International Journal of Innovative Computing, Information and Control
Volume **9**, Number **3**, March **2013**
 ISSN 1349-4198
pp. 1245–1257

ADAPTIVE NEURAL NETWORK MODEL PREDICTIVE CONTROL

Ramdane Hedjar

Computer Engineering Department
College of Computer and Information Sciences
King Saud University
Riyadh 11543, Kingdom of Saudi Arabia
hedjar@ksu.edu.sa

Received January 2012; revised May 2012

Abstract. *Neural network model predictive controllers have demonstrated high potential in the non-conventional branch of nonlinear control. However, the major issue in process control of nonlinear systems is the sensitivity to parameters variations and uncertainties. Indeed, when the process is controlled by neural network model predictive control (NNMPC) and subject to parameters variations or uncertainties, unsatisfactory tracking performances are obtained. To overcome this problem, we propose in this paper an adaptive neural network model predictive control (ANNMPC) where a neural model identification block is incorporated in the scheme and online update of the weights is provided when the process is subject to parameters variations and uncertainties. Simulations have been carried out to show the robustness of this control algorithm.*

Keywords: Predictive control, Adaptive system, Neural network, Parameters variations and uncertainties

1. Introduction.

Model predictive control (MPC), which is more advanced than the well-known PID control, has achieved great success in practical applications in recent decades. Indeed, we find more than 2000 applications of this controller that have been reported in [1]. The concept of MPC, introduced in the late seventies, has nowadays evolved to a mature level and become an attractive control strategy implemented in a variety of process industries. One of the key advantages of MPC is its ability to deal with input and output constraints while it can be applied to multivariable process control [2,3,29]. Till now, most of the implemented predictive controllers (Generalized Predictive Controllers) have used linear models even if most of physical systems are nonlinear. However, linear MPC always results in poor performance for strong nonlinear models. For this reason some authors combined MPC with other nonlinear approaches to improve the tracking performances [30-32]. In order to use MPC to control a highly nonlinear system, nonlinear model has to be used. Consequently, an online nonlinear optimization problem has to be performed. The optimization problem can be non-convex and for the fast dynamics systems the optimal solution should be determined within a slot of time (time processing) [4,5,33]. However, no effective method that solves this problem exists. Moreover, nonlinear model predictive control (NMPC) relies on a mathematical model of the process for the prediction. Thus, the success of NMPC is highly dependent on having a reliable mathematical model. It is very important to look for a model that may effectively describe the nonlinear behavior of the system and should also be easily usable in designing the NMPC algorithm. However, a large number of nonlinear industrial processes may have nonlinear complex models with parameters uncertainties. To this end, a potential method is to use neural network model predictive control instead of NMPC.

Neural network which is able to approximate any continuous nonlinear function has been used for modeling and control of complex nonlinear processes [6]. Neural networks with time delay have also been used for switching system [34] and for discrete-time stochastic system with Markovian jumping parameters [35].

In neural network model predictive control (NNMPC), neural network is used to model the unknown process and there are typically two steps which are combined to design the NNMPC algorithm:

- System identification using neural network (offline);
- MPC control design using neural network model as a predictor.

Many methods have been proposed in the literature that combine neural network and model predictive control algorithm. For instance, in [7], the authors have applied nonlinear predictive control using neural networks to affine nonlinear systems. It is shown that the non-linear programming techniques can be avoided using a set of affine nonlinear predictors.

In [8], a multivariable neural network modeling and neural network model predictive control technique are applied to a steel pickling industry. The highly nonlinear dynamic behavior of the process, which is multivariable in nature, and with interaction between states causes difficulties to control this kind of system by conventional controllers. Therefore, the contribution of this work is the application of an iterative multistep neural network prediction model in a predictive control strategy for controlling such a nonlinear system. Pand and Wang [9] have applied two neural network models for model predictive control based on linear and quadratic programming formulation. Both neural networks have good convergence performance and low computational complexity. In [10], Smith et al. have presented the design and experimental validation of a nonlinear multivariable predictive controller for an educational 3-DOF helicopter system. The control strategy *approximate predictive control* based on neural network model of nonlinear plant and its linearization at each sampling instant have been used to generate the control signal. The authors in [11] have used an adaptive algorithm to update RBF models which increase the complexity of the algorithm. Moreover, persistent excitation condition should be respected to ensure the convergence of the algorithm. An adaptive neural network predictor has been used for tracking control of a nonlinear system with unknown time-delay [36].

In [12], the authors have used closed loop system identification with PI controller to model a five stage evaporator. Recurrent neural networks (RNNs) are able to provide long-range predictions, even in the presence of the measurement noise. However, the main drawback is in their training process due to the large number of sensitivity equations to be solved in the associated nonlinear optimization problem.

Neural generalized predictive control (NGPC) was applied to a three-joint robotic manipulator in [13]. NNMPC presented in [14] is applied to a float column and has achieved good performances with regards to a conventional PI controller. Recurrent neural networks have been combined with fuzzy logic approach to design a model predictive control in [15]. The proposed algorithm can be used to control a large class of industrial processes with satisfactory performance under set point and load changes. The radial basic function neural network is used in [16] to model the dynamics of distributed parameters systems. The efficiencies of the proposed MPC formulation have been tested in a tabular reactor and have produced reasonable results. The work in [17] describes an algorithm for neural models training (identification) which directly takes into account the specific fact that these models are next used recursively in MPC for long range prediction. In [18], the authors have used a new optimization algorithm-based TCPSO to obtain the optimal input for NNMPC.

We have to note that most of the previous algorithms are complex and the robustness with regards to parameters variations of the process has not been reported.

