Nonlinear Model Predictive Control Containing Neural Model and Controller

Bambang L. Widjiantoro[1], The Houw Liong[2], Yul Y. Nazaruddin[3] and Bambang SPA[3]
[1]Jurusan Teknik Fisika FTI – ITS Kampus ITS Sukolilo Surabaya
[1]Departemen Fisika ITB – Jl. Ganesa 10 Bandung
[2]Departemen Teknik Fisika ITB – Jl. Ganesa 10 Bandung
email : blelono@tf.itb.ac.id,blelono@ep.its.ac.id

Abstrak

Algoritma sistem kontrol prediktif nonlinier terdiri dari tiga elemen utama yakni model proses nonlinier, fungsi obyektif dan optimasi. Model proses nonlinier berfungsi untuk melakukan prediksi terhadap output proses waktu mendatang sepanjang horison prediksi. Sinyal kontrol optimal dibangkitkan berdasarkan optimasi terhadap suatu fungsi obyektif. Makalah ini membahas pengembangan sistem kontrol prediktif nonlinier berbasis jaringan syaraf tiruan. Jaringan syaraf tiruan digunakan untuk membangun suatu model proses nonlinier, serta untuk membangun kontroler nonlinier sebagai pengganti prosedur optimasi sebagaimana dilakukan oleh kontroler konvensional. Dengan demikian sinyal kontrol optimal merupakan output dari kontroler jaringan syaraf tiruan, dan tidak ada lagi masalah optimasi yang harus diselesaikan pada tiap cacah. Struktur jaringan syaraf tiruan yang digunakan adalah multilayer perceptron. Hasil simulasi menunjukkan bahwa sistem kontrol yang dikembangkan mampu memberikan performansi yang baik serta memerlukan waktu komputasi yang lebih kecil dibanding sistem kontrol prediktif nonlinier menggunakan optimasi.

Kata kunci: *multilayer perceptron, sistem kontrol prediktif nonlinier, optimasi.*

Abstract

Nonlinear model predictive control (NMPC) strategies consist of three main elements: nonlinear process model, objective function and optimization. The nonlinear process model is used to predict the future behavior of the process output along prediction horizon. The optimization procedure is used to minimize the objective function for generating the control signal. The paper provides the development of neural networks based nonlinear model predictive control algorithm. The neural networks will be used to develop the nonlinear process model and nonlinear controller to replace the optimization procedure. Thus, the current control signal is given by the output of neural networks controller, instead as the solution of the optimization problems and no nonlinear optimization must be solved at every sampling time. The structure of neural networks for nonlinear process model as well as nonlinear controller is multilayer perceptron (MLP). The structure consists of three layers including input, hidden and output layer. Simulation results show the NMPC with neural networks model and controller can yield good control performances. In addition, the NMPC with neural networks model and controller required less computational time compare to the NMPC with optimization procedure.

Keywords: *Multilayer perceptron, Nonlinear model predictive control dan optimization.*

1. Introduction

As growth of the requirements to get the quality control system better, a model based control that will improve the control system performances had been developed. In the model based control algorithm, a process model is used directly in computational routines to produce the control signals. One of the model based control algorithm is the model predictive control (MPC).

Model predictive control consist of three main elements: the process model, the objective function and the optimization procedure to produce the control signals (Camacho, 2000). In MPC strategies, the process model is used to predict the future behavior of the process. So, the process model should have the similar characteristics with the handled process. Since most real world dynamics systems are highly nonlinear, the predictive control based on linear models may result in poor controlled performance caused by the limitations in linear model such as linearization and small operation range. It is therefore a nonlinear model is needed to anticipate the nonlinearities and complexities of the process.

The neural networks offer many advantages to build up the nonlinear model of the process. Their ability in nonlinear mapping between input – output is the potential benefits in the development of nonlinear model. The nonlinear model can be built with no complicated mathematical equations and need no detail information.

