the maximum pre-specified size of the network (number of rules) is exceeded.

3. Problem Description

Tight control of polymer properties is one of the major problems in polymerization processes. In free-radical polymerization processes, controlling both reactor temperature and the number-average molecular weight is of primary importance. This is achieved by manipulating the volumetric flow rates of the initiator and the cooling water into the reactor.

Conventional control of polymerization reactors is based on the single-input-single-output design principles. Until recently, control was implemented using standard PI or PID controllers. The polymerization process, however, is a typical multivariable system with strong mutual interaction between process variables. A new design strategy is thus necessary to overcome the problem of the mutual interaction and is important for the improvement of control accuracy and robustness to disturbances and noise. As the chemical industry requires a better product quality under constantly increasing environmental, operational and

energy restrictions, a viable scheme for control is of substantial interest to the industry. Consequently, considerable industrial research effort [3], [4], [13] has been devoted to finding the best possible solution.

The presence of the computer in control loops enables us to investigate the use of more sophisticated control methods such as fuzzy neural networks. We propose to use a recurrent FNN controller dynamically constructed from the process input-output data and a reference model for direct MRAC. The perceived benefits of using FNN include reductions in commissioning time, better-tuned control over a diverse polymer product range, improved control over changes of process characteristics during polymerization, the ability to control nonlinear effects in control loops, and robustness to process disturbances and noise.

The continuous stirred tank polymerization reactor (CSTR) is shown in Fig. 3 [3]. A free-radical polymerization of methyl methacrylate (MMA) takes place in the rector, with azo-bis-isobutyronitrile (AIBN) as initiator and toluene as solvent. The reaction is exotermic and a cooling jacket allows the heat removal. The standard mechanism of free-radical polymerization is assumed, together with the resulting rate laws [2].

The following assumptions were made: perfect mixing of the reactor contents; constant density of the reacting mixture (no volume shrinkage); gel effect is absent (due to low monomer conversion); constant reactor volume (constant volumetric flow rate of the monomer stream); no polymer in the fluids in inlet streams; constant heat capacity of the reacting mixture; uniform coolant temperature in the jacket; insulated reactor and cooling system; constant heat capacity of the coolant; negligible initiator solution flow rate in comparison to that of the monomer stream; negligible inhibition and chain transfer to solvent reactions; quasi-steady state and long-chain hypothesis.

The dynamic behaviour of the process is described by the mass and energy balances as a set of ordinary differential equations [2], [3]. Consult [3] for the ordinary differential equations involved and the kinetic data, physical parameters, and steady-state values used.

The dimensionless state variables are defined as follows: $x_1 = C_m, x_2 = C_I, x_3 = T, x_4 = D_0, x_5 = D_I, x_6 = T_j$ (Fig. 3). The control of a polymerization reactor requires the number-average molecular weight (NAMW) $y_1 = D_I/D_0$ and the reactor temperature $y_2 = T$ to be regulated. This is achieved by manipulating the inlet initiator volumetric flow rate $u_1 = F_I$ and the cooling water volumetric flow rate $u_2 = F_{cw}$. The process disturbances are the molar concentration of monomer in the inlet stream $d_1 = C_{m_{in}}$ and the temperature of the inlet stream $d_2 = T_{in}$.

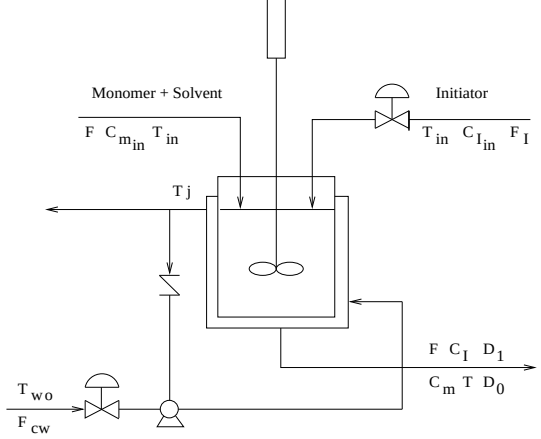

Fig. 3: Continuous stirred tank polymerization reactor. Here F is the volumetric flow rate of the inlet and outlet streams, $C_{m_{in}}$ is the molar concentration of monomer in inlet stream, T_{in} is temperature of the inlet streams, $C_{I_{in}}$ is the molar concentration of the initiator in inlet stream, F_I is the volumetric flow rate of the inlet initiator stream, T_{w_O} is the temperature of the inlet coolant stream, F_{cw} is the volumetric flow rate of the cooling water, C_I is the molar concentration of the initiator, D_1 is the mass concentration of the dead polymer chains, C_m is the molar concentration of the monomer, T is the reactor temperature, D_O is the molar concentration of the dead polymer chains, and T_j is the cooling jacket temperature.

