Neural Predictive Control for Wide Range of Process Systems

Ali Jazayeri *. Alireza Fatehi **, Houman Sadjadian*, Ali Khaki-Sedigh******

Advanced Process Automation & Control (APAC) Research Group., Faculty of Electrical Eng.
K. N. Toosi University of Technology, Tehran, Iran
** (e-mail: sajazayerim@yahoo.com)*
*** (Tel: +98 21 88469084; e-mail: fatehi@kntu.ac.ir).*
**** (e-mail: sadjadian@dena.kntu.ac.ir).*
***** (e-mail: sedigh@kntu.ac.ir).*

Abstract: In this paper a Neural Predictive Controller (NPC) designed to control a broad class of process systems. Neural network identification yields nonlinear global model of the unknown system. Levenberg-Marquardt (L-M) optimization method is used to find optimal control signal to minimize future errors of the objective function of predictive controller. Inequality constraints of actuators are added to the objective function through a penalty term which increases drastically as it approaches the limitations. To use the controller for wide range of process systems, an initial phase runs before the main controller to determine parameters. This phase moves the system output to operating point and applies PID controller with APRBS reference signal. The gathered data are used to estimate parameters such as pure delay, prediction horizon, control coefficient and identification order. To validate the approaches, the controller has implemented in level, pressure and flow pilot plants and compared with conventional controller which shows faster and smoother tracking results.

1. INTRODUCTION

Conventional linear controllers are vastly used in process control industries. In addition to simplicity, these systems can be applied to wide range of systems without much interference of expert personnel. For instance, control handbooks show straightforward procedures to determine PID coefficients (O'Dwyer, 2006). Auto-tuning methods have also developed to tune PID afterward (Åström, 2005). In these algorithms, one or few tests are employed into the system to design the controller. However to design an advanced controller it is necessary to do more tests. This happens because of the large number of design parameters in these types of controllers. To overcome this draw backs, researchers recently try to introduce some simple but general purpose design method for advanced controllers (Khaki-Sedigh 2008, Shridhal and Cooper 1998).

Controllers which can be used for wide range of systems are called Universal Controller Systems (UCS). In this paper, we propose a thorough approach to apply an intelligent UCS based on nonlinear model predictive control (NMPC) to wide range of process systems. The ultimate aim of the system is not to work without *a priori* data; but to make the required data as general and as few as possible. These *a priori* data, which will be introduced later, are parameters like control signal saturation level and admissible range of output of the plant. Then the system extracts controller parameters for a neural predictive controller (NPC) from some automatic, yet save, tests on the plant. The whole process is very similar to design PID controller using simple techniques like Ziegler-Nicols technique. Before applying the main PID controller, a closed-loop test is implemented on the process. Oscillation frequency and amplitude are captured from these tests. The important part is to develop simple relations to determine PID coefficients from these two parameters.

The key idea in this paper is to conduct a comprehensive test and to offer a table or graph for each of controller parameters. Important parameters are determined from the tests. Design parameters are divided to three sets. The first set of parameters is less effective set and can be assumed constant. The second set is effective parameters and some approaches are presented in this paper to obtain them. The third set is also effective but is exempted in this paper because of involvedness.

The paper is organized as the following. In the next section, start-up and PID tuning phases are explained. These phases move the system to the operating point, make a primary PID controller based on previous projects. The third section describes designed identification phase. The neural predictive controller is discussed in section 4. Section 5 is about the parameters of neural network identification and offer automatic approaches to determine its important parameters. Parameters of the main controller are determined in Section 6. Implementation results on three pilot plants are presented in Section 7.

2. START-UP AND PID TUNING PHASES

Table 1 summarizes the whole phases of the design procedure. The first 3 phases move process to the desired operating point and design a primary PID controller for it in closed-loop. This controller is used later in identification phase. Details of the procedures of these phases are out of the scope of this paper and will be explained in summery in the following; however, they are well explained in (Khaki-Sedigh 2008). The procedure needs the parameters mentioned in Table 2 as pre-requisite parameters. These parameters are general information which can simply be offered by operators of the process.

