J. King Saud Univ., Vol. 19, *Comp. & Info. Sci*., pp. 75-94, Riyadh (1427H./2007)

Online Adaptive Control of Non-linear Plants Using Neural Networks with Application to Temperature Control System

R. Hedjar
Department of Computer Engineering, College of Computer & Information Sciences,
King Saud University, Riyadh, Saudi Arabia
E-mail: hedjar@ccis.ksu.edu.sa

(Received 18 November 2006; accepted for publication 16 April 2007)

Abstract. Although the neural inverse model controllers have demonstrated high potential in the non-conventional branch of non-linear control, their sensitivity to parameter variations and/or parameter uncertainties usually discourage their applications in industry. Indeed, when the controlled system is subject to parameter variations or uncertainties, unsatisfactory tracking performances are obtained. To overcome this problem, a neural inverse model is added to the control scheme and an online update of the weights is provided. Simulations have been carried out to show the robustness of this control algorithm. Moreover, this adaptive neural inverse model controller is implemented on a temperature control system. Good tracking performances are obtained for different set points regulation. The large parameter variations and disturbances have no effect on the tracking performance since they have been compensated online.

Keywords: Neuro-control of non-linear systems, Neural inverse model, Adaptive control, Temperature control system.

1. Introduction

Making the output of a system track a given reference trajectory is a common industrial problem. To obtain satisfactory performances, the dynamics of controlled systems are usually simple (e.g. linear) and explicitly known so that modern control strategies can be successfully applied. However, when the structure of the plant is unknown or the parameter variation is excessive, the effectiveness of modern control theory diminishes. For instance, when a fixed controller setting is employed in an industrial drive system with widely changing environment, unsatisfactory performance often occurs. Even though it is possible to develop a reasonably accurate model, the resulting control algorithm is so computationally intensive that it becomes infeasible to implement in a real-time control environment. According to the performance exhibited by an experienced human operator, it is believed that a controller should be designed to have abilities to learn from experience and to use the knowledge gained during the training process.

Several tracking control techniques are evolving such as the sliding mode control [1], feedback linearization [2], and self-tuning and model reference adaptive controls [3]. The sliding mode control and feedback linearization control systems require a valid model dynamics of the plant being controlled. Thus, they are not robust in the sense that the controller is, mainly due to structured uncertainty of the controlled plants, sensitive to large parameter variations and noise. In addition, conventional adaptive control schemes require information about the plant structure and may not guarantee the stability of the system in the presence of unmodeled dynamics and noises.

In the recent years, an emerging technique that mimics the adaptive distributed architecture in the human brain, namely artificial neural networks (ANN), provides potential alternatives to tackle the ask mentioned above. Indeed, the incredible learning and adaptive capability of biological neural mechanisms have inspired many scientists and engineers to apply control methodologies on the biological counterparts. It is well known that a multilayer neural network model is basically a non-linear extension of a linear adaptive model. It possesses many advantages in the controller design, compared with conventional control methods, such as the capability of approximating arbitrary non-linear function, fault-tolerance, parallel computing and so on [4, 5]. Therefore, applications of neural networks have received considerable attention in the control of complex systems. One of the simplest approaches for the implementation of the neuro-controller is the direct inverse model control approach pioneered by Widrow and Stearns [6] and Psaltis *et al*. [7]. In the Psaltis *et al*. scheme, referred to as a general learning, a multi-layered neural network is first trained offline using a back-propagation algorithm to learn the inverse dynamic model of a plant and, once trained, it is configured as a direct controller to the plant.

Khalid *et al*. [8] compared the inverse model neural controller to three other control algorithms: fuzzy logic control, generalized predictive control and proportional-integral control. Experimental results showed that the neural controller performs very well and offers encouraging advantages. Moreover, according to the experimental studies, the best performances may be obtained by combining the neural controller with other classes of control systems.

Although, neural networks have been used for inverse modeling in number of applications, these controllers are not robust with respect to model uncertainties and parameter variations. Hence, many authors have integrated an adaptive mechanism to overcome this problem [9]. Indeed, in [10], the authors have proposed the inverse NN adaptive control that allows the controller to respond online to changes in the plant dynamics. However, this approach is complex and more computationally complicated. In [11], the authors developed an adaptive neural network by using back stepping method and higher order neural network. However, this approach is applied only for a class of discrete MIMO non-linear systems with triangular form inputs.

In this paper, we present a tracking control design using neural networks, and we will show by simulation that the neural inverse control is sensitive to parameter uncertainties and unsatisfactory tracking performances are obtained. To overcome this problem, an adaptive neural inverse model is added to the structure and tracking performances are enhanced. Moreover, to validate the effectiveness of this control scheme, the neural controller is applied to control a temperature system. The control algorithm is performed on real-time by using C++ language.

The remaining part of this paper is arranged as follows. Section 2 describes the neural network structure and learning scheme. Section 3 describes the non-linear inverse model control and it is shown that this kind of non-linear controller is sensitive to parameter uncertainties. To overcome this problem, the adaptive non-linear inverse model control is described in Section 4. In Section 5, the adaptive non-linear neural controller is applied to real-time control of temperature system. Finally, conclusions are given in Section 6.

