Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 9 (19): 3513-3521, 2009
ISSN 1812-5654

Adaptive Neuro-PID Controller Design with Application to Nonlinear Water Level in NEKA Power Plant

[1]A. Yazdizadeh, [1]A. Mehrafrooz, [2]J. Jouzdani and [2]R. Barzamini
[1]Faculty of Electrical Engineering, Power and Water University of Technology, Tehran, Iran
[2]Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran

Abstract: In this study, two novel adaptive PID-like controllers capable of controlling multi-variable, non-linear Multi-Input Multiple-Output (MIMO) systems are proposed. The proposed controllers are designed based on neural networks techniques. The learning algorithms are derived according to minimization of the error between the output of the system and the desired output. At first, two kinds of PID-like neural network controller named neural network PID and Neural network PID with internal dynamic feedbacks are introduced. Both of the proposed controllers may be used for controlling multivariable systems. The difference between these two controllers is mainly in the structure of their hidden layers that leads to their different performance. These controllers are applied to different kinds of black box, linear or nonlinear and time variant or time invariant systems. The stability of the proposed algorithm is also proven mathematically. Compared to conventional methods, very good results are achieved using the proposed methods. The simulation results show the quality performance of the proposed adaptive controllers and algorithms. Finally to show the performance of the proposed method, it is applied to the water level of tanks in water refinement process in NEKA Power Plant which is generally a very nonlinear system. Simulation results in this study show good performance of the proposed adaptive controllers.

Key words: Adaptive PID controller, neural networks, multi-variable systems, fluid level control, nonlinear control, multi input-multi output systems

INTRODUCTION

Nowadays, a wide variety of advanced control methods is used in industries in order to control important process variables. Many engineers have applied advanced control methods (El-Kouatly and Salman, 2008; Farivar *et al.*, 2009; Otero, 2004) and prefer these methods to conventional methods (Astrom and Hagglund, 1995; Astrom and Wittenmark, 1989). Different people may have different understandings of advanced control methods according to their uses and applications.

Generally, selection of a control method depends on where it is to be used. Nevertheless, the objective is always reaching to the desired conditions. Up to late 90's, monitoring in industrial systems was done manually. Advances in technology, make these systems more and more automated.

Nowadays, PID controllers are used in more than 80% of feedback control systems in different industries (Cheng-Ching, 1999; Bao *et al.*, 1999; Shi and Chen, 2001; Shu and Pi, 2005). Astrom and Hagglund (1995) has shown how to adjust the parameters according to frequency responses and experimental rules. In case of process parameters change, Astrom and Wittenmark in their novel work (Astrom and Wittenmark, 1989) show adaptive methods for adjusting the parameters of a controller. The adaptive methods for controlling such systems are growing rapidly (Barzamini *et al.*, 2009). It can be proved that a PI controller can be used for controlling a first order system but for controlling a second order system a PID controller is needed. In practice, the characteristics of the real systems are non-linear and time-varying, so that linear model of the system is almost useless. In order to adjust the parameters of the controller in such systems, some solutions are proposed. One of these methods is to store many parameters in the system and to select a number of them according to the circumstances. In other words, at the time when it is detected that the system is moving from a region to another, system parameters are switched to new ones; but in general, this method cannot be applied to continuous and model-free systems.

On the other hand in adaptive methods, parameters of the model are determined frequently according to the

Corresponding Author: Alireza Yazdizadeh, Power and Water University of Technology, P.O. Box 16765-1719, Tehran, Iran
Tel: +98 21 22832069 Fax: +98 21 77310425

process situation. These parameters contain the characteristics of the system at each moment. These parameters are used for calculating and adjusting the parameters of the controller and the controller adjustments change frequently as the process model change.

MATERIALS AND METHODS

In this study, which is the result of extensive research in the field of neural network and its application as the controller on industrial plants during 2005 to 2008 in Power and Water University of Technology, we aimed to propose a new neural-network based adaptive controller. The proposed controller is capable of solving the problems and shortages of the other similar methods.

Fundamentals of neural networks-based adaptive controller design: Due to the nonlinearity involved in the system under investigation, it is preferred to propose a method that dose not need the linear model of the system. Therefore, the system is considered as a black box, where only the inputs and outputs are assumed to be available at all time instants. Having considered the above assumption, the introduced method is supposed to be applicable in many industrial systems.