Table 1 – Start-up and identification phases

No.	Phase Name	Phase Description	Estimated Parameters
1	Start-up	Small steps are applied to control signal until system output reaches the operating point.	Settling time, time constant, level of noise
2	Relay feedback	To design the coefficients of the PID controller, Relay feedback put the system in limit cycle.	Ultimate frequency and gain (required for auto-tuning PID algorithm; Zigler-Nichols)
3	PID Controller	A PID controller runs in different operating point.	closed-loop raw data and parameters of Table 3
4	Nonlinear System Identification	An APRBS reference signal is implemented in operating range; between Min_Y and Max_Y.	MLP Neural Network model with tapped delay inputs.

Table 2 – Parameters which operator should determine

Variable Name	Description
Max_U and Min_U	Actuator saturation level
Max_Y and Min_Y	Admissible process output
Delta_U	Admissible step change in control signal
Max_Delay	The time that is more than process delay
Operating point	For linear system identification
Process sign	Sign of process gain (Positive/Negative)
Sample_Time	Extracted from a rough estimation of process lag

At the first phase, the process is moved to the desired operating point, since it might be in a different operating point. This is called start-up. Start-up phase applies small steps to the system input until the system output reaches the desired value. Step size should be less than Delta_U. Since the plant is assumed to be stable, this procedure can be implemented. For unstable system, if it is mandatory to know a primary stabilizing controller and start-up is done under closed-loop.

In the operating point, a relay feedback controller applies. Typical processes, in which Nyquist diagram crosses horizontal axis have a limit cycle under relay feedback. Output level of the relay is set less than Delta_U. Input-Output data is used to design PID controller based on Zigler-Nicols closed-loop method (Åström, 1994). At this point the PID controller runs and moves system to different operating points during the nonlinear identification phase.

3. NONLINEAR IDENTIFICATION

In the next step, a nonlinear model is identified for the plant. To produce the required input-output data, reference signal of the control system under PID controller is set to be Amplitude modulated Pseudo-Random Binary Sequence (APRBS) signal. APRBS signal should be persistently excited (PE) to motivate all system modes. The signal remains constant in a time interval that is integer multiply of Minimum Hold Time (MHT). MHT is assumed time-constant of the system. We use the time constant in the desired operating point. However, shortest time-constant in different operating points can be used for nonlinear systems.

Common approach to make an APRBS signal is to multiply Pseudo-Random Binary Sequence (PRBS) signal by random levels (Nelles , 2000). In this case two sequential random numbers are rarely the same. However, for steady-state identification, excitation signal has to be constant for at least the settling time of the system. As a result, this common approach for producing APRBS signal does not identify low frequency modes of the process and results in incorrect estimation of system final gain. If PRBS interval time is increased, high frequencies will be missing in the identification. In another approach, we propose to have both small and large intervals in the excitation signal. In the randomly shifted PRBS some intervals are repeated randomly. At each interval, level of the previous interval is repeated with a low probably, say 0.3. Otherwise, new random level is assigned:

$$R(t+1) = \begin{cases} R(t) & ; \quad \textit{with probability of } 0.3 \\ Rand & ; \quad \textit{with probability of } 0.7 \end{cases} \qquad (1)$$

where, $R(t)$ is the APRBS sequence at time interval *t*. *Rand* is the new random number in the full span of the operating points, *i.e*. between Min_Y and Max_Y with uniform distribution. Random numbers can be limited to 20% to 80% of the full span of the operating points because this signal is reference signal of the control system but the controller might have overshoot near the margins. Furthermore, APRBS length must be 24 to 30 times of MHT (Jazayeri, 2007). The controller system with this APRBS reference signal offers data for input-output closed loop system identification. In general, due to safety issue, many systems are not allowed to have large input values for open loop identifications. Hence, they are excited in closed loop using the primary PID controller. The shape of the reference signal and its frequent changes are prerequisites of unbiased identification. Other prerequisites such as noise content, exogenous inputs and controller complexity could not fully comply in this system.