Many attempts have been made to increase the robustness of NNMPC to model mismatching and disturbances. Indeed, using neural network model as the predictor in NNMPC always causes a steady state tracking error due to the parameters variations of the process since the neural network modeling and identification process is done offline (batch form) [19]. Different approaches are used to eliminate this steady state error. For example, Akesson et al. [20] directly subtracted the approximated prediction error from output of neural network model to obtain offset free responses. Kuure-Kinsey et al. [21] used Kalman filter to remove the steady state error. Jazayeri et al. [22] estimated the external disturbances and model mismatches to eliminate the steady state error.

In this paper, we present a new neural network model predictive control of nonlinear systems that can deal with uncertainties and high parameters variations. Using simulation, we will show that the NNMPC is sensitive to parameters uncertainties and a steady state error will be induced. To overcome this problem, a new neural network block is added to the NNMPC scheme. The proposed algorithm is based on a neural network model of the process that is able to correct itself as new information is available. We have to note that adaptive neural network mechanism has been used for prediction to compensate time-delay and nonlinearity in [36]. In this work, the adaptive neural network mechanism is used to compensate the steady-state error due to parameters variations. Indeed, the proposed new scheme is simple, robust against parameters uncertainties, and does not need to estimate the parameters uncertainties or to subtract the predicted error from the output [20-22]. Thus, the steady-state error induced by the parameters variations during the operation is eliminated. The effectiveness of the proposed algorithm will be verified through its application to a nonlinear system with parameter variations.

Therefore, the proposed method is adequate for model predictive control of nonlinear systems subject to parameter variations and/or uncertainties. For instance, this method can be used to control different machines (DC motor, AC motor, PMS motor, etc.) where resistances are subject to variations due to overheating of the machine during operation. The proposed method can also be used to control complex nonlinear system where it is difficult to obtain explicitly the mathematical model of the plant (chemical processes). Further, the main drawback of NMPC is the time processing needed to reach the optimal solution. In this work neural networks have been chosen for their parallel structure which can decrease considerably the time processing especially for fast dynamic systems.

The remaining parts of the paper are outlined as follows. Section 2 deals with the problem statement and preliminaries. This section describes the neural network model predictive control for a class of nonlinear systems represented by nonlinear autoregressive moving averaging model (NARMA). Main results are given in Section 3 where an adaptive mechanism is added to the previous neural predictive controller to overcome the problem induced by the parameters variations. To see the effectiveness of the proposed algorithm, simulations are performed for both matched and mismatched case in Section 4. Section 5 concludes this paper.

2. **Problem Statement and Preliminaries.** In industrial processes, model predictive control has been widely used for set point tracking. Since most processes in industry contain non-linearity, therefore predictive control based on nonlinear models (NARMA polynomial model) is more convenient to deal with these kinds of nonlinear systems. In this work, we consider the class of systems described by the following nonlinear difference equation (NARMA model):

$$y(k+1) = f(y(k), y(k-1), \cdots, y(k-n), u(k-1), \cdots, u(k-m)) \tag{1}$$

where n and m are the known structure orders of the system. $u(k)$ and $y(k)$ are the scalar input and output of the plant respectively. $f(.)$ is the unknown nonlinear function. For the sake of simplicity we will consider SISO systems and the proposed method could be extended to MIMO systems. The purpose of the control algorithm is to select a proper control signal $u(k)$, such that the output of the plant $y(k)$ is made as close as possible to a prescribed set point $y_{ref}(k)$. To this end, we have to minimize the cost function which is based on the square predicted error and it is represented by

$$J(N_1, N_2, \Delta u) = \sum_{j=N_1}^{N_2} (y_{ref}(k+j) - \hat{y}(k+j))^2 + \sum_{j=1}^{N_u} \alpha(j) \Delta u(k+j)^2 \tag{2}$$

where $\hat{y}(k+j)$ is the predicted output signal, $y_{ref}(k)$ is the reference signal, $\Delta u(k)$ is the control action increment, α is the weight of the control action, N_1 is the minimum horizon prediction, N_2 is the maximum horizon prediction, and N_u is the control horizon. The objective of the control problem is to minimize the cost function J with respect to the control action.

2.1. **Neural network model predictive control.** The original idea of applying artificial neural network (ANN) was to imitate the way the human brain processes information. For our purpose, ANN will simply be regarded as a convenient way to model the nonlinear input-output mapping of the process.

A. Identification of Nonlinear System: The structure for identification of a nonlinear model is shown in Figure 1 and requires that the error between the neural output $y_{NN}(k)$ and the output of the plant $y(k)$ be back propagated. In this case, the cost function to be minimized is

$$\bar{J} = \frac{1}{2} \sum_k (y(k) - y_{NN}(k))^2 \tag{3}$$

The output neural network will be modeled by

$$y_{NN}(k) = f_{NN}(U(k-1), Y(k-1)), \tag{4}$$

where $f_{NN}(.)$ denotes the input-output transfer function of the neural network which replaces the nonlinear model given in Equation (1). $U(k-1)$, $Y(k-1)$ are vectors which contain m and n delayed elements of u and y respectively starting from the time instant $(k-1)$, i.e.,

$$U(k-1) = |u(k-1) \;\; u(k-2) \;\; \cdots \;\; u(k-m)|^T,$$
$$Y(k-1) = |y(k-1) \;\; y(k-2) \;\; \cdots \;\; y(k-n)|^T$$

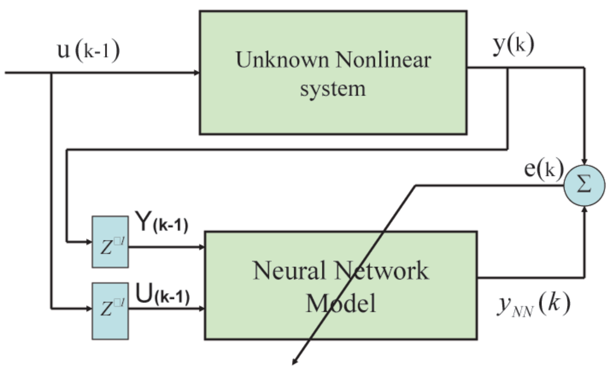

FIGURE 1. Identification by NN of the nonlinear system

In this work, the multilayer feed forward neural network architecture [23], [24] has been adopted to approximate the nonlinear system. The neural network structure of neural network model (NNM) has a two-layer perceptron network and one output variable $y_{NN}(k)$. Therefore, the structure is: $\Re_{n+m,N,1}$ with $(n+m)$ inputs, one hidden layer (N is the number of nonlinear neurons) and one linear neuron as output.