The goal of this study is to substitute the conventional PID controller with a neural-network-based controller that is capable of dealing with nonlinear plants. A neural network-based controller dose not needs a priori information about the dynamic of the system and only input-output data would be enough for running the control scheme (Chen and Cheng, 1996; Chen *et al*., 1995).

Introduction to general PID neural network: Several static neural network based identifiers and controllers have been introduced in the literature for identification and control of dynamic systems (Chen *et al*., 1990; Shu and Guo, 2004). Nowadays, the use of neuron with dynamic structure is being rapidly increased. By using dynamic elements embedded in a static neuron structure, like adaptive delays (Yazdizadeh and Khorasani, 2000, 2002), embedded adaptive filter at the out put of each neuron (Yazdizadeh and Khorasani, 1998; Cong and Li, 2005), the desired dynamics can be created.

Structure of the dynamic neuron in order to create dynamics of a PID controller is changed and proposed. Each neuron includes some inputs and outputs (Fig. 1). The governing equation of the neuron that relates the inputs to the outputs are given. Proportional, Integral and Differential type neurons which are given are basically different in their governing equations.

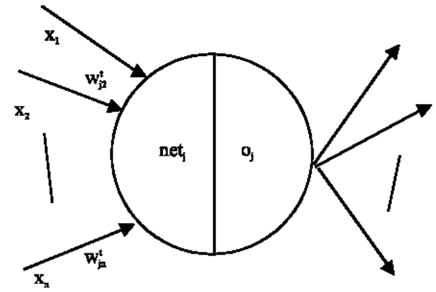

Fig. 1: Schematic of a neuron

Structure of dynamic neurons: The general structure of three different types of the neurons is shown in Fig. 1. The governing equation of each neuron is presented as follows.

Proportional type neuron: This simple neuron is characterized by its activation function as $o_j(t) = net_j(t)$ for continuous time or as $o_j(k) = net_j(k)$ for discrete time where its input is given by

$$net_j(k) = \sum_{i=1}^{n} w_{ji} x_i(k)$$

This neuron has a linear activation function and acts as an adaptive weighted adder. In fact, the P-Type neuron is an adaptive gain unit.

Integral type neuron: The I-Type neuron acts as an integrator and the output of the neuron is, in fact, the weighted integral of the input. The relationship between the input and the output in continuous and discrete form are presented by Eq. 1 and 2, respectively.

$$o_j(t) = \int_0^t net_j(t')dt' \tag{1}$$

$$o_j(k) = \sum_{k'=1}^{k} net_j(k') = \sum_{k'=1}^{k-1} net_j(k') + net_j(k) = o_j(k-1) + o_j(k) \tag{2}$$

Differential type neuron: The D-type neuron acts as a derivative operator. The relationship between the input and output of this neuron is presented by Eq. 3 and 4, respectively:

$$o_j(t) = \frac{d}{dt} net_j(t) \tag{3}$$

$$o_j(k) = \frac{net_j(k) - net_j(k-1)}{\underbrace{k-(k-1)}_{1}} = net_j(k) - net_j(k-1) \tag{4}$$

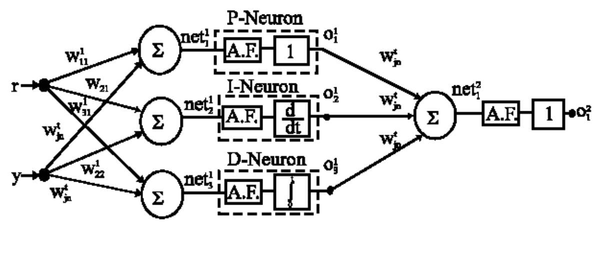

Fig. 2: Structure of the PID neural network controller

Proportional integral differential type neural network controller: The general structure of this controller is shown in Fig. 2. The network has two separate layers where the hidden layer consists of all three types of neurons (P, I and D). The output layer consists of a P-type neuron and computes the weighted sum of the outputs of the hidden layer. In order to show the input-output representation of the network, the weights are initially set as in the following equations Eq. 5, 6:

$$w^1_{j2} = -1, \quad w^1_{j1} = 1 \quad j = 1,2,3 \tag{5}$$

$$\begin{aligned}
&net^1_1 = w_{11}x_1 + w_{12}x_2 = r - y = e, \quad w^2_{11} = K_P, \quad o^1_1 = net^1_1 = e \\
&net^1_2 = w_{21}x_1 + w_{22}x_2 = r - y = e, \quad w^2_{21} = K_I, \quad o^1_2 = \int_0^t net^1_2 dt = \int_0^t edt \\
&net^1_3 = w_{31}x_1 + w_{32}x_2 = r - y = e, \quad w^2_{31} = K_D, \quad o^1_3 = \frac{dnet^1_3}{dt} = \frac{de}{dt} \\
&o^2_1 = net^2_1 = \sum_{j=1}^{3} w^2_{j1}.o^1_j = w^2_{11}o^1_1 + w^2_{21}o^1_2 + w^2_{31}o^1_3 = K_P e + K_I \int_0^t edt + K_D \frac{de}{dt}
\end{aligned} \tag{6}$$

Generalization of the PID neural network controller: Here, a generalized PID neural network controller consisting of several previously introduced PID neural networks which can be used as a PID controller for Multi-Input Multi-Output systems is introduced (Mehrafrooz and Yazdizadeh, 2007). Figure 3 shows the structure of the controller and the multi-variable system as a feedback loop.

Regardless of the type of the multi-variable system and due to the importance of the proposed controller, we take a closer look at the neuro-controller. Figure 4 shows the structure of the MIMO type of the proposed controller. In Fig. 4, r and y represent the desired values and the output respectively and the second layer of the neural network generates the control command. It can be easily concluded that:

- The network can operate for systems with n inputs and n outputs (the systems in which the number of inputs and the number of outputs are equal)
- The number of neurons in the output layer represents the number of inputs of the system and can be changed depending on the application

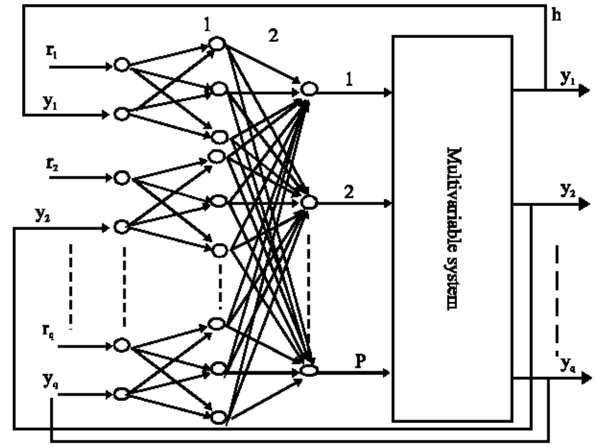

Fig. 3: Generalized PID neural network controller for MIMO systems

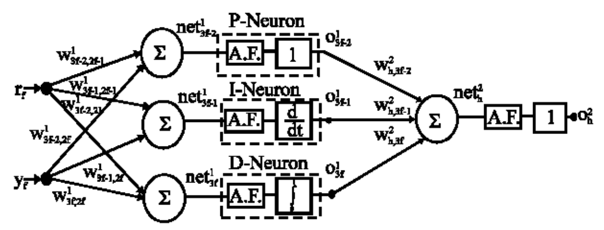

Fig. 4: Structure of the PID neural controller for MIMO systems with p inputs and q outputs

- Therefore, it may be inferred that the network is capable of controlling systems with p inputs and q outputs and in fact in this method there is no obligation on the equivalency of the number of inputs and outputs

Dynamic neural network controller with local feedback: The structure introduced here, like one is discussed earlier, has the structure of a PID controller. In this section a neural network with only 3 neurons in the hidden layer is proposed. Also, in this controller, which we call it as Dynamic Neural Network Controller with Local Feedbacks, internal feedbacks are considered as output and activation feedbacks (Fig. 5).