When the predictive control strategy is based on nonlinear model, independent of the nature of the model, the prediction equation cannot be solved explicitly, as in the linear case, and an iterative solution of the performance function evaluating the future behaviour of the system is required (Henson, 1998). At every sampling time, the current manipulated variable is calculated using an optimization procedure, which determines the optimal profile control actions that minimize the objective function. Hence, the use of nonlinear model allows the development of predictive control strategy for nonlinear dynamic process, but it also implies that at every sampling time a nonlinear optimization problem should be solved. This is a potential drawback of those control strategy because the solution of optimization problem are usually computational laborious. Most nonlinear optimization algorithms use some form of search technique to scan the feasible space of the objective function until the extreme point is found. This search is generally guided by calculation of the objective function and/or its derivatives and it implies a large computational effort because several iterations should be carried out to reach the extreme point. The computational time requirements may be an obstacle for nonlinear model predictive control in real time application. As consequences, it worthwhile developing nonlinear model predictive control strategy that require less computational effort.

In this paper, the neural networks will be used as controller in nonlinear model predictive control scheme. The optimization routine is replace by neural networks controller. Thus, the current control signal is given by the output of the neural networks controller, instead of the solution of the optimization problem and no nonlinear optimization problem should be solved at every sampling time. For this reason, the control strategy requires less computational effort.

The paper is organized as follows. Section two describes the concept of model predictive control. Section three discusses about the achievement of nonlinear process model using neural networks. Section four describes about the development of neural networks controller in model predictive control algorithm. Section five will illustrate the simulation of the control strategy to nonlinear process.

2. Model Predictive Control (MPC)

Model predictive control (MPC) refers to a class of control algorithms in which a dynamics process model was used to predict and optimize process performance (Camacho, 2000). In the beginning, the MPC algorithm was based on linear dynamic models and therefore was referenced by term linear model predictive control (LMPC). Although often unjustified, the assumption of process linearity greatly simplifies model development and controller. Many processes are sufficiently nonlinear to preclude the successful application of linear model predictive control algorithm. This has led to the development of nonlinear model predictive control (NMPC) in which a more accurate nonlinear model was used for process prediction and optimization. The nonlinear model predictive control algorithm offers the potential for improved process control performance (Henson, 1998).

The concept of model predictive control (MPC) is depicted in figure 1 (Garcia et.al, 1989).

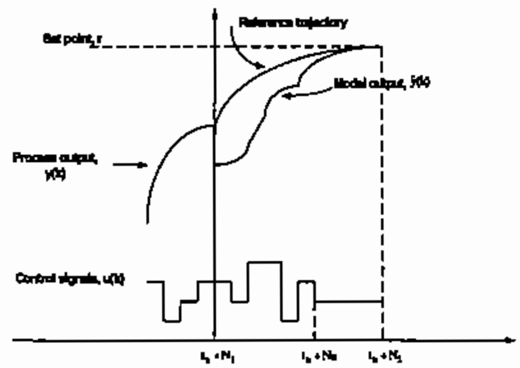

Figure 1.
The concept of MPC

The idea behind model predictive control (MPC) is at each iteration to minimize the objective function of the following type:

$$1.\quad J=\lambda\sum_{i=N_1}^{N_2}[r(k+i)-\hat{y}(k+i)]^2+\rho\sum_{i=1}^{N_u}[\Delta u(k+i-1)]^2 \qquad (1)$$

N_1 and N_2 denote minimum and maximum prediction horizon, espectively.
Nu denotes control horizon.
λ and ρ denote weighting factor for process output and control signal, respectively.
$\hat{y}$ is the prediction of future process output from the model.

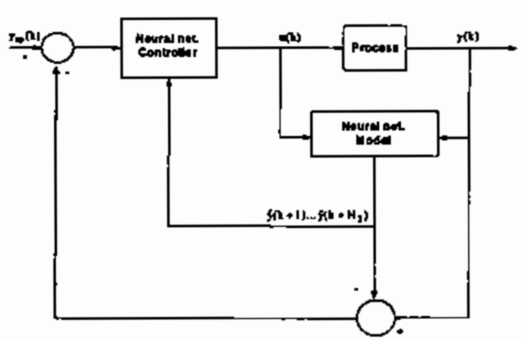

Figure 2.
The block diagram of NMPC with neural model and controller

The optimization problem should be solved at each sample, resulting in a sequence of future control signals. From this sequence, only the first element, u(k), is applied to the process. A prime characteristic that distinguish model predictive control (MPC) from other controller, is the idea of a receding horizon; at each sample the control signals is determined to achieve a desired behavior in the following N_2 time steps.