Note that it is a known fact that a feedforward neural network with one hidden layer is a universal approximator [23]. Indeed, Cybenko has demonstrated that a single hidden layer is sufficient to uniformly approximate any continuous nonlinear function [25].

The neural network model (NNM) has the following input/output mapping relationship:

$$y_{NN}(k) = \sum_{j=1}^{N} W_j^o \rho_j (W_j^u\, U(k-1) + W_j^y\, Y(k-1) + b_j) + b, \tag{5}$$

where

ρ_j represents the activation function for the j^{th} neuron from the hidden layer;

W_j^u represents the weight vector (row vector) for the j^{th} neuron with respect to the inputs stored in $U(k-1)$;

W_j^y represents the weight vector (row vector) for the j^{th} neuron with respect to the inputs stored in $Y(k-1)$;

b_j represents the bias for the j^{th} neuron from the hidden layer;

W_j^o represents the weight for the output layer corresponding to the j^{th} neuron from the hidden layer;

b represents the bias for the output layer.

To determine the mathematical model of the nonlinear system, the weights are updated by using the cost function defined in (3). The weights can be recursively adjusted in order to reduce the cost function $\overline{J}(k)$ to its minimum value by the gradient descent method. The weights are updated using:

$$W(k+1) = W(k) - \mu \frac{\partial \overline{J}(k)}{\partial W(k)},$$

where μ is a positive learning rate.

Once the learning phase of NNM has been completed, the trained model can be used to obtain a *j-step-ahead* prediction by using Equation (5).

B. Neural Network model predictive control: Predictive control is based on the prediction of the future behavior of the process to be controlled. This prediction is obtained by a recursive prediction of the neural network predictor. Thus, the nonlinear model of the process is represented by the neural network with constant weights.

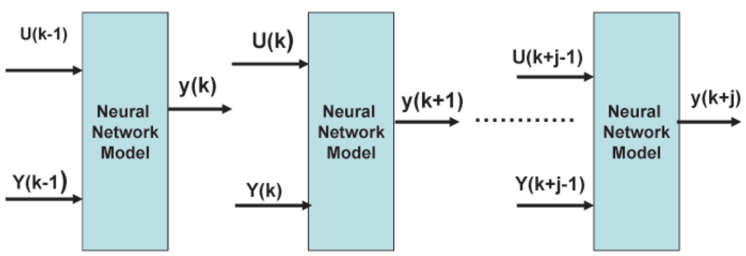

FIGURE 2. Recursion prediction of the future process outputs by neural networks

The output of the neural network model (NNM) at the k time instant is

$$y_{NN}(k) = \sum_{j=1}^{N} W_j^o \rho_j(W_j^u U(k-1) + W_j^y Y(k-1) + b_j) + b, \tag{6}$$

where

$$U(k-1) = |u(k-1) \;\; u(k-2) \;\; \cdots \;\; u(k-m)|^T,$$
$$Y(k-1) = |y(k-1) \;\; y(k-2) \;\; \cdots \;\; y(k-n)|^T$$

A sequential algorithm based on the knowledge of the current and past values of u and y with the neural network model gives the *j-step-ahead* neural network predictor. Figure 2 represents the construction of the *j-step-ahead* predictor and it is modeled by the equation:

$$y_{NN}(k+j) = \sum_{j=1}^{N} W_j^o \rho_j(W_j^u U(k+j-1) + W_j^y Y(k+j-1) + b_j) + b, \tag{7}$$

where

$$U(k+j-1) = |u(k+j-1) \;\; u(k+j-2) \;\; \cdots \;\; u(k+j-m)|^T,$$
$$Y(k+j-1) = |y(k+j-1) \;\; y(k+j-2) \;\; \cdots \;\; y(k+j-n)|^T.$$

The predictive control algorithm will use the neural network predictor shown in Figure 2 to calculate the future control signal.

In predictive control, the objective function to minimize over a finite prediction horizon is a function based on the error between the NNM predicted output, the predicted reference trajectory and the weighted control signal:

$$J(U,k) = \frac{1}{2} \sum_{j=N_1}^{N_2} (y_{ref}(k+j) - y_{NN}(k+j))^2 + \lambda \sum_{j=1}^{N_u} u(k+j-1)^2 \tag{8}$$

The optimal control signal is found by minimizing the cost function J with respect to $U(k)$ over the prediction horizon using Equation (7). One of the most common rules used in NNMPC to update the control action is the deceasing gradient rule [26], in which the actualization is made in the direction of the negative gradient of the function. In this work a modified decreasing gradient is used and it can be expressed in the form:

$$U(k+1) = U(k) - \eta \left(\left| \frac{\partial J(U,k)}{\partial U(k)} \right|^2 + \delta I \right)^{-1} \frac{\partial J(U,k)}{\partial U(k)} \tag{9}$$

where $\frac{\partial J(U,k)}{\partial U(k)}$ is the gradient of the cost function, η is a small positive step size, and δ is a small positive constant chosen to ensure the existence of the inverse. The structure of the ANN predictive control is depicted in Figure 3.