4. Simulation Results

The robustness to disturbances and noise of the dynamically constructed recurrent FNN was evaluated by numerical simulations.

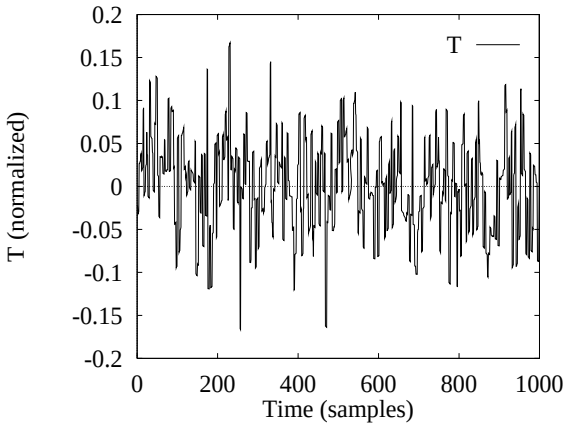

Fig. 4: Time responses for uncontrolled reactor temperature (T) in the presence of molar concentration of monomer and the temperature in inlet steam disturbances, and the measurement noise.

We used as a reference model Butterworth's [1] characteristic equation for the 5th order system

$$s^5 + 3.24\omega_n s^4 + 5.24\omega_n^2 s^3 + 5.24\omega_n^3 s^2 + 3.24\omega_n^4 s + \omega_n^5 \quad (9)$$

where ω_n is a natural frequency of the system. This form of characteristic equation gives us a damping ratio $\xi = 0.71$, and the settling time can be determined through approximate relationship $t_s = 4/\xi\omega_n$.

We used the input-output data generated using above reference model to train our FNN. We have

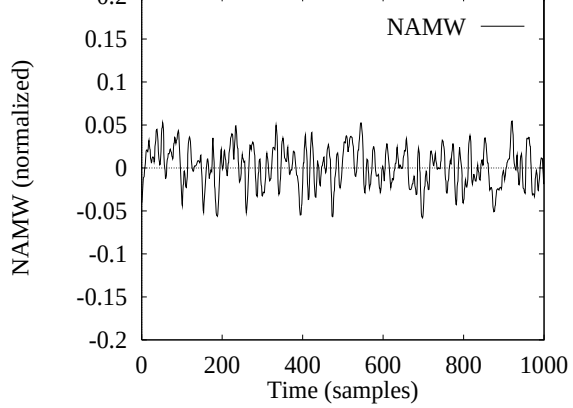

Fig. 5: Time responses for uncontrolled number-average molecular weight (NAMW) in the presence of molar concentration of monomer and the temperature in inlet steam disturbances, and the measurement noise.

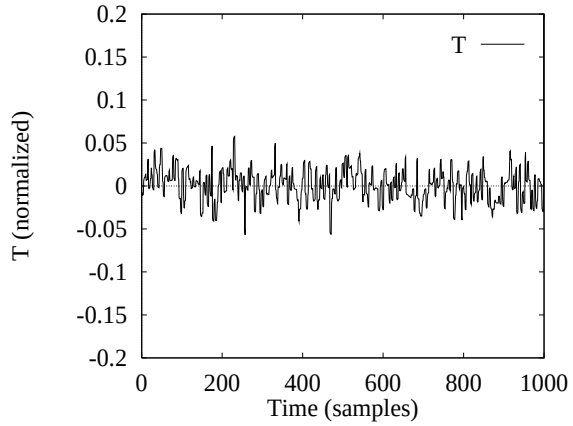

Fig. 6: Time responses for reactor temperature (T) in the presence of molar concentration of monomer and the temperature in inlet steam disturbances, and the measurement noise with PI controllers.