In general, there are no significant differences between LMPC and NMPC procedures, except in the use of the nonlinear model and optimization in NMPC strategies. The block diagram of the proposed control system is depicted in figure 2.

3. Nonlinear Process Model Using Neural Networks

Some characteristics and properties of neural networks that will be useful in system identification of the process are as follows (Sbarbaro, 1992):

- Nonlinear system. Neural networks have greatest promise in the realm of nonlinear control problem. This stems from their ability to approximate arbitrary nonlinear mappings.
- Learning and adaptation. Neural networks are trained using past data records from the system under study. Suitably trained networks have the ability to generalize when presented with inputs not appearing in the training data. Neural networks can also be adapted on line.
- Multivariable systems. Neural networks naturally process many inputs and have many outputs. It is readily applicable to multivariable process.

The neurons by themselves are not very powerful in terms of computation or representation but their interconnection allows us to encode relations between the variables giving different powerful processing capabilities. The connection of several layers gives the possibility of more complex nonlinear mapping between the inputs and the outputs. This capability can be used to implement classifiers or to represent complex nonlinear relations among the variables.

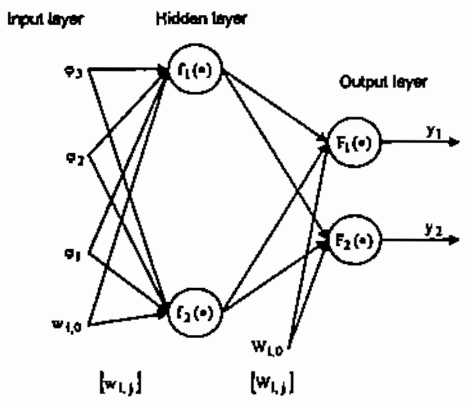

Figure 3.
The multilayer perceptron

The most common of neural networks structure is multilayer perceptron (MLP). The basic MLP networks is constructed by ordering the neurons in layers, letting each neuron in a layer take as input only the outputs of units in the previous layer or external inputs. If the networks have two such layers of neurons, it refers to as a two layer networks, if it has three layers it is called a three layer networks and so on. Figure 3 illustrates the example of three layer networks including input layer, hidden layer and output layer. The mathematical formula expressing what is going on in the MLP networks takes the form:

$$y_i = F_i\left[\sum_{j=1}^{n_h} W_{i,j} \cdot f_j\left(\sum_{l=1}^{n_\varphi} w_{j,l}\,\varphi_l + w_{j,0}\right) + W_{i,0}\right] \quad (2)$$

Cybenko (1989) has shown that all continuous function can be approximated to any desired accuracy with neural networks of one hidden layer of hyperbolic tangent hidden neuron and a layer of linear output neuron.

In order to determine the weight values, it should be available a set of examples of how the outputs, $\hat{y}_i$, should relate to the inputs, φ_i. The task of determining the weights from these examples is called training or learning. The aim of training procedure is an adjustment of the weights to minimize the error between the neural networks output and the process output (also called by target). A learning algorithm is associated with any change in the memory as represented by the weights; learning does not in this sense imply a change in the structure of the memory. Therefore, learning can be regarded as a parametric adaptation algorithm.

The learning algorithm is Levenberg-Marquardt method. This algorithm requires the information of gradient and Hessian matrices. The convergence will generally be faster than for the back-propagation algorithm. The detail derivation of Levenberg-Marquardt method can be seen in Norgaard et.al (2000).

The series-parallel model or NARX (Nonlinear Auto Regressive with eXogenous input) model can be as nonlinear model in model predictive control strategy. The model structure can be expressed as follows:

$$\hat{y}(k) = \hat{f}(x(k))$$
$$x(k) = \left[u(k-1) \ldots u(k-n_u)\ y(k-1) \ldots y(k-n_y)\right]^T \quad \ldots\ldots(3)$$

where $\hat{f}$ is nonlinear function, while n_u and n_y are history length for process inputs and outputs, respectively. The NARX model used the previous process output as regressor in x(k). The NARX can predict either for one step-ahead prediction or multi step-ahead prediction and has good stability.