4. NEURAL PREDICTIVE CONTROL

Neural predictive controller is based on one-step-ahead predictor neural network model of the process. The neural model is an MLP with tapped delayed inputs. This structure brings global identification features further to its simplicity and availability due to vast applications. There are two layers in the neural network. Activation functions of the layers are sigmoid and linear, respectively. To train the network, current and previous inputs to the process together with its previous output are used. Since the training uses previous output of the actual system, it is call Teacher Forcing training (Haykin, 1998) which produces a predictor model.

Levenberg-Marquardt (L-M) algorithm is used to train the network. Neural network trains at two times. The initial training takes place at the end of PID control phase using gathered data of identification phase. The second training is during the main controller at certain periods using sliding window of previous data (Dias, 2005). In most process control applications, reference signal of the main controller does not necessarily vary to meet P.E. condition. As a result, training data are chosen by a high-pass filter to delete steady-state data from training set. In other words, in the practical process systems, it is frequently happens that all the signals remain constant for a long time. If all gathered data are used for identification algorithm, developed model reflects merely the steady-state. Neural model predicts only one step ahead prediction but the predictive controller requires predicting multi-step ahead outputs. For this purpose, the neural network is used sequentially to yield N_2-step ahead prediction. This approach is preferred to NN with N_2 outputs to avoid high number of training parameters and increasing complexity and computational burden.

The objective function of predictive controller is

$$J(t) = E^T(t)E(t) + \tilde{U}^T(t)Q\tilde{U}(t) + \sum_{k=1}^{N_u} f\left(u(t+k)\right) \qquad (2)$$

where, $\tilde{U}(t)_{N_u \times 1}$ is vector of future control signals, N_u is control horizon, Q is matrix of control coefficients and assumed diagonal, *i.e.* $Q = \lambda \cdot I_{N_u \times N_u}$. In Eq.2 $E(t)_{(N_2 - N_1)\times 1}$ is vector of future errors and is equal to $E(t) = \hat{Y}(t) - R(t)$ where $\hat{Y}(t)_{(N_2 - N_1)\times 1}$ is vector of system output predictions, $R(t)_{(N_2 - N_1)\times 1}$ is vector of future references which assumed given. N_1 and N_2 are minimum and maximum prediction horizon, respectively.

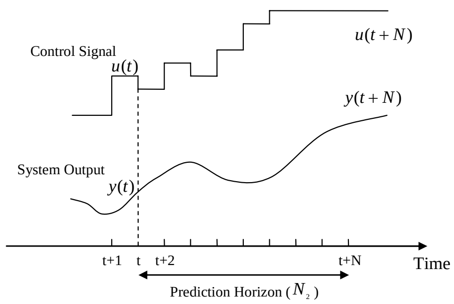

Figure 1 ─ Predictive Control Scheme

To consider the constraints, the constraint penalty term $f(.)$ is used in Eq. 2. This function exponentially increases when its argument approaches its constraint. It is defined as:

$$f(u) = \exp\left(\frac{2S.(u-\bar{u})sign(u-\bar{u})}{u_{\max} - u_{\min}}\right) / \exp(S) \qquad (3)$$

where, $u_{\min}$ and $u_{\max}$ are actuator limitations and $\bar{u}$ is mean of them and S is sharpness of the penalty term.

The goal of predictive control is to minimize the objective function of Eq.2 by manipulating future control signals. In Eq.2 future error can be rewritten according to future control signals and the prediction of the output of the plant based on the neural network model. The neural network should be used N_2 times. The objective function entirely is written based on prediction of future errors. To minimize this objective function different approaches have been proposed such as Newton-Raphson (Soloway and Haley, 1998), Gradient Descent (Gil, 2000) and Linear Programming (Leskens, 2005). In this paper, Levenberg-Marquardt optimization method is used, since it is much faster and more reliable optimization technique (Nogaard, 2003). The first and the second derivatives (gradients and Hessians) are derived in (Nogaard, 2003) and (Jazayeri, 2007). In each sampling time, this optimization is repeated to find a sequence of appropriate control signals; however, only the first element is applied to the system (receding horizon control). Practical results of this controller for a sample system are shown in (Jazayeri, 2008).