3. Main Results: *Adaptive Neural Network Model Predictive Control.*

In the last years, numerous adaptive control techniques have been proposed to replace the conventional classical methods [24]. The ability to adapt to variations in plant dynamics and environment automatically has made such adaptive controllers increasingly important for various applications. Before they can be implemented, mathematical modeling of the plant has to be done. This task is sometimes difficult and laborious. In addition, inaccuracy in the modeling of the plant could lead to degraded performances of the controllers. Artificial neural networks are trainable dynamical systems that estimate input-output functions, and are sensitive to parameters variations (a steady state error will be induced in mismatched case). Thus, to eliminate this induced steady-state error

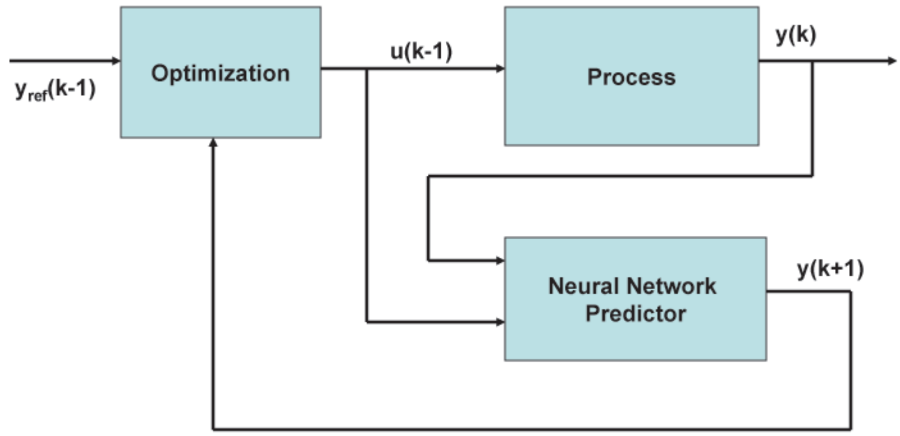

FIGURE 3. Neural network model predictive control

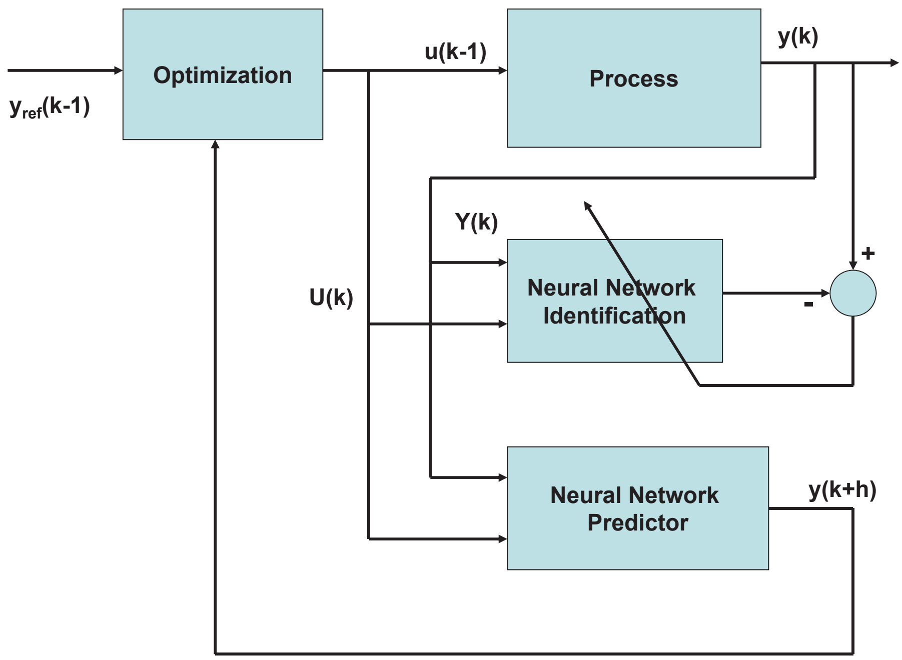

FIGURE 4. Adaptive neural network model predictive controller

a new neural network block (adaptive mechanism) is added to the scheme in order to update the weights of the neural network model online. The proposed neural network structure of this adaptive neural network controller is shown in Figure 4.

This scheme uses two sub-networks (NNI) as a neural model and (NNP) as a neural predictor. The sub-network (NNI) represents the neural model obtained in the identification phase. This sub-network (NNI) is used to update the weights when there are uncertainties or parameters variations of the plant. The error between the output $y(k)$ and the neural model $y_{NN}(k)$ is used to update the weights of the neural model. The obtained new weights are used by the neural network predictor block and the optimization block in order to calculate the predictive control action to be applied to the process.

4. **Numerical Examples.** In this work two examples have been used to check the effectiveness of the proposed algorithm. Moreover, for simplicity a *one step-ahead* predictive control with $\lambda = 0$ has been adopted. Thus, the cost function to minimize is

$$J = \frac{1}{2}e(k+1)^2$$

where $e(k+1) = y_{ref}(k+1) - y_{NN}(k+1)$, $y_{NN}(k+1) = \sum_{i=1}^{N} W^o(i)\sigma(v(i)) + b^o$ and $v(i) = \sum_{l=1}^{n} W^y(i,l)y(k-l+1) + \sum_{l=1}^{m} W^u(i,l)u(k-l+1) + b(i)$.

By using the chain rule, we can easily determine the gradient as

$$\begin{aligned} p(k) &= -\frac{\partial J(k)}{\partial u(k)} \\ &= -e(k+1)\frac{\partial e(k+1)}{\partial u(k)} = e(k+1)\frac{\partial y_{NN}(k+1)}{\partial u(k)} \\ &= e(k+1)\sum_{i=1}^{N} W^o(i)\sigma(v(i))W^u(i,1) \end{aligned}$$

where $\sigma\left(v(i) = \frac{d\sigma}{dv(i)}\right)$ is the derivative of the activation function with respect to $v(i)$. The updated control signal is

$$u(k+1) = u(k) + \eta\frac{p(k)}{p(k)^2 + \delta}$$

4.1. **Example 1: NNMPC.** The nonlinear model of the plant, which is taken from the benchmark models used in [27], was considered in this work and is given by

$$y(k+1) = \frac{y(k)y(k-1)\left(y(k)+1\right)}{1 + y(k)^2 + y(k-1)^2} + bu(k) \tag{10}$$

where y is the measured output of the plant and u is the applied control input.