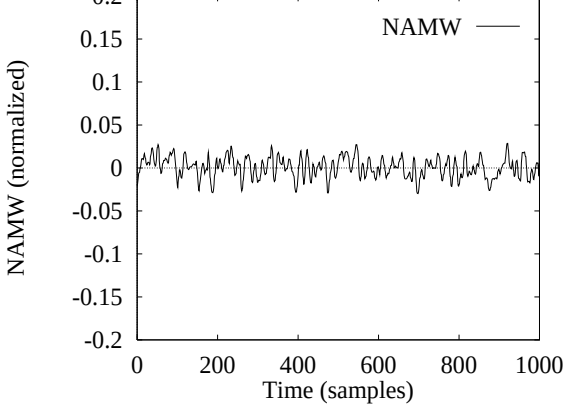

Fig. 7: Time responses for the number-average molecular weight (NAMW) in the presence of molar concentration of monomer and the temperature in inlet steam disturbances, and the measurement noise with PI controllers.

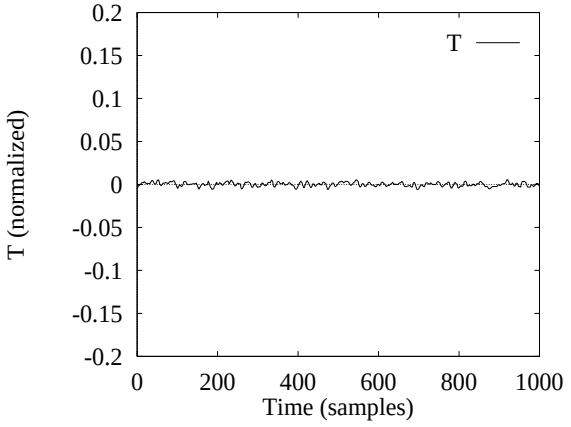

Fig. 8: Time responses for reactor temperature (T) in presence of molar concentration of monomer and the temperature in inlet steam disturbances, and the measurement noise with the FNN controller.

generated 1000 input-output data tuples using fifth order Runge-Kutta integrator [1] with a sampling time $h = 0.005$ for the numerical integration of the set of ordinary differential equations describing the process. In order to prevent numerical ill-conditioning, all equations were firstly appropriately normalized by dividing the deviation of each variable from its nominal operating value by its nominal operating value. Thus we obtained variables values that are zero at the nominal operating condition and have roughly equivalent ranges. We applied 25% square wave disturbance to the inlet monomer molar concentration $C_{m_{in}}$ and 10% square wave disturbance to inlet temperature T_{in} with a frequency 10 dimensionless units. We also applied zero-mean white measurement noise with a variance 0.006 and a sampling time 0.014. The disturbances and noise were applied concurrently to represent a real situation when the disturbances and noise in the polymerization reactor are present at the same time.

The FNN was trained using on-line learning using first 500 samples for training, the other 500 for validation of network's ability to generalize, and the whole data for final testing of a resulted network. For comparison two PI controllers were used for the same process. The parameters of the PI controller, i. e., the gain $K_c = 15$ and the integral time constant $\tau_I = 0.4$, were the same as used in [3].

Simulation results are presented in Fig. 4-9. While rejecting the process disturbances and noise to some extend, the performance of the PI controller is not totally satisfactory (Fig. 6 and 7). On the contrary, our FNN controller is able to reject the disturbances and noise almost completely (Fig. 8 and 9).

A quantitative summary of the results is given in Table 3.

	RMSE		
	Uncontrolled	PI	FNN
D_1/D_o	0.022981	0.011862	0.002755
T	0.055037	0.018452	0.002302

Table: 1: Root Mean Squared Error (RMSE) for the reactor temperature and the number-average molecular weight for uncontrolled, PI and FNN controlled cases.