4. Neural Networks Controller

In the nonlinear model predictive control (NMPC), the controller has a task to replace the optimization problem for generating the control signal. Thus, there is no nonlinear optimization when the neural networks used in NMPC strategies.

The training procedure of neural networks controller can be divided into two methods: generalized training and specialized training (Psaltis, 1988). The weakness associated with the generalized training method: the criterion expresses the objective to minimize the discrepancy between the neural output and sequence control signal. This is not really a relevant objective. In practice, it is not possible to achieve zero generalization error and consequently the trained networks will have certain inaccuracies. Although these generally reasonably small, in terms of the networks output being close to the control signal, there may be large deviations between reference signal and the process output when the networks is applied as controller for the process. It is therefore, in the NMPC algorithm the neural networks controller must be trained in specialized method based on the objective function.

The following section will derive the algorithm of the neural networks controller in NMPC. The algorithm was derived and modified from Galvan et.al (1997) and Norgaard et.al (2000). Suppose that the objective function of predictive control strategy is formulated as follows:

$$e(k+1)=\frac{1}{2}\sum_{i=N_1}^{N_2}\left(y_{sp}(k+i)-\hat{y}(k+i)\right)^2 \quad (4)$$

where y_{sp} (k + i) and $\hat{y}$(k + i) are the sequences of set point and the output prediction from model, respectively, N_2 is prediction horizon. The goal of the predictive control strategy is to produce the control signals that minimize the objective function in equation (4).

The derivation of control algorithm using neural networks controller can be viewed as the analog of equation (3). Hence, the control signal u(k) minimizing the objective function in equation (3) can be written as follows :

$$u(k)=g\big(\ y_{sp}(k+N_1),\cdots,y_{sp}(k+N_2),y(k),\ \hat{y}(k+N_1),\hat{y}(k+N_2),u(k-1),u(k-n_u)\ \big) \quad \ldots\ldots(5)$$

where g is a mapping function that determines the solution of optimization problem given by the objective function in equation (4) and $\hat{y}$ is the model output. Suppose that the set point remains constant along the prediction horizon (N_1, ..., N_2), equation (5) can be simplified as follows:

$$u(k)=g\big(\ y_{sp},y(k),\hat{y}(k+N_1),\hat{y}(k+N_2),\ u(k-1),u(k-n_u)\ \big) \quad \ldots\ldots(6)$$

If the function g was known, the expression given by equation (6) would provide at every time k the control signal that must be applied to the system to reach the desired control objective, when a predictive control strategy is employed. The problem of finding an expression for the function g can now be interpreted as functional approximation problem. Based on approximation capabilities of neural networks, one can decide to approximate the functional g by neural networks controller.

The learning process of the predictive neural controller consists of determining the weight parameter set (W_c) such that the control law, given by equation (6), provides the predictive controller performance. The training procedure is carry out on line by specialized training. In specialized training, the neural networks controller is trained to minimize the objective function so the system output will follow the reference signal closely. The learning algorithm used in this research was backpropagation algorithm.

Suppose W_c be a weight parameter of the neural networks controller and set point remains constant along the prediction horizon, the gradient of the error function given by equation (4) with respect to the parameter can be written as follows:

$$\frac{\partial e(k+1)}{\partial w_c}=-\sum_{i=N_1}^{N_2}\left(y_{sp}-\hat{y}(k+i)\right).\frac{\partial\hat{y}(k+i)}{\partial w_c} \quad \ldots\ldots(7)$$

By application of the chain rule, the gradient model output with respect to the parameter w_c along the prediction horizon can be calculated by:

$$\frac{\partial \hat{y}(k+i)}{\partial w_c}=\frac{\partial \hat{y}(k+i)}{\partial u(k+i)}\cdot\frac{\partial u(k+i)}{\partial w_c}, i={}_1,\ldots,N_2 \quad \ldots\ldots(8)$$

where $\frac{\partial \hat{y}(k+i)}{\partial u(k+i)}$, i = $N_1,\ldots, N_2$ are the model Jacobians along prediction horizon, and $\frac{\partial u(k+i)}{\partial w_c}$, i=$N_1,\ldots, N_2$ represents the gradient of the neural networks controller output with respect to the parameters w_c. Based on the above description, the neural predictive controller parameters can be updated at every sampling time using the following learning rule:

$$W_c^{new}=W_c^{old}-\alpha\cdot\frac{\partial u(k)}{\partial w_c}\cdot\sum_{i=1}^{H}\left(y_{sp}-\hat{y}(k+i)\right)\cdot\frac{\partial \hat{y}(k+i)}{\partial u(k)} \quad \ldots\ldots(9)$$