5. IDENTIFICATION PARAMETERS

Before running system identification phase, parameters of Table 3 are extracted from gathered data.

Table 3. Identification parameters

#	Name	Description	Solution
1	Pure delay	Transportation delay	Lipschitz index and Makaremi approach
2	Model Order	Number of dynamics (tapped delay input and output of the NN)	Linear system identification
3	Hidden Neurons	Number of Sigmoid neurons in the middle layer of NN	Constant for process systems
4	Training termination error	When training error reaches this value, training finishes.	Constant for normalized data
5	Training epoch	Maximum number of training epoch	Constant because of time limitations

From identification parameters of Table 3, last three parameters are not crucial in the system and considered in Table 4. On the other hand, the pure delay and the system order are particularly important and are determined from the following two approaches.

To estimate delay in a general nonlinear system (Makaremi , 2008) proposed Lipschitz index of Eq. 4:

$$LB^d = \left\| \left[LB_{ij}^d \right]_{N \times N} \right\|_\infty \qquad (4)$$

where,

$$LB_{ij}{}^d = \frac{|y(i) - y(j)|}{\sqrt{(u_d(i) - u_d(j))^2 + (u_{d+1}(i) - u_{d+1}(j))^2 + \cdots + (u_{D_0}(i) - u_{D_0}(j))^2}}$$

where, d is delay pointer and D_0 is maximum tested delay which should be equal or more than Max_Delay of Table 2. Lipschitz index tries to estimate system gradient based on input-output data (He and Asada, 1997),(Nelles, 2000). When d increases, first inputs are gradually deleted from this index. If removing a dynamic has no effect on the index, it means that it is before the delay. At the dynamic of system delay, this index increases considerably. Thus, the second derivative of this index has a peak on the pure delay of the system. It is shown that if the reference signal of the initial controller is set to be at least two positive steps, this approach can estimate the pure delay precisely.

In spite of the pure delay which does not change in closed-loop, order of a nonlinear system varies in closed-loop. Speed of the designed controller profoundly influences the Lipschitz index and consequently Makaremi approach is unreliable to estimate system order. In this paper, another assumption about the general nonlinear system simplifies the issue. For many process systems, changing from one operating point to another one dose not change number of state variables. In other words, order of the nonlinear system can be assumed constant through operating interval. In case number of state variables varies in different operating points, one chooses the largest number of state variables instead. Then, the system order which is obtained from linear system identification can be generalized to all other points. Thus, n_a (number of tapped delayed inputs) and n_b (number of tapped delayed outputs) are set to the poles and the zeros of the identified transfer function.

Other parameters of Table 3 are not as important as the two mentioned ones. Different simulations showed that these parameters have no significant effect on closed-loop performance of the neural predictive controller. These parameters are assumed constant. To evaluate importance of each parameter, a percentile comparing index is defined by Eq. 5:

$$I_C(\%) = \frac{\sqrt{\mathrm{var}(IAE(p))}}{mean(IAE(p))} \times 100 \quad (5)$$

where, p is the parameter and IAE is the Integral Absolute Error of closed-loop tracking error of the neural predictive controller. Table 4 summarizes effects of less important parameters on the overall performance using index of Eq. 5 and the final selections.

Table 4. Effect of less important parameters on performance index

#	Parameter	Test Interval	$I_C(\%)$	Selection
1	Hidden Neurons	2- 6	20.3	4
2	Training termination error	10^{-8} - 10^{-5}	17.7	10^{-7}
3	Training epoch	200-1000	4.40	500
4	Initial step size in Optimization	0.05-0.4	5.61	0.1
5	Optimization termination error	10^{-6}-10^{-3}	6.23	10^{-4}
6	Optimization iteration	3-10	1.66	5

6. CONTROL PARAMETERS

Another set of parameters that is determined based on gathered data, is parameters of the neural predictive control. After finishing identification, these parameters are found automatically. Control parameters are shown in Table 5.