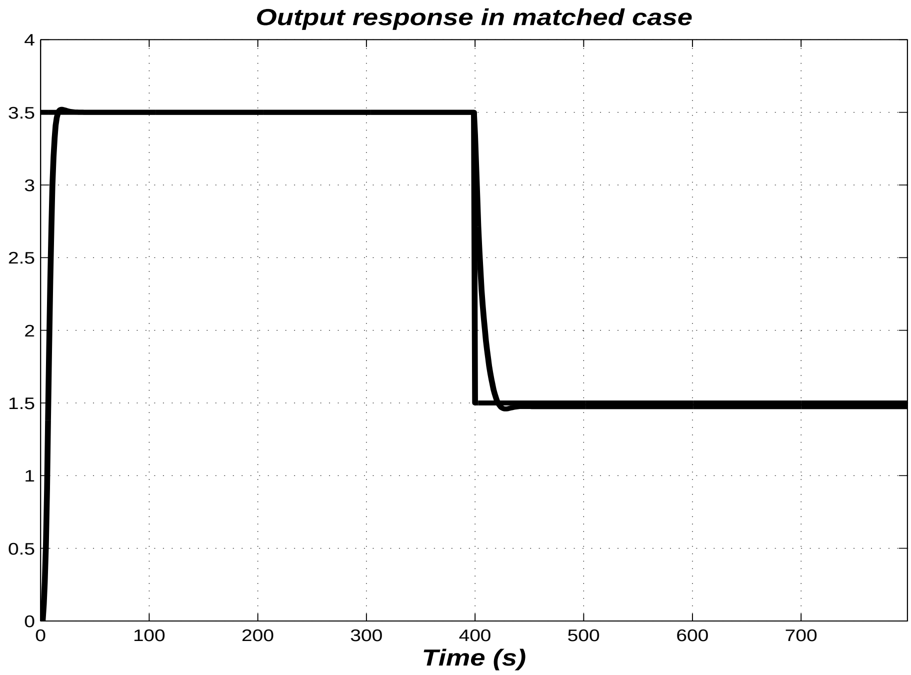

FIGURE 5. Output response and the control signal in the mismatched case

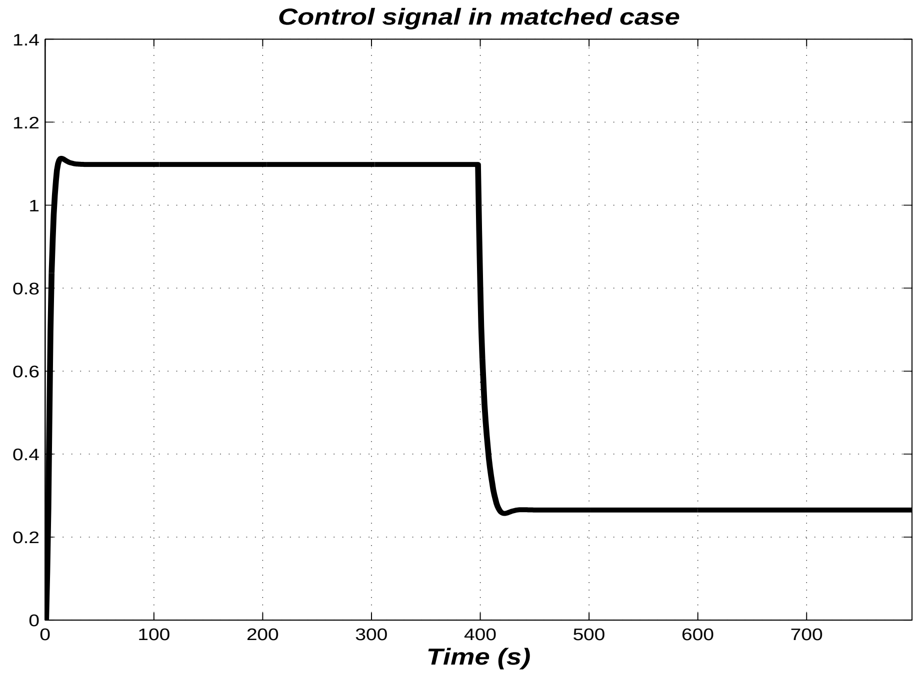

FIGURE 6. The applied control signal in matched case

The reference model is given by

$$y_{ref} = \begin{cases} 3.5 & \text{for} \quad 0 \leq t < 400 \\ 1.5 & \text{for} \quad 400 \leq t < 800. \end{cases}$$

The structure of the neural network is $\Re_{3,8,1}$. For the learning method, the Levenberg-Marquardt method has been used. During the learning phase the parameter b is set to 1.5. The neural model (Figure 1) has been trained using a selection of training patterns with a learning rate set to $\mu = 0.05$. Afterwards, this neural model is used to derive the predictive controller of the plant as it is shown in Figure 3. Note that, one-step-ahead predictive controller is used in this simulation ($h = 1$) and the parameters of the modified gradient descent method algorithm are $\delta = 0.01$ and $\eta = 0.05$. Figure 5 shows the good tracking performance achieved by the controller in matched case ($b = 1.5$). The applied control signal is depicted in Figure 6.

In the mismatched case, for $k \geq 300$ the parameter b is set to 1. The same neural network controller is used and the simulation result is illustrated in Figure 7. This figure depicts the tracking performances. The variation of the parameter b induces a steady state tracking error. Hence, we conclude that the neural network predictive control is sensitive to parameters variations when the identification phase is performed offline (batch form).