2. Network Structure and Learning Scheme

2.1. Multilayer neural network

Although several ANN architectures have been applied to process control, most of them concentrate on multilayer neural networks (MNNs) (Fig. 1). MNNs are particularly attractive to control complex systems due to the following reasons:

- MNNs are essentially feedforward structures in which the information flows forward, from the inputs to outputs, through hidden layers.
- MNNs with one hidden layer using arbitrary sigmoidal activation function are able to perform any non-linear mapping between two finite-dimensional spaces to any desired degree of accuracy, provided there are enough number of hidden units [12].
- The basic algorithm for learning in MNNs, the back-propagation (BP) algorithm, belongs to the broad class of gradient methods largely applied in optimal control.

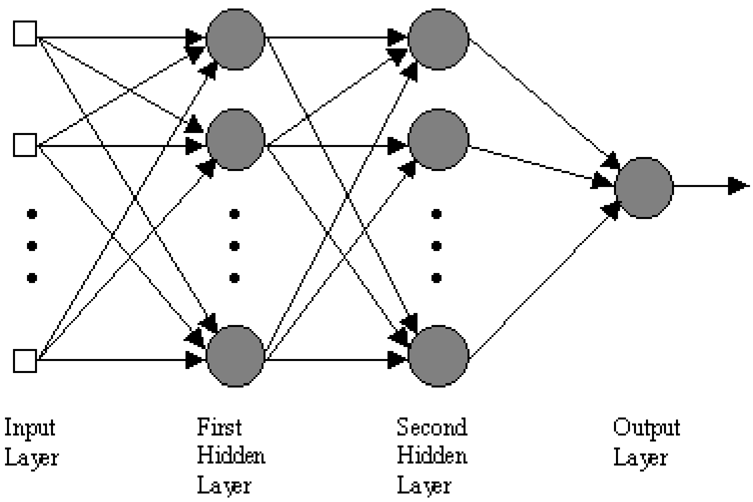

Fig. 1. Structure of multi-layer neural networks.

In this paper, the network used for this study is a two-layer feedforward neural network with the error back-propagation learning algorithm [4] that consists of input, two hidden and output layers. Each layer contains several processing elements or perceptrons with sigmoid non-linearities, expect for the output neurons where saturated linear functions are used.

2.2. Learning scheme

Back propagation algorithm: This algorithm, which performs a stochastic gradient descent, provides an effective method to train a feedforward neural network to approximate a given continuous function over a compact domain $\boldsymbol{D}$. Let $u \in \boldsymbol{D}$ be a given input. The network approximation error for this input is given by:

$$e(t) = f(u) - NN(u) \tag{1}$$

Training $NN(.)$ to closely approximate $f(.)$ over $\boldsymbol{D}$ is equivalent to minimizing the objective function:

$$J = \frac{1}{2}\sum_{j \in C} e_j^2(n) \tag{2}$$

where the set ***C*** includes all the neurons in the output layer of the network. Let $w_{ji}(n)$ denote a weight between neuron i and neuron j, the correction $\Delta w_{ji}(n)$ applied to $w_{ij}(n)$ is defined by [4, 5]:

$$\Delta w_{ji}(n) = -\eta \frac{\partial J}{\partial w_{ji}} \tag{3}$$

where η is the learning parameter of the back propagation algorithm.

The correction $\Delta w_{ji}(n)$ can be written as:

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \tag{4}$$

where $\delta_j(n) = \begin{cases} e_j(n)\varphi'(v_j(n)) & \text{if neuron } j \text{ is an output node.} \\ \varphi'(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) & \text{if neuron } j \text{ is a hidden node.} \end{cases}$

Note that $\varphi(.)$ is the activation function. In this work, a logistic function is used and it is given by: $\varphi(v_j(n)) = \dfrac{1}{1 + exp(-v_j(n))}$.

3. Non-linear Inverse Control

The representation of discrete-time dynamical system (single-input/single-output) using difference equation is currently well known. Let the controlled system be represented by:

$$y(k+1) = f\big(y(k), y(k-1), \ldots\ldots, y(k-p+1), u(k), u(k-1),\ \ldots\ldots, u(k-q)\big) \tag{6}$$

where $y \in \Re$ denotes the output, $u \in \Re$ is the input, k is the discrete time index, p and q are non-negative integers and $f(.)$ is a non-linear function. In many practical cases, the plant input is limited in amplitude, i.e. there exits $u_{\min}$ and $u_{\max}$ such that, for any k: $u_{\min} \leq u(t) \leq u_{\max}$.

It is assumed that the only available a priori acknowledge about the plant is that p and q are known.

The task is to learn how to control the plant described in Eq. (6) in order to follow a specified reference $y_d(k)$, minimizing some norm of the error:

$$e(k) = y_d(k) - y(k) \tag{7}$$

Neural networks have been used for inverse modeling in a number of control applications [17]. With reference to the general non-linear system of Eq. (6), it would seem that the inverse function generating u(k) could be represented by:

$$\begin{aligned} u(k) &= f^{-1}(y(k+1), y(k), \ldots\ldots, y(k-p), u(k-1), \ldots\ldots, u(k-q)) \\ &= NN(y(k+1), y(k), \ldots., y(k-p), u(k-1), \ldots, u(k-q)) \end{aligned} \tag{8}$$