5. Simulation

In the following section, the NMPC with neural networks model and controller will be applied to the nonlinear system. The proposed control system is applied to the nonlinear process of a mass-spring-damp system. The process can be expressed as follows (Norgaard,2000):

$$\ddot{y}(t)+\dot{y}(t)+y(t)+y^3(t)=u(t) \quad (10)$$

The first step in designing of NMPC is to develop the process model by system identification. The training data was generated using amplitude pseudo random binary signal (APRBS) as the input signal o the process. Figure 4 illustrates the input-output data.

The system identification with neural networks algorithm was done in NARX (Nonlinear Auto Regressive with eXogenous input) or series parallel model structure. The history length for both the input signal and the output signal was 4. It means that the input spaces consist the present and three past values of the input-output signal. Thus, the input layer has 9 variables as the input of regressor. The hidden layer consists 5 neurons with hyperbolic tangent function, while the output layer has a single neuron with linear function. Structure of the neural networks model is depicted in figure 5. Some experiments were carried out to find the good structure. The best results were obtained using the above structure. The goodness of identification was measured using Root Mean Squared Error (RMSE):

$$RMSE=\sqrt{\frac{\sum_{i=1}^{N}(y_i-\hat{y}_i)^2}{N}} \quad (11)$$

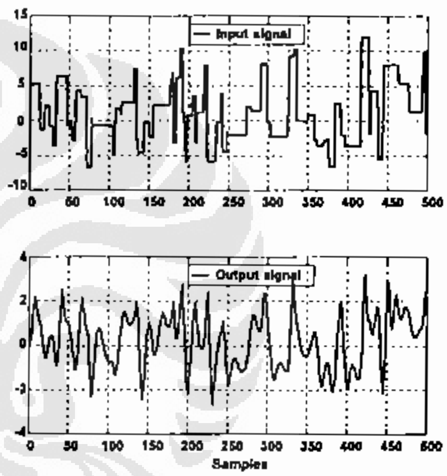

Figure 4.
Input-output data

The result of neural networks model in identification of the process is shown in figure 6. From this figure, it can be seen that the neural model yields a good performance. The model can identify and anticipate behaviour of the process satisfactorily. The neural model has RMSE = 0.003. With this result, the neural networks model can be used as the process model in NMPC algorithm. The model will predict the future behaviour of the process. Than, the output model is used in calculation of the objective function.

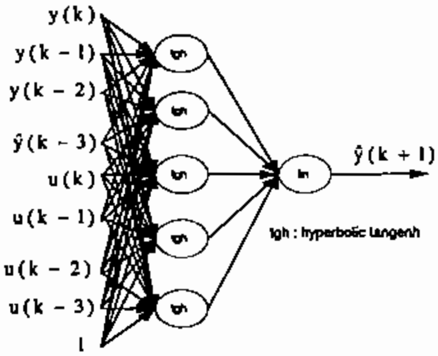

Figure 5.
Structure of the neural networks model

After developing the nonlinear process model, the next step is to determine the neural networks controller to generate the optimal control signal of the process. As mentioned earlier, the controller is trained in specialized manner to minimize the objective function expressed in equation 4. The structure of the neural networks controller was also multi layer perceptron (MLP), with input, hidden and output layer, and used the back-propagation learning algorithm to up date the weights of neural controller.

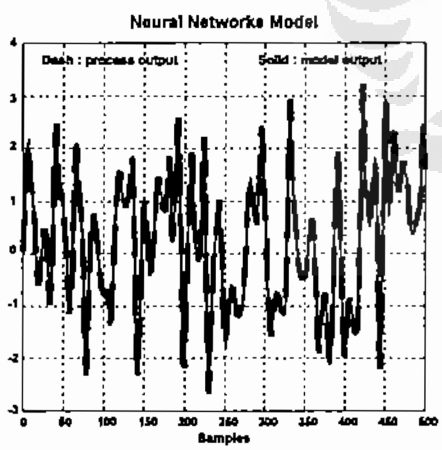

Figure 6.
Output of the neural networks model

The input pattern of neural networks controller takes some variables, including set point, current process output, model outputs and past control signals, to produce the current control signal. Structure of the neural networks controller is depicted in figure 7.