Three last parameters of Table 5 are not important on overall performance. Simulation results of Table 4 demonstrate that variation of these parameters are not important in overall performance of the controller. Maximum prediction horizon (N_2) will be important if sample time is not set properly. Maximum prediction horizon multiplied by sample time should be at least equal to the settling time of open loop system (Comacho, 2007). Control horizon (N_u) is typically set to two-third to three-forth of maximum prediction horizon. If sample time is set a small number, optimization will be time consuming. Prediction and control horizons are set to constant numbers of 7 and 5 respectively. The operator approximately may set the sample time to one-seventh of open loop settling time.

Table 5. Neural Predictive Control parameters

#	Name	Solution
1	Maximum prediction Horizon (N_2)	Constant. Set by sample time
2	Control Horizon (N_u)	Constant. Set by sample time
3	Control coefficient (λ)	Determined based on system gain, N_u and N_2
4	Initial step size in L-M	Constant
5	Optimization termination error	Constant
6	Optimization iteration	Constant

Among all these parameters, control coefficient (λ) is one of the most influential parameters. The small value of λ means less importance of control signal variation and consequently controller have better tracking but with high frequency changes in the actuator. Conversely, when λ increases control signal is bounded tightly. Accordingly, control signal is smoother but tracking influences. Choosing λ is actually a trade-off between control smoothness and fast tracking. As it is compared in Table 6, transient performance of NPC controller greatly impacted by control coefficient.

Table 6. Effect of control coefficient on NPC control performance

Control coefficient (λ)	$\lambda = 1$	$\lambda = 2$	$\lambda = 4$
Overshoot	1.93	13.0	23.3
Steady-state Error	79.0×10^{-3}	79.2×10^{-3}	78.6×10^{-3}
Settling Time	58.8	105	183
Rise Time	48.7	53.7	55.7

First step to determine control coefficient is to notice the units of the control signal and the system output in the objective function of Eq. 2. If the control signal is in 10^{-3} scale but the system output in 1 scale, control coefficient first should be multiplied by $(10^{-3})^2$ to take account the units and to normalize two terms of the objective. In general, λ can be divided into two terms as Eq. 6

$$\lambda = \lambda^{(1)} \lambda^{(2)} \quad (6)$$

where, $\lambda^{(1)}$ is a free parameter and $\lambda^{(2)}$ is the square of system gain. Using this partitioning, performance changes independently from the system gain when $\lambda^{(1)}$ changes. To determine $\lambda^{(2)}$ it is necessary to use start-up data of Table 2 to estimate system gain. To determine $\lambda^{(1)}$ (Shridhal and Cooper, 1998) proposed an algorithm for Dynamic Matrix Control (DMC). In this approach, the system is approximated by first order plus delay transfer function and the coefficient is proposed to be:

$$\lambda^{(1)^2} = \frac{N_2}{500}\gamma^2\left\{N_u - \frac{\theta}{T} - \frac{3}{2}\frac{\tau}{T} + \frac{3}{2} - \frac{N_2}{2}\right\} \quad (7)$$

where, γ is the final parameter which determines overall performance, θ is the pure delay of the system, τ is the time-constant of the system and T is the sample time.

7. PRACTICAL RESULTS

The proposed methods are simulated for different systems to validate viability of the system. In (Jazayeri et al., 2008) simulation results of the proposed method compared to linear Model Predictive and non-adaptive systems and the result justifies use of proposed method in simulated plants. In this section, only practical results on pilot plants are presented.

Figure 2 – GUNT RT512 level plant in Process Control Lab.