The structure for identification of an inverse plant model is shown in Fig. 2 and requires that the error between the neural output $u(k)$ and the desired set point $y_d(k)$ be back propagated. In this case, the cost function to be minimized is:

$$J = \frac{1}{2}\sum_k \left(y_d(k) - \hat{u}(k)\right)^2$$

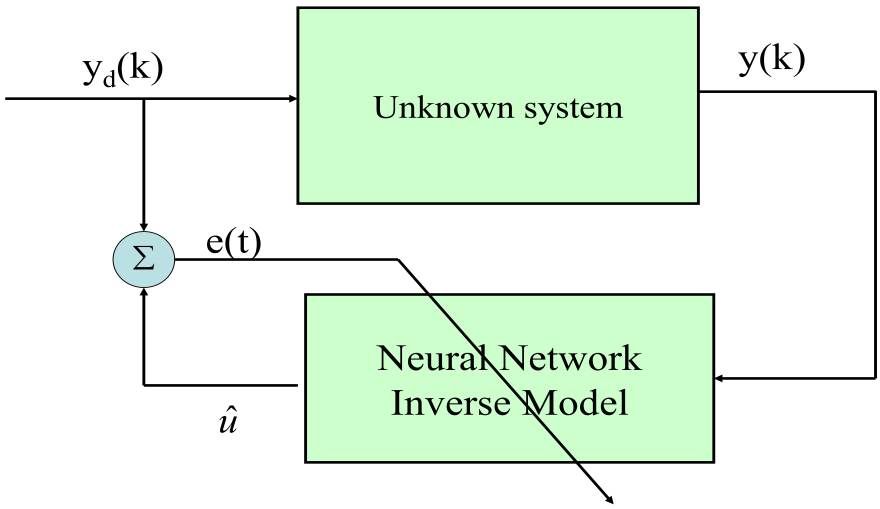

Fig. 2. Inverse model training block diagram.

The inverse model is subsequently applied as a controller for the process by inserting the desired output or reference signal $y_d(k+1)$ instead of the output $y(k+1)$. Figure 3 shows the direct inverse control scheme.

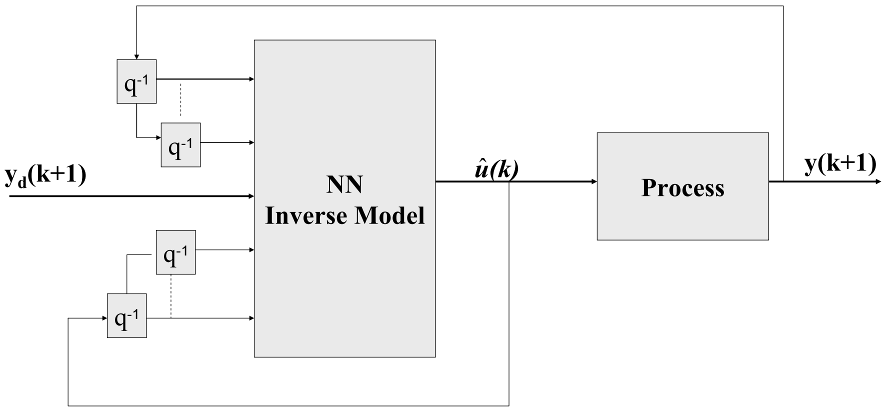

Fig. 3. Direct inverse control structure.

Learning an inverse model was one of the more viable techniques in the application of neural networks for control. An inverse model, unlike a forward model, has immediate utility for control. There has been some application of direct inverse model control and this requires non-parameters variations or no disturbances in the process. Unfortunately, process control systems are often non-linear and difficult to control accurately. Their dynamic models are more difficult to derive and they are subject to parameters variations and disturbances. In this part, we will show that this kind of processes, with direct inverse model controller, fails to achieve a good tracking performance.

Simulation example: Let the process model given by [13]:

$$y(k+1) = \frac{y(k)\,y(k-1)\big(y(k)+1.5\big)}{1+y(k)^2+y(k-1)^2} + a\,u(k) \tag{9}$$

The structure of the neural network is: $\Re_{3,7,10,1}$ since the control signal is given by: $u(k) = NN\big(y(k+1), y(k), y(k-1)\big)$

and the input signal is given by:

$$u(k) = 0.8\sin(2\pi k/100) + 0.4\sin(2\pi k/250).$$

During the learning phase, the parameter a is set to 1. The inverse neural model (Fig. 2) has been trained using a selection of training patterns taking from the open-loop plant response with learning rate set to 0.25. Afterwards, the inverse neural model is used as a direct controller of the plant as it is shown in Fig. 3, but without delayed control signal.

Figure 4 depicts the desired reference signal. Figures 5 and 6 show the good tracking performance achieved by the controller in the matched case (a=1). In the mismatched case, the parameter a is set to 1.5 and, by using the same neural controller, the simulation results are illustrated in Figs 7 and 8. Theses figures depict the tracking performances with high tracking error. Hence, we conclude that the direct inverse control is sensitive to parameter variations.