4.2. **Adaptive neural network model predictive control for Example 1: ANN-MPC.** The same example seen previously is tested with the same parameters and the structure of the neural network model. The error $e(k) = y(k) - y_{NN}(k)$ is used to update the weights of the neural model (NNM) (see Figure 4). These weights are used by the neural predictor (NNP). The neural prediction equation is utilized by the optimization block (modified gradient descent algorithm) to determine the predictive control signal. The simulation results are shown in Figure 8. The output of the system is close to the desired signal although the parameter value b is unknown after $k = 300$ (set to $b = 1$ and the neural network has been trained with $b = 1.5$). Thus, the adaptive neural predictive

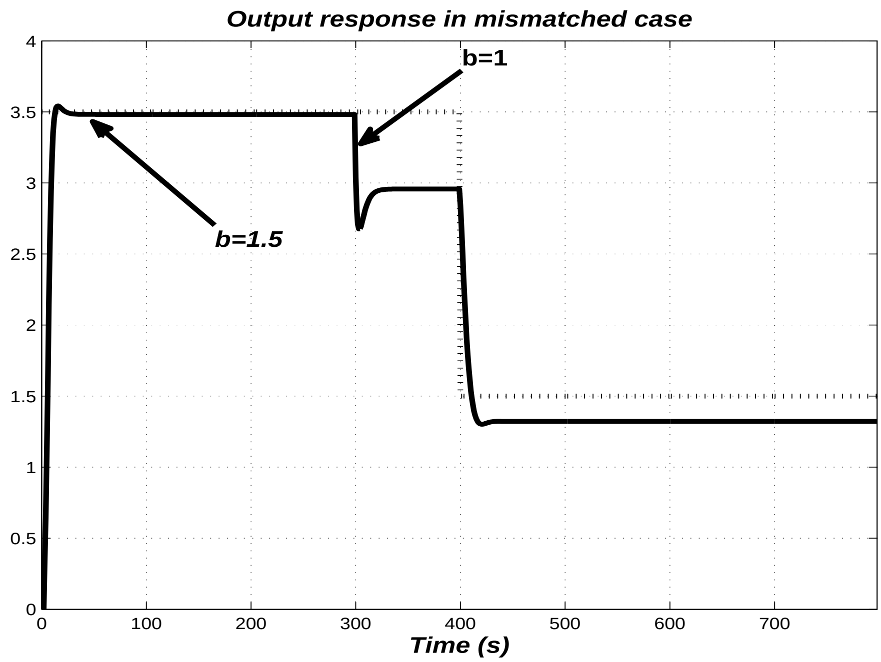

FIGURE 7. Output response in the mismatched case ($b = 1$ for $k \geq 300$)

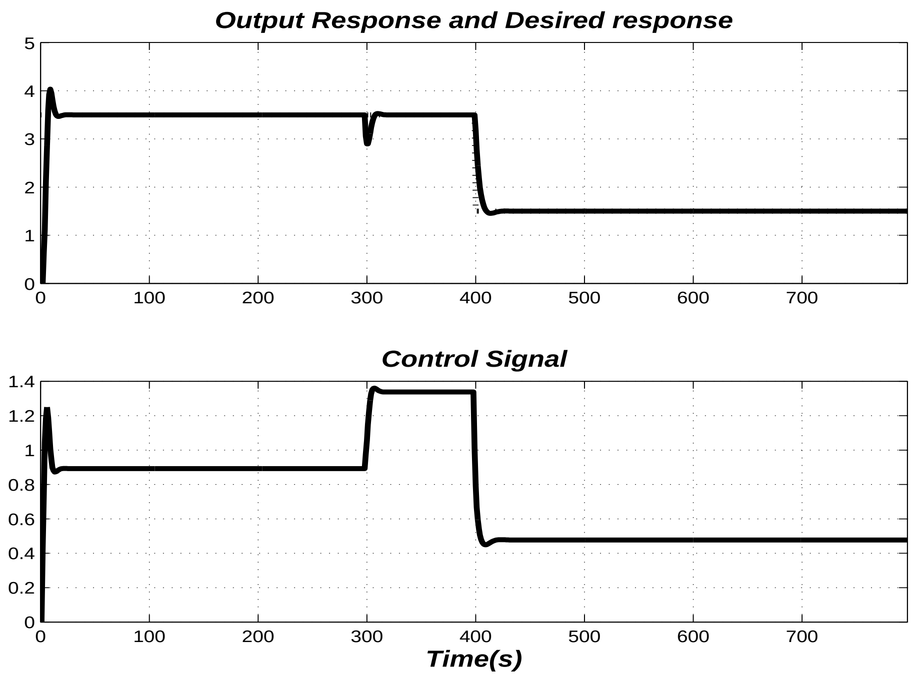

FIGURE 8. Output response and the control signal in the mismatched case

controller has achieved best tracking error performance with regards to neural predictive control (Figure 7).

4.3. **Example 2.** In this second part, another highly nonlinear model (NARMA), utilized in [28] as a benchmark, is used in this work to check the robustness of the NNMPC with regards to an unknown model. The NARMA model is given by