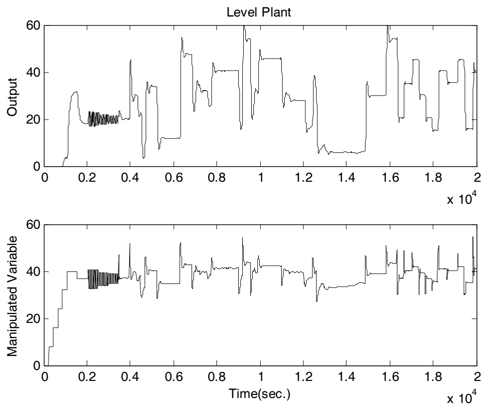

Figure 3 – Whole process of implementation of the system on level plant

The automatic tuning strategy has successfully tested in level, pressure and flow pilot plants and depicted in Figure 3 to Figure 6. Figure 3 shows the whole control system design stages from start-up to the neural predictive control which is implemented on the level plant. The level plant is a liquid tank with controllable inlet valve and measurable liquid level. External disturbance may apply by changing outlet hand valve. The ultimate objective of the controller is to track desired liquid level by manipulating inlet valve independent from outlet valve position. At first, the system is in its initial state. Few small steps bring the system to the operating point of $y = 20$ at $t = 0.2 \times 10^4$. Then, feedback relay runs. After that, PID is designed based on closed-loop Zigler-Nichols method. Reference signal of APRBS is built at this time. Next, APRBS signal applies and after gathering enough data, the neural network is trained using parameters of section 5. Finally, parameters of the neural predictive control are obtained and the neural predictive controller is applied.

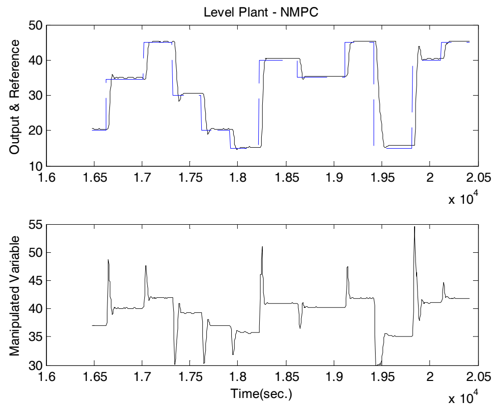

Figure 4 – NPC implementation on level plant

Figure 3 does not clearly show performance of the main controller because of the long time of the implementation. The main performance is shown separately in **Error! Reference source not found.**. For pressure and flow plants, only the neural predictive controller is depicted in Figure 5 and Figure 6. The pressure plant aims to stabilize pressure of a vessel using inlet valve of high pressure line and might be disturbed by outlet valve openness and pressure of inlet line. In flow plant, water is circulating through pipes from and to a reservoir tank and controller manipulates valve position to achieve required flow rate of water. As it illustrated in **Error! Reference source not found.** to Figure 6, tracking is done very well for all these systems and transient response, overshoot, rise time, settling time and smoothness in control signal is completely acceptable. The overshoot of these controllers are very small and negligible in most cases. In addition, rise time and settling time are short. However, if one magnifies these figures, there are very small steady-state errors. This final error is caused by minute bias in neural network one step ahead prediction and accumulation of this error in long term horizon. The steady-state error is thoroughly studied and dealt with in (Jazayeri et al., 2008) for level plant. In spite of this, overall performance of the controller is improved comparing with conventional PID controller. This comparison is made in Table 7.

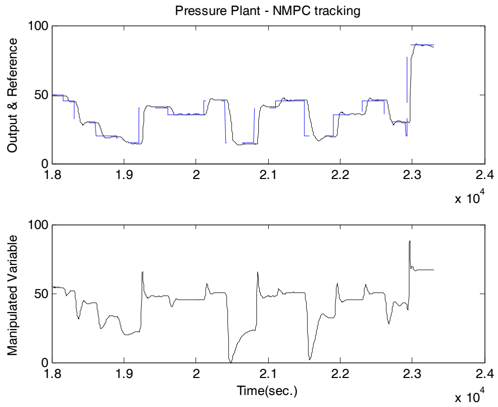

Figure 5 – NPC implementation on pressure plant

Table 7. Comparison to conventional controller for Pressure Plant

Controller	Overshoot	Steady-state error	Steady-state variance of output
Zigler-Nichols PID	47%	0	0.746
Neural Predictive	2.1%	0.11	0.038