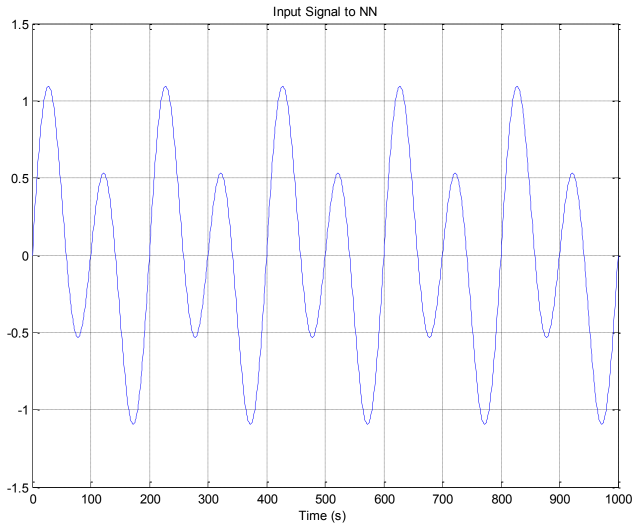

Fig. 4. The desired signal.

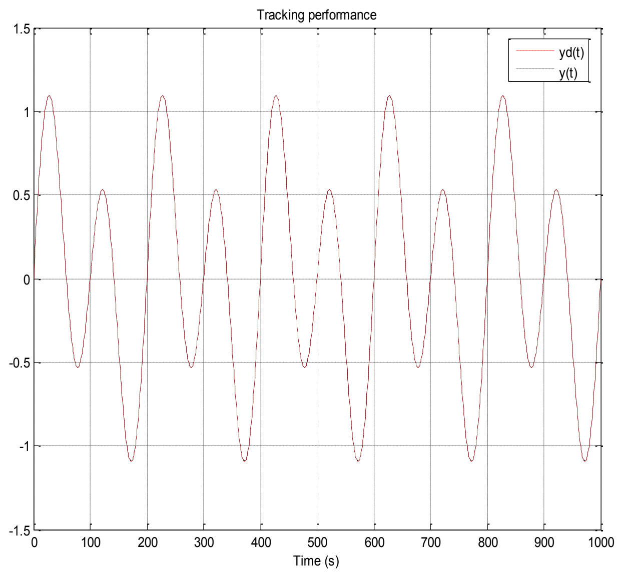

Fig. 5. The desired signal and the output of MNN when a=1.

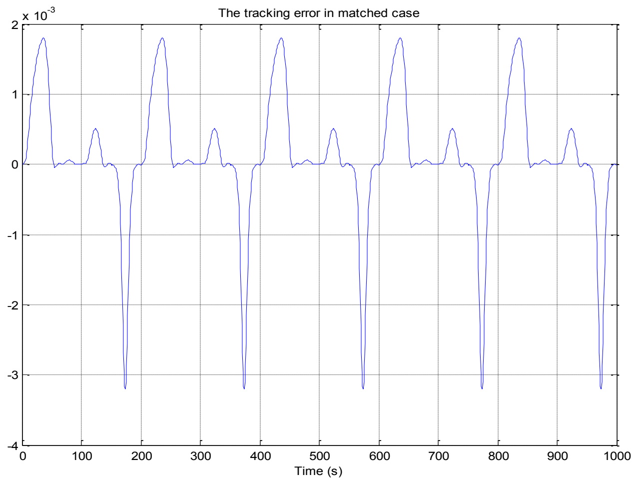

Fig. 6. The tracking error in matched case.

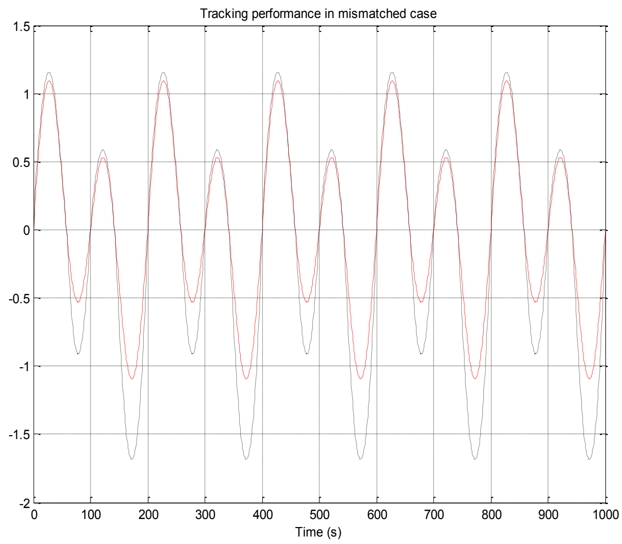

Fig. 7. The tracking performance in mismatched case (a=1.5).

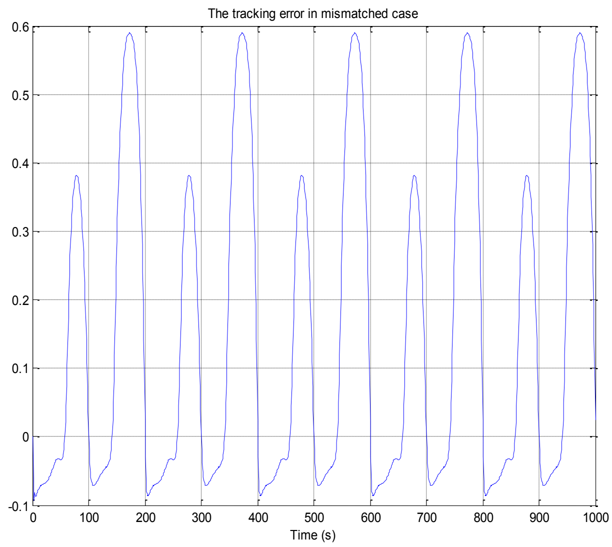

Fig. 8. Tracking error performance (a=1.5).