$$y(k+1) = \sin(y(k-1)) + u(k)\,(b + \cos(u(k)y(k)))$$

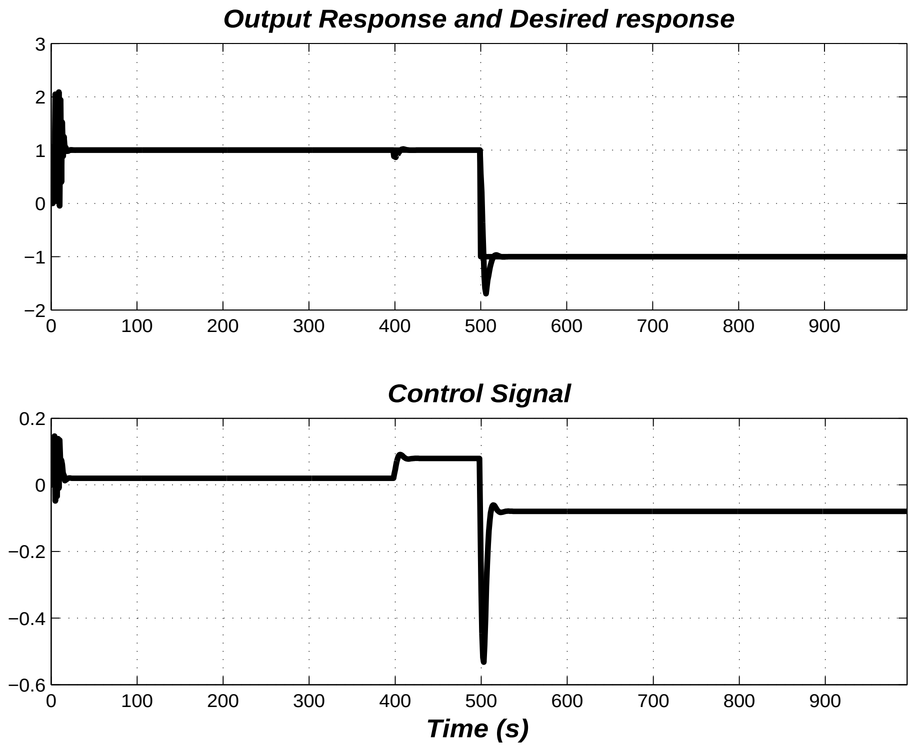

FIGURE 9. Output response and the control signal in the mismatched case (Example 2)

Note that parameter b is unknown and variable. In this mismatched case, parameter b is taken as

$$b = \begin{cases} 7 & \text{for} \quad 0 \le t < 400 \\ 1 & \text{for} \quad 400 \le t < 800 \end{cases}$$

and the reference is

$$y_{ref} = \begin{cases} 1 & \text{for} \quad 0 \le t \le 500 \\ -1 & \text{for} \quad 500 \le t \le 800. \end{cases}$$

The neural network structure and all parameters are the same as in the first example and only the learning parameter has been increased to $\mu = 0.1$ in order to reduce oscillations in the start up. Figure 9 shows the good tracking performance achieved by the proposed controller in spite of the unknown parameters of the process. Note that in Figure 9, oscillations have appeared in the startup due to the initial weights of neural networks which were chosen randomly. Consequently, the adaptive neural predictive controller has achieved a good tracking performance in spite of the unknown model with parameters variations.

4.4. **Convergence of the algorithm.** Feedforward neural networks are universal approximators, which means that any continuous nonlinear function can be approximated by feedforward neural networks [25]. In this work the adaptive neural network predictive control algorithm converges to the desired reference signal if the dynamic of the identification algorithm and predictive control algorithm have different rates of convergence. Indeed, in the first part of the first example, we deduced that uncertainties and/or parameters variations induce a steady state error in the tracking performance. To eliminate this steady state error an adaptive mechanism is introduced in the scheme to ensure the convergence of the tracking performance to zero. Thus, the dynamic of the identification algorithm must be very high compared to the dynamic of the control process. Consequently, we used in the first example, the Levenberg-Marquardt algorithm for identification which is very fast with regards to the gradient method used for control processing. In the second example, both identification and control processing have used the gradient descent method. However, the learning rate of the identification algorithm has been increased to ensure rapid convergence of the tracking error to zero and to reduce the oscillations around the reference signal.

5. **Conclusions.** In this paper, an online adaptive model predictive control of nonlinear system using neural networks is presented. First, it is shown that the neural network predictive controller is sensitive to parameters variations or/and uncertainties. Indeed, unsatisfactory tracking performances have been obtained when the parameters are unknown or variable. To enhance the robustness of this neural model predictive control algorithm with respect to parameter variation or/and uncertainties a sub-neural model identification is added to the structure scheme and weights are updated online. The proposed new scheme is simple, robust against parameters uncertainties, and does not need to estimate parameters uncertainties or to subtract the predicted error from the output.

Simulation results have shown that good tracking performance has been obtained although parameters variations are unknown and/or the mathematical model of the process is unknown to the neural network predictive controller. Additional research should be oriented firstly towards the application of the adaptive neural networks predictive control on other systems models, especially NARMAX models where an exogenous signal is introduced that may complicate the adaptive mechanism based on the modeling error. Secondly, it should also be oriented towards the real-time implementation of this proposed control algorithm to processes that are subject to parameters variations.

Acknowledgment. This work is supported by the research center of College of Computer and Information Sciences-King Saud University.

REFERENCES

[1] G. C. Goodwin, S. F. Graebe and M. E. Salgado, *Control System Design*, Prentice-Hall, Upper Saddle River, New-Jersey, 2001.
[2] D. W. Clarke, C. Mohtadi and P. S. Tuffs, Generalized predictive control, Part I: The basic algorithm, Part II: Extension and interpretation, *Automatica*, vol.32, pp.137-160, 1987.
[3] E. F. Camacho and C. Bordons, *Model Predictive Control*, 2 Edition, Springer-Verlag, London, 2004.
[4] R. Findeisen, F. Allgwer and L. T. Biegler, Assessment and future directions of nonlinear model predictive control, *Lecture Notes in Control and Information Sciences*, vol.358, 2007.
[5] L. Magni, D. M. Raimondo and F. Allgwer, Nonlinear model predictive control: Towards new challenging applications, *Lecture Notes in Control and Information Sciences*, vol.384, 2009.
[6] K. S. Narendra and K. Parthasarathy, Identification and control of dynamical systems using neural networks, *IEEE Transactions on Neural Networks*, vol.1, pp.4-27, 1990.
[7] P. Liu, V. Kadirkamanathan and S. A. Billinks, Predictive control for nonlinear systems using neural networks, *International Journal of Control*, vol.71, no.6, pp.1119-1132, 1998.
[8] P. Kittisupakom, P. Thitiyasook, M. A. Hussein and W. Daosud, Neural network based model predictive control for a steel pickling process, *Journal of Process Control*, vol.19, no.4, pp.579-590, 2009.
[9] Y. Pand and J. Wang, Two neural network approaches to model predictive control, *American Control Conference*, Washington, 2008.
[10] J. Smith, S. S. Boonto and H. Werner, Approximate model predictive control of 3-DOF helicopter, *Proc. of the 46th IEEE Conference on Decision and Control*, USA, pp.4501, 2007.
[11] A. Alexandris and H. Sarimvers, Nonlinear adaptive model predictive control based on self-correcting neural network models, *AIChE Journal*, vol.51, no.9, pp.2495-2506, 2005.
[12] J. C. Atuonwu, Y. Cao, G. P. Rangaiah and M. O. Tad, Identification and predictive control of a multistage evaporator, *Control Engineering Practice*, vol.18, pp.1418-1428, 2010.