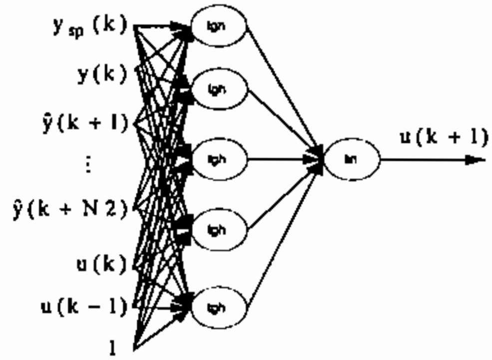

Figure 7.
Structure of the neural networks controller

Initialization of the weights either for neural networks model or neural networks controller was obtained in randomize manner. The neural controller should be trained several times (epoch) until the process output close to the desired set point and will give the minimum value of objective function. Then, the best value of weights was chosen as the neural networks controller to generate the control signal for the process. The experiment was carried out with various values of N_2 or prediction horizon to investigate how this parameter affects the control system performances. Simulation of NMPC with neural model and controller is illustrated in figure 8. The value of prediction horizon (N_2) = 7.

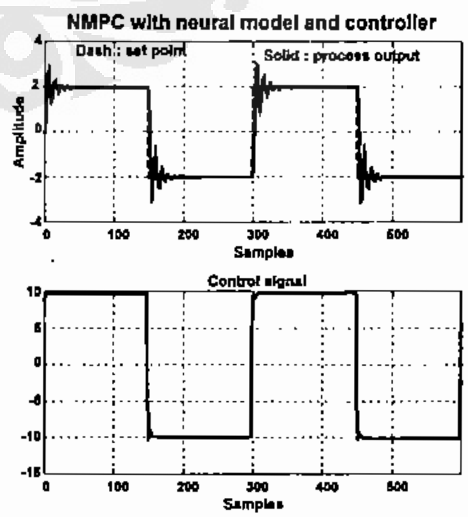

Figure 8.
Response of NMPC with neural model and controller (N_2 = 7)

In figure 8, the neural networks controller was trained in 7 epochs to reach the best performance and required 150 seconds for training phase. Then, the best values of the weight were used as controller weights parameter to result the control system performance. The good set point tracking was achieved in figure 8. The process output can track the set point without offset. The neural networks controller can also produce the smooth control signal for the process. This research was accomplished using sampling time = 0.03 seconds. Settling time of NMPC with neural model and controller was 0.75 seconds and overshoot was about 40 %.

The response of control system with different value of N_2 is depicted in figure 9. The experiment was carried out with N_2 = 2. With decreasing value of N_2, the neural networks controller was trained in 20 epochs and required 365 seconds for training phase. After training phase had been completely done and the best weight parameters were found, then the neural networks controller was used to control the nonlinear system with sampling time=0.03 seconds. In general, the performance of the NMPC (N_2=2) was same as the previous experiment that the good set point tracking was achieved, but the amplitude of control signal was a little higher than that of the previous work.

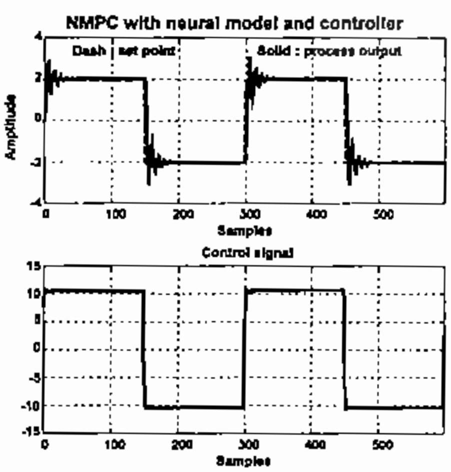

Figure 9.
Response of NMPC with neural model and controller (N_2 = 2)

The weakness of NMPC with neural networks is in training of the controller. The controller should be trained if parameters of the objective function were changed.