4. Adaptive Neural Inverse Control

In last years, numerous adaptive control techniques [3] have been proposed to replace the conventional classical methods. 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, however, mathematical modeling of the plant has to be done, which sometimes is 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, in the previous section, we saw that the inverse model controller is sensitive to parameter variations and uncertainties. Thus, to deal with this problem, a new NN block (adaptive mechanism) is added to the scheme to update online the weights of the inverse model controller.

The neural network structure of this non-linear controller is shown in Fig. 9. This scheme uses two sub-networks (NN1 and NN2). The sub-network NN1 is connected in forward path and initially represents the inverse model controller obtained in the first stage (Identification phase). On the other hand, the sub-network NN2, which is connected in feedback path, is used to update the weights when there are uncertainties (unmodeled dynamics) or parameter variations of the plant.

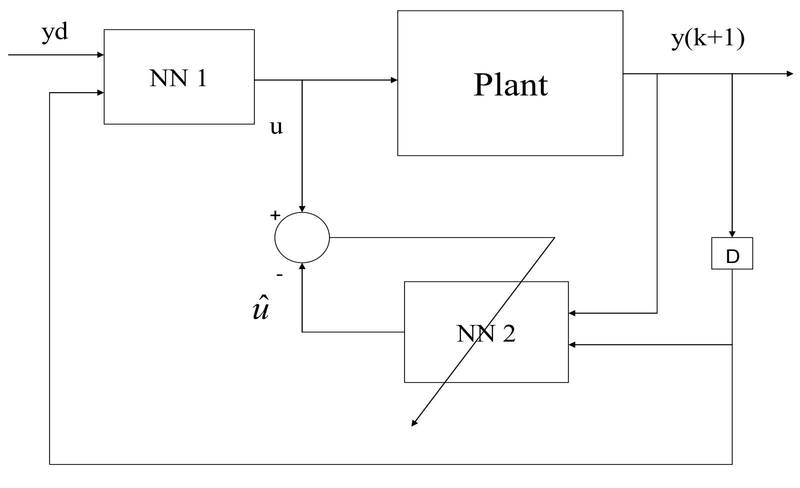

Fig. 9. Adaptive inverse model control scheme.

Thus, the error between the control signal u and the response $\hat{u}$ of the inverse model NN2 is used to update the weights of the inverse model NN2. These new weights are used in the forward inverse model controller NN1.

Simulation example: First the inverse model has been obtained with the parameter value of the plant set to a=1. Note that we will use the same structure of the neural inverse model used in the previous part and is implemented in forward path as a NN1 block and the second block NN2 is added to the structure (Fig. 9). The error $e_u(t) = u(t) - \hat{u}(t)$ is used to update the weights of NN2. These weights are used by the NN1 block. In the test phase, the learning rate is set to 0.05 and the parameter a is set to 2. The simulation results are shown in Figs 10 and 11. The output of the system is close to the desired signal although the parameter value is unknown (since the network has been trained with a=1). Thus, the adaptive neural controller has achieved best tracking error performance (Fig. 11) with regards to inverse model control (Fig. 8).

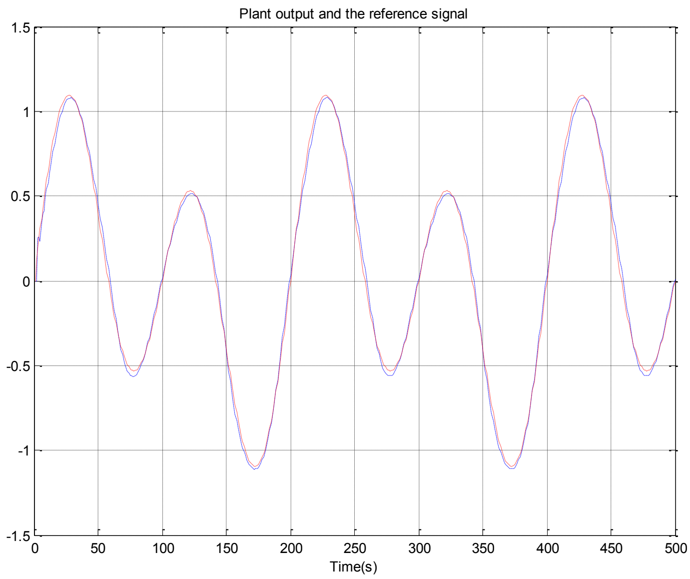

Fig. 10. Performance in mismatched case (a=2).

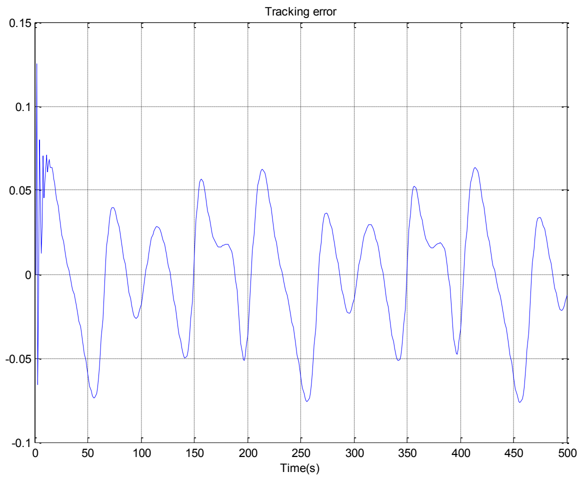

Fig. 11. Tracking error performance (a = 2).