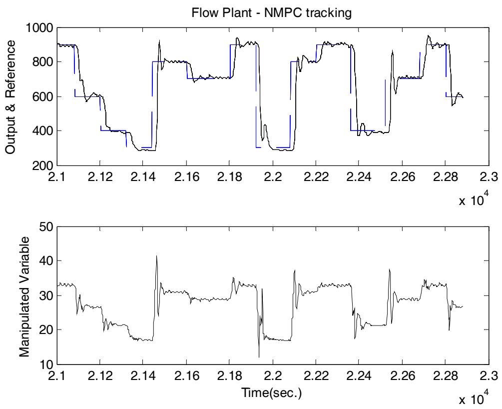

Figure 6 – NPC implementation on flow plant

8. CONCLUSIONS

This paper presents a neural network identification and Levenberg-Marquardt predictive controller for wide class of process systems. Identification and control parameters are estimated by Lipschitz index, linear system identification, system gain and other heuristic approaches to make the controller as automatic as possible. Implementation results on three industrial pilot plants shows better performance compared to conventional auto-tuning strategies.

REFERENCES

K. J. Åström and B. Wittenmark (1994) “Adaptive Control” Prentice Hall

K. J. Åström and T. Hägglund (2005) "Advanced PID Control" ISA

E. F. Comacho and C. Bordons (2007) “Model Predictive Control” Springer

S. Haykin (1998) “Neural Networks: A Comprehensive Foundation” Prentice Hall

F. M. Dias; A. Antunes; J. Vieira and A. M. Mota (2005) “On-line Training of Neural Networks: A Sliding Window Approach for the Levenberg-Marquardt Algorithm” Springer, Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, Springer, pp. 577-585

P. Gil; J. Henriques; A. Dourado and H. Duarte-Ramos (2000) “Extended Neural Model Predictive Control of Non-linear Systems” Proceeding of International Association of Science and Technology for Development, pp. 94-100

X. He and H. Asada (1997) A new method for identifying orders of input-output models for nonliner dynamic systems, American Control Conference, pp. 3294-3299

A. Jazayeri; A. Fatehi; H. Sadjadian; and A. Khaki-Sedigh, (2008) Disturbance Rejection in Neural Network Model Predictive Control, Proceeding of the 17th IFAC World Congress, Korea

A. Jazayeri (2007) Intelligent Adaptive Model-based Control, Master of Science thesis, K. N. Toosi University of Technology

A. Khaki-Sedigh (2008) Universal Control System, K. N. Toosi University of Technology and Ministry of Industries and Mines, Internal Report

M. Leskens; J. L. van F. Kessel; van Hof, P. M. J. van den and O. H. Bosgra (2005) “Nonlinear Model Predictive Control with Moving Horizon State and Disturbance Estimation – with Application to MSW Combustion” Proceedings of 16th IFAC World Congress, pp. 1-6

I. Makaremi, A. Fatehi and B. Nadjar Araabi (2008) “Lipschitz Numbers: A Medium for Delay Estimation” Proceedings of 17th IFAC World Congress

O. Nelles (2000) “Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models” Springer-Verlag

M. Nogaard; O. Ravn; N. K. Poulsen; and L. K. Hansen (2003) “Neural Networks for Modelling and Control of Dynamic Systems” Springer-Verlag

A. O'Dwyer (2006) "Handbook of Pi and Pid Controller Tuning Rules" Imperial College Press

R. Shridhal and D. J. Cooper (1998) “A Tuning Strategy for. Unconstrained Multivariable Model Predictive Control,” Industrial & Engineering Chemistry Research, vol. 37, pp. 4003-4016

D. Soloway and P. J. Haley (1996) “Neural Generalized Predictive Control; A Newton-Raphson Implementation” Proceeding of IEEE International Symposium on Intelligent Control, pp.277–282