[13] F. Temurtas, H. Temurtas and N. Yumusak, Application of neural generalized predictive control to robotic manipulators with cubic trajectory and random disturbances, *Robotics and Autonomous Systems*, vol.54, pp.74-83, 2006.

[14] S. Mohanty, Artificial neural network based system identification and model predictive control of a float column, *Journal of Process Control*, vol.19, pp.991-999, 2009.

[15] C.-H. Lu and C.-C. Tsai, Generalized predictive control using recurrent fuzzy neural networks for industrial processes, *Journal of Process Control*, vol.17, pp.83-92, 2007.

[16] E. Aggelogiannaki and H. Sarimveis, Nonlinear model predictive control of distributed parameter systems using data driven artificial neural network models, *Computer and Chemical Engineering*, vol.32, pp.1225-1237, 2008.

[17] M. Lawrynczuk, Training of neural models for model predictive control, *Neurocomputing*, vol.73, pp.1332-1343, 2010.

[18] Y. Song, Z. Cheng and Z. Yuan, New chaotic PSO-based neural network predictive control for nonlinear process, *IEEE Trans. on Neural Network*, vol.18, no.2, pp.595-600, 2007.

[19] V. Alamdari, A. Fatehi and A. Khaki-Sedigh, Neural network mode-based predictive control for multivariable nonlinear systems, *IEEE International Conference on Control Applications, Part of 2010 IEEE Multi-Conference on Systems and Control*, Yakohama, Japan, 2010.

[20] M. Akesson, H. T. Toivonen, J. B. Waller and R. H. Nystrom, Neural network approximation of a nonlinear model predictive controller applied to a pH neutralization process, *Journal of Computers and Chemical Engineering*, vol.29, no.2, pp.323-335, 2005.

[21] M. Kuure-Kinsey, R. Cutright and B. W. Bequette, Computationally efficient neural predictive control based on a feedback architecture, *Industrial and Engineering Chemistry Research*, vol.45, pp.8575-8582, 2006.

[22] A. Jazayeri, A. Fatehi, H. Sadjadian and K. Khaki-Sedigh, Disturbace rejection in neural network model predictive control, *Proc. of the 17th IFAC World Congress*, Korea, 2008.

[23] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd Edition, Prentice Hall, 1999.

[24] J. C. Principe, N. R. Euliano and W. C. Lefebre, *Neural and Adaptive Systems: Fundamentals through Simulations*, John Wiley, 2000.

[25] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Mathematics of Control, Signals, and Systems*, vol.2, pp.303-314, 1989.

[26] D. I. Soloway and P. J. Haley, Neural generalized predictive control: A Newton-Raphson implementation, *Proc. of the 11th International Symposium on Intelligent Control*, Dearborn, pp.277-282, 1996.

[27] J. B. D. Cabera and K. S. Narendra, Issues in the application of neural networks for tracking based on inverse control, *IEEE Transactions on Automatic Control*, vol.44, no.11, pp.2007-2027, 1999.

[28] K. S. Narendra, Neural Networks for control: Theory and practice, *Proc. of the IEEE*, vol.84, no.10, pp.1385-1406, 1996.

[29] A. Yanou, A. Inoue, M. Deng and S. Masuda, An extension of two degree-of-freedom of generalized predictive control for M-input M-output systems based on state space approach, *International Journal of Innovative Computing, Information and Control*, vol.4, no.12, pp.3307-3317, 2008.

[30] Y. Xia, L. Li, G. Liu and P. Shi, H-infinity predictive control of networked control systems, *International Journal of Control*, vol.84, no.6, pp.1080-1097, 2011.

[31] Y. Xia, H. Yang, P. Shi and M. Fu, Constrained infinity-horizon model predictive control for fuzzy discrete-time systems, *IEEE Trans. on Fuzzy Systems*, vol.18, no.2, pp.429-436, 2010.

[32] Y. Xia, G. Liu, P. Shi, J. Shen and D. Rees, Robust constrained model predictive based parameters-dependent Lyapunov functions, *Circuits Systems and Signal Processing*, vol.27, pp.429-446, 2008.

[33] P. Radhakant and K. Mangal, Model predictive static programming: A computationally efficient technique for suboptimal control design, *International Journal of Innovative Computing, Information and Control*, vol.5, no.2, pp.399-411, 2009.

[34] Z. Wu, P. Shi and J. Chu, Delay-dependent stability analysis for switched neural networks with time-varying delay, *IEEE Trans. on Systems, Man and Cybernetics, Part: B Cybernetics*, vol.41, no.6, pp.1522-1530, 2011.

[35] Z. Wu, P. Shi, H. Su and J. Chu, Passivity analysis for discrete-time stochastic Markovian jump neural networks with mixed time-delays, *IEEE Trans. on Neural Networks*, vol.22, no.10, pp.1566-1575, 2011.

[36] J. Na, X. Ren, Y. Gao, Gr. Robert and C.-C. Ramon, Adaptive neural network state predictor and tracking control for nonlinear time-delay systems, *International Journal of Innovative Computing, Information and Control*, vol.6, no.2, pp.627-639, 2008.