5. Real-time application to temperature system

- **Description of the temperature control system**

  The temperature control system is an important component in many industrial processes. In this work, the used temperature control system, which is from the Leybold-Didactic Company [14], consists of a glass channel that can be viewed as an oven. A picture of this system is shown in Fig. 12.

The system consists of main components:

- ⋆ A power supply (± 15v).
- ⋆ Power amplifier.
- ⋆ Temperature system which includes: Fan, heater (halogen lamp) and flap. All these components are inside the glass channel.

Note that in all experiences, the flap and fan are set to 2 div. without disturbances (matched case).

The sensor module is used to measure the temperature over the range that can transform the measured temperature over the range of 0^oC to 100^oC into the corresponding voltage range of 0V to 10V. Data acquisition board CE122 from ***TQ*** Company [15] is used as an interface between PC and the controlled system.

The main control program, which calculates the control signal to be sent to the temperature system through the interface CE122, is written in C++ language.

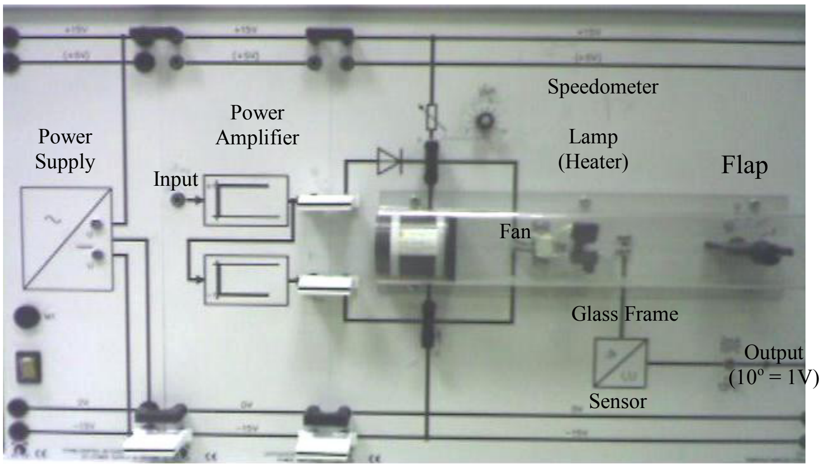

Fig. 12. Picture of the temperature control system.

- **Input-output characteristics of temperature system**

  The neural network is trained to learn the inverse dynamics model of temperature system by using back-propagation method. Since the temperature system is an open loop stable process, a ramp signal between 0V-10V is injected to temperature system via interface-amplifier, with an increment of 0.01V/sample to obtain the corresponding output trajectory over one cycle (1000 samples) as shown in Fig. 13.

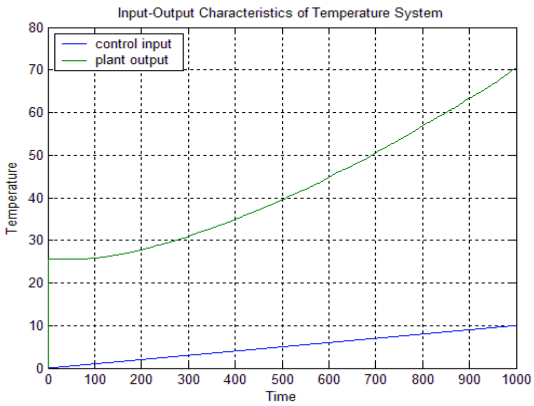

Fig. 13. Input-output characteristics of the process.

- **Inverse identification of temperature system**

  To realize the inverse dynamic, it is necessary to model the plant as a function of measured variables. A judicious choice of variables is essential for the success of the neural network controller design. In general, the neural network model of the inverse dynamics of the plant *I(k)* is a non-linear function *f*, which consists of delayed plant outputs *y* and inputs *u*. As we had no priori knowledge of the temperature model, we selected five cases of input vectors as follows [16]:

  1. Case a: $I(k) = f\big(y(k)\big)$,
  2. Case b: $I(k) = f\big(y(k),\ y(k-1)\big)$,
  3. Case c: $I(k) = f\big(y(k), y(k-1), y(k-2)\big)$,
  4. Case d: $I(k) = f\big(y(k), y(k-1), y(k-2), y(k-3)\big)$,
  5. Case e: $I(k) = f\big(y(k), y(k-1), u(k-1)\big)$.

- **Neural networks structure**

  A two-layered neural network having 12 neurons in the first hidden layer and 6 neurons in the second hidden layer, one output neuron in the output layer and the number of inputs depends on different cases (see above). The hidden neurons have sigmoid functions and the output has a saturating linear function between 0V and 15V. By using the data obtained from the input-output characteristics, all cases have been used to determine the optimal inverse model of the temperature system. Note that training is stopped once the error reaches the value of 10^{-5} or it completes 50,000 epochs. All errors of the five cases are between 10^{-5} and 510^{-5}.