For comparation purposes, the response of NMPC with optimization was shown in figure 10. This optimization was solved using Quasi Newton method. The inverse of Hessian matrix was approximated using BFGS (Broyden-Fletcher-Goldfarb-Shanno) method. Due to the optimization must be solved in every sampling, the sampling time in this case should be able to handle the time requirements for calculation routine. Therefore, sampling time in this experiment was 0.09 seconds.

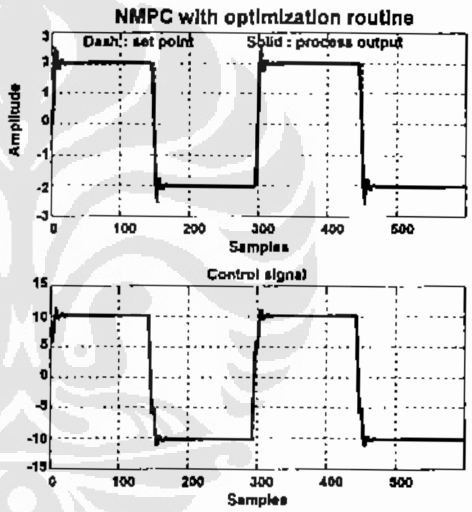

Figure 10.
Response of NMPC with optimization

The value N_2 in this algorithm is 7 and will be compared to figure 8. In general, the NMPC with optimization can produce a good performance with smaller value of the overshoot (about 25%). However, the settling time of NMPC with optimization was 1.8 seconds. This algorithm also produced higher amplitude of the control signal and showed small fluctuation.

6. Conclusion

The development of nonlinear model predictive (NMPC) using neural networks model and controller has been presented in this paper. The neural model was used to build up the nonlinear process model, while the neural controller was used as nonlinear controller to generate the control signal. The neural networks controller replaced the optimization routine in NMPC algorithm.

Simulation results show that the designed control system can handle the nonlinear process and yield the good performance. The good set point tracking was achieved with the control algorithm. Compare to NMPC with optimization, the NMPC with neural controller can yield smaller settling time and smoother control signal profiles. However, as consequences of small settling time, the NMPC with neural controller generated larger overshoot.

References

1 Demetri Psaltis, Athanasias Sideris, Alan A. Yamamura (1988), A Multilayered Neural Networks Controller, *IEEE Control System Magazine*, pp. 17-21, April.

2 Camacho E.F, Bordons C (2000), Model Predictive Control, *Springer Verlag*.

3 Cybenko G (1989), Approximation by Super-position of A Sigmoid Function, *Mathematics of Control, Signal, and Systems*, Vol. 2(4), pp. 303-314.

4 Carlos E. Garcia, David M. Prett, Manfred Morari (1989), Model Predictive Control: Theory and Practice – a Survey, *Automatica*, Vol. 25(3), pp. 335-348.

5 Narendra, K.S., Parthasarathy, K. (1990), Identification and Control of Dynamical Systems Using Neural Networks, *IEEE Transactions on Neural Networks*, Vol. 1(1), pp. 4-27.

6 Nelles O, Isermann R (1996), Basis Function Networks for Interpolation of Local Linear Models, *Proc. of 35th Conference on Decision and Control*, Kobe, Japan, December.

7 Galvan I.M, Zaldivar J.M (1997), Application of Recurrent Neural Networks in Batch Reactors. Part I: NARMA Modelling of The Dynamics Behaviour of The Heat Transfer Fluid Temperature, *Chemical Engineering and Processing*, 36, 505 – 518.

8 I.M.Galvan, J.M. Zaldivar (1998), Application of recurrent neural networks in batch reactors. Part II: Nonlinear inverse and predictive control of the heat transfer fluid temperature, *Chemical Engineering and Processing*, Vol. 37, pp. 149-161.

9 Henson, M.A (1998), Nonlinear Model Predictive Control: Current Status and Future Directions, *Computers and Chemical Engineering*, Vol. 23, pp. 187-202.

10 Norgaard M, Ravn O, Poulsen NK, Hansen LK (2000), Neural Networks for Modelling and Control of Dynamics Systems, *Springer Verlag*, London.