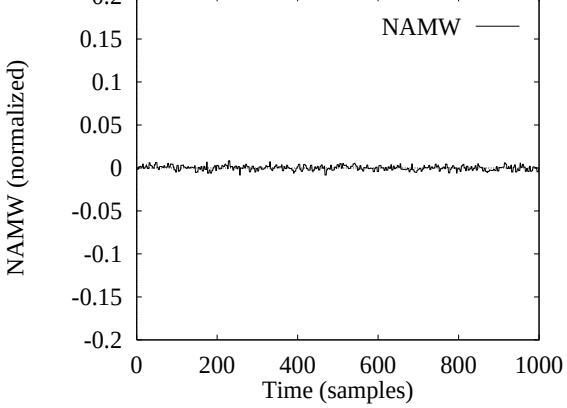

Fig. 9: Time responses for the number-average molecular weight (NAMW) in presence of molar concentration of monomer and the temperature in inlet steam disturbances, and the measurement noise with the FNN controller.

5. Conclusion

In this work we have discussed the benefits of using our dynamically constructed recurrent FNN for multivariable nonlinear control on a real-world example of control of polymerization reactor. The effectiveness of our approach was demonstrated by simulation results and is compared with the performance of the conventional PI controllers. Exploiting the advantage of nonlinear control, our dynamically constructed FNN significantly increases the control robustness compared to linear PI controllers. We argue that our dynamically constructed recurrent FNN with both structure and parameter learning can provide a computationally efficient solution to control of many real-world nonlinear systems in presence of disturbances.

References

[1] K. Åström and B. Wittenmark, *Computer controlled systems, theory and design*, 2nd ed., Prentice-Hall, 1990.

[2] J. P. Congalidis, J. R. Richards and W. H. Ray, "Feedforward and feedback control of a copolymerization reactor," AIChE Journal, vol. 35, 1989, pp. 891-907

[3] P. Daoutidis, M. Soroush and C. Kravaris, "Feedback / feedforward control of multivariable nonlinear processes," *AIChE Journal*, vol. 36, no. 10, 1990, pp. 1471-1484

[4] F. J. Doyle, B. A. Ogunnaike and R. K. Pearson, "Nonlinear model-based control using second-order Volterra models," *Automatica*, vol. 31, no. 5, 1995, pp. 697-714

[5] D. Dubois and H. Prade, "A unifying view of comparison indices in a fuzzy set theoretic framework," In R. R. Yager, (Ed.), *Fuzzy sets and possibility theory: recent developments*, Pergamon, NY, 1982

[6] J. Eaton and J. Rawlings, "Feedback control of chemical processes using on-line optimization techniques", *Comput. Chem. Eng.* vol. 14, 1990, pp. 469–479

[7] J. Farrell and W. Baker, "Learning control systems," In P. Antsaklis and K. Passino (Eds.), *Intelligent autonomous control systems*, Kluwer Academic, 1993

[8] Y. Frayman and L. Wang, "Torsional vibration control of tandem cold rolling mill spindles: a fuzzy neural approach," *Proc. of Australasia-Pacific Forum on Intelligent Processing and Manufacturing of Materials*, 1997, vol. 1, pp. 89-94.

[9] Y. Frayman and L. Wang, "Backup rolls eccentricity control in cold rolling mills using dynamically constructed recurrent fuzzy neural networks," *Proc. of 9th Australian Conference on Neural Networks*, 1998, pp. 26-30

[10] Y. Frayman and L. Wang, "Dynamically constructed recurrent fuzzy neural network for cold rolling mill thickness control," Submitted to *11th Australian Joint Conference on Artificial Intelligence*, 1998

[11] V. Gorrini and H. Bersini, "Recurrent fuzzy system," *Proc. Third IEEE Int. Conf. Fuzzy Systems*, 1994, pp.193-198

[12] E. Khan and F. Unal, "Recurrent fuzzy logic using neural networks," *Proc. 1994 IEEE Nagoya World Wisepersons Workshop*, 1994, pp. 48–55

[13] C. Kravaris and M. Soroush M., "Synthesis of multivariable nonlinear controllers by input / output linearization," *AIChE Journal*, vol. 36, no. 2, 1990, pp. 249-264

[14] K. Warwick, C. Kambhampati, P. Parks, and J. Mason, "Dynamic systems in neural networks," in K. Hunt, G. Irwin, and K. Warwick (Eds.), *Neural Network Engineering in Dynamic Control Systems*, Springer-Verlag, 1995, pp. 27-41

[15] P. D. Wasserman, *Advanced Methods in Neural Computing*, Van Nostrand Reinhold, 1993