The non-adaptive inverse model control has been used by simulation for the five models cited below. The best tracking performance (with a short steady state error) is obtained by the inverse model defined in case b. This model is used in the adaptive inverse model controller to control the temperature system.

- **Experimental results**

  In this experiment, it is desired that the temperature reach two set points: 40°C for $0 \le k \le 500$ and 60°C for $500 \le k \le 1000$. Thus, we have to inject a reference signal y_{ref} = 4V for $0 \le k \le 500$ and y_{ref} = 6V for $500 \le k \le 1000$. In the matched case (without parameter variations), the experiment results are shown in Figs. 14 and 15.

It is shown that good tracking performances are achieved for two points regulation.

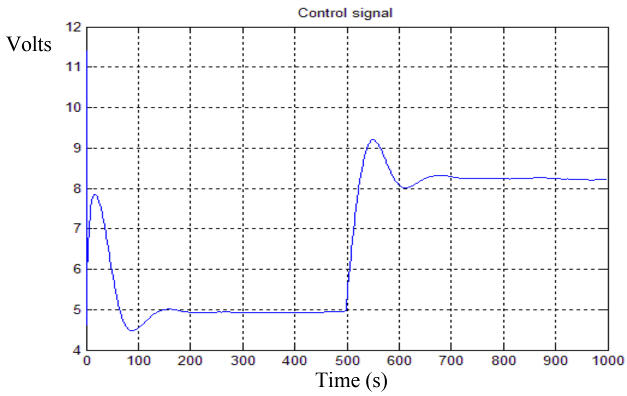

Fig. 14. Control signal.

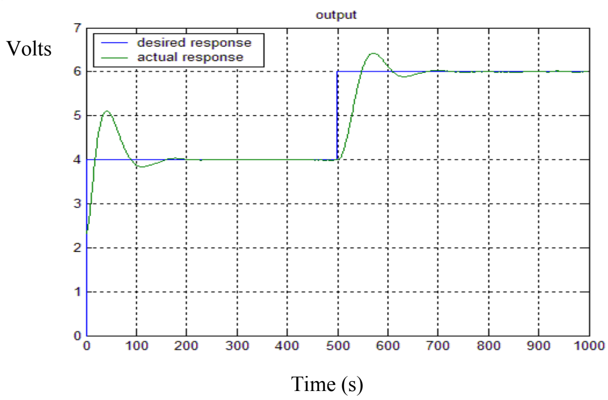

Fig. 15. Tracking performance.

As it is stated in the introduction and in the previous part, the variation of parameters during operation may degrade the tracking performance. In this part of experiment, it was considered that the temperature must reach the value $T = 50^{o}$ C, which is equivalent to y_{ref} = 5V. The flap division is set to 2 and the fan speed division is set to 4. Note that in all previous experiments, both fan and flap divisions are set to 2. To check the robustness of the proposed algorithm, at t=320s the fan speed division is increased to 6 and the flap division to 4. At t=700s both the fan speed division and flap division are decreased to initial values.

The experiment results are illustrated in Figs 16 and 17. Figure 16 depicts the resulted control signal applied to temperature via interface-amplifier. The voltage control signal is within the saturation limits. From Fig. 17, we see that the desired temperature trajectory is well tracked. The variations of the fan speed and flap have no effect on the tracking performances, since they have been compensated. The above results demonstrate that the proposed controller has strong robustness properties in the presence of disturbance and parameter variations.

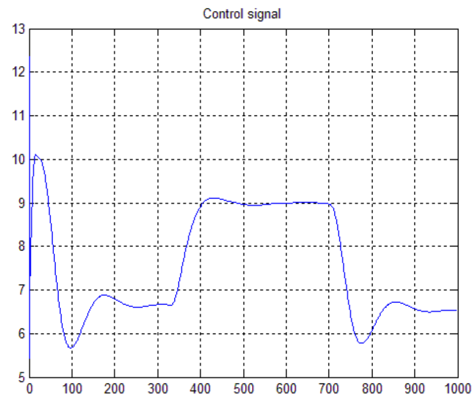

Fig. 16. The applied control signal in the mismatched case.

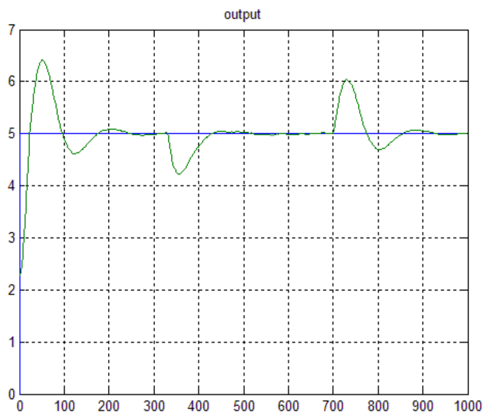

Fig. 17. Tracking performance in the mismatched case.

6. Conclusions

In this paper, an online adaptive control of non-linear system using neural networks is applied to the temperature control system. First, it is shown that the neural inverse model controller is sensitive to parameter variations and/or uncertainties. Indeed, unsatisfactory tracking performances have been obtained. To enhance the robustness of this neural control algorithm with respect to parameter variation and/or uncertainties, a sub-neural inverse model is added to the structure scheme and weights are updated online. Good tracking performances have been obtained in both simulations and experimental application although parameter variations and disturbances are unknown to the neural controller.

For future work, the proposed control algorithm will be used to control more complex non-linear systems.

Acknowledgment: The author would like to thank Eng. Abdullah Saad Alrasheed who carried out the experimental part of this work.

References

[1] Slotine, J. and Li, W. *Applied Non-linear Control*. Englewood Cliffs, N. J.: Prentice-Hall, 1991.
[2] Isidori, A. *Non-linear Control Systems.* 3rd ed., London: Springer-Verlag, 1995.
[3] Åström, K. J. and Wittenmark, B. *Adaptive Control.* 2nd ed., Massachusetts: Addison-Wesley, 1995
[4] Haykin, S. *Neural Networks: A Comprehensive Foundation.* 2nd ed., N.J.: Prentice Hall, 1999.
[5] Principe, J. C.; Euliano, N. R. and Lefebre, W. C. *Neural and Adaptive Systems: Fundamentals through Simulations*. New York: John Wiley, 2000.
[6] Widrow, B. and Stearns, S. D. *Adaptive Signal Processing.* Englewood Cliffs, N. J.: Prentice Hall, 1985.
[7] Psaltis, D.; Sideris, A. and Yamamura, A. "A Multilayered Neural Network Controller." *IEEE Contr. Sys. Mag.,* 10, No. 3 (1988), 44-48.
[8] Khaled, M., Omatu, S. and Yusof, R. "Temperature Regulation with Neural Networks and Alternative Control Schemes." *IEEE Trans. on Neural Networks*, 6, No. 3 (May 1995), 572-582.
[9] Narendra, K. S. and Lewis, F. L. "Special Issue on Neural Network Feedback Control." *Automatica*, 37, No. 8 (August 2001), 1147-1301.
[10] Hoskins, D. A.; Hwang, J. N. and Vagners, J. "Iterative Inversion of Neural Networks and Its Application to Adaptive Control." *IEEE Trans. on Neural Networks*, 3, No. 2 (March 1992), 292-301.
[11] Zhang, J.; Ge, S. S. and Lee, T. H. "Output Feedback Control of a Class of Discrete MIMO Nonlinear System with Triangular form Inputs." *IEEE Transaction on Neural Networks*, 16, No. 6 (2005), 1491-1503.
[12] Pogio, T. and Girosi, F. "Networks for Approximation and Learning." *Proceedings IEEE*, l38 (1990), 1481-1497.
[13] Cabera, J. B. D. and Narendra, K. S. "Issues in the Application of Neural Networks for Tracking Based on Inverse Control." *IEEE Trans. on Automatic Control*, 44, No. 11 (1999), 2007-2027.
[14] www.leybold-didactic.de
[15] www.tq.com

[16] Khalid, M. and Omatu, S. “A Neural Network Controller for a Temperature Control System.” *IEEE Contr. Sys. Mag*, (June 1992), 58-64.
[17] Meirels, M. R. G.; Almeida, P. E. M. and Simoes, M. G. “A Comprehensive Review for Industrial Applicability of Artificial Neural Networks.” *IEEE Trans. on Industrial Electronics*, 50, No. 3 (2003), 585-601.
[18] Narendra K. S. and Parthasarathy, K. “Identification and Control of Dynamical Systems Using Neural Networks.” *IEEE Trans. on Neural Networks*, 1, No. 1 (1990), 4-27.
[19] Yuan, M.; Poo, A. N. and Hong, G. S. “Direct Neural Control System: Nonlinear Extension of Adaptive Control.” *IEEE Proceedings-control Theory and Appl*., 142, No. 6 (1995), 661-667.

التحكم الموائم بشكل مباشر للنظم غير الخطية باستخدام الشبكات العصبية مع تطبيقها على نظام تحكم بالحرارة

رمضان حجار

قسم هندسة الحاسب، كلية علوم الحاسب والمعلومات،
جامعة الملك سعود، الرياض، المملكة العربية السعودية
البريد الإلكتروني: hedjar@ccis.ksu.edu.sa

(قدّم للنشر في ١٨/١١/٢٠٠٦م؛ وقبل للنشر في ١٦/٠٤/٢٠٠٧م)

ملخص البحث: رغم أن النموذج العكسي للمتحكمات العصبية قد أظهر إمكانيات كبيرة في فرع نظم التحكم غير الخطية وغير التقليدية، فإن حساسيتها لتغيرات المعامل و/أو المعاملات غير المؤكدة عادة لا تشجع على تطبيقاتها في الصناعة. وبالفعل، فعندما يكون النظام المتحكم به عرضة لتغيرات المعامل أو المعاملات غير المؤكدة فإن أداء الملاحقة يكون غير مقبولاً. ولتفادي هذه المشكلة تم إضافة نظام عكسي إلى دارة التحكم وإجراء تحديث مباشر للأوزان. وقد أظهرت المحاكاة المنفذة صلابة خوارزمية التحكم. إضافة إلى ذلك فإن النموذج الموائم العصبي للتحكم قد تم تطبيقه على نظام تحكم بالحرارة. وتم الحصول على أداء جيد للملاحقة في مجموعة متنوعة من النقاط تنظيم الحرارة. علماً بأن التغييرات الكبيرة للمعامل والاضطرابات لم تؤد إلى إظهار أية آثار على أداء الملاحقة بسبب التعويض بشكل مباشر.