By using the above structure, the number of neurons is decreased that in turn leads to a faster adjusting algorithm. The hidden layer of the network can act as a P, PI, PD and PID controller. The proposed controller like generalized neural network PID controller has the capability of automatic adjustment of weights by using adaptive algorithms and can be used for controlling non-linear systems. Having combined the simple structure of the PID controllers with the automatic learning capability

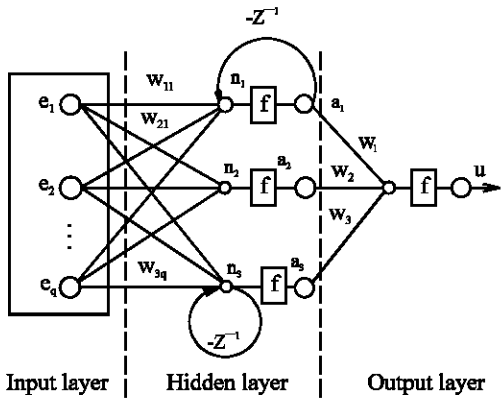

Fig. 5: Dynamic neural network controller with local feedbacks

of the neural networks in the proposed method, a very efficient structure is achieved.

Structure of a dynamic neural network controller with local feedbacks: As shown in Fig. 5, the structure consists of a hidden layer and an output layer. In the hidden layer there are 3 neurons called as a_1, a_2 and a_3. The output layer generates the output signal by using a single neuron. The activation functions in input and output layers are linear. The outputs of the neurons of the first layer are sent to the hidden layer neurons. There are 3q weights: w_{i1}, w_{i2} and w_{i3}, $1 \le i \le q$. Neuron a_1 in the hidden layer is an integral neuron whose feedback causes a delay. Neuron a_3 in the same layer is a derivative neuron that sends the weighted values to a_3 with a delay feedback. Neuron a_2 in the hidden layer with a linear activation function and without any feedback acts as the proportional part in the PID controller. Having considered the above explanation and according to Fig. 5, the outputs in n_1, n_2 and n_3 are as follows:

$$a_1(k) = f(\sum_{j=1}^{q} w_{1j}(k)e_j(k) + a_1(k-1)) = \sum_{j=1}^{q} w_{1j}(k)e_j(k) + a_1(k-1) \quad (7)$$

$$a_2(k) = f(\sum_{j=1}^{q} w_{2j}(k)e_j(k)) = f(\sum_{j=1}^{q} w_{2j}(k)e_j(k)) \quad (8)$$

$$a_3(k) = f(\sum_{j=1}^{q} w_{3j}(k)e_j(k) - \sum_{j=1}^{q} w_{3j}(k-1)e_j(k-1)) = \sum_{j=1}^{q} w_{3j}(k)e_j(k) - \sum_{j=1}^{q} w_{3j}(k-1)e_j(k-1) \quad (9)$$

It can be seen that in contrary with the former neural network, discussed in the beginning of the study, the activation functions are linear and origin-passing and the hidden layer neurons may be distinguished based on their feedbacks. The output layer neuron generates the control command as:

$$U(k) = \sum_{i=1}^{3} w_i(k)a_i(k) \quad (10)$$

where,

$$a(k-1) = a(k).z^{-1}, e(k-1) = e(k).z^{-1} \quad (11)$$

Hence the equations may be re-written as:

$$a_1(k) = \frac{\sum_{j=1}^{q} w_{1j}(k)e_j(k)}{1-z^{-1}} \quad (12)$$

where, $1\text{-}z^{-1}$ as the denumerator shows the integration behaviour; that's the reason for calling the neuron as integral neuron. Also we have:

$$a_3(k) = (\sum_{j=1}^{q} w_{1j}(k)e_j(k))(1-z^{-1}) \quad (13)$$

where, $1\text{-}z^{-1}$ shows the derivative behavior. That's the reason for calling the neuron as *derivative neuron*. By using a_1, a_2 and a_3 neurons in the hidden layer the input-output representation of a PID controller in its multi-input multi-output case is achieved.

Weight adjustment algorithm for the dynamic neural network controller with local feedbacks: Like all industrial controllers, the goal of using a Dynamic Neural Network Controller with Local Feedbacks is to minimize the difference between the outputs and the desired trajectory. The closed loop structure of this controller is depicted in Fig. 6. The error for each one of the outputs of the system is:

$$e_j(k) = r_j(k) - y_j(k), \quad j = 1, 2, \ldots, q \quad (14)$$

Assuming r_j as the desired output, the cost function for each output is defined as follows:

$$J_j(k) = \frac{1}{2}(r_j(k) - y_j(k))^2 \quad (15)$$

And the total cost function which presents the error in each sampling is:

$$J(k) = \sum_{j=1}^{q} J_j(k) \quad (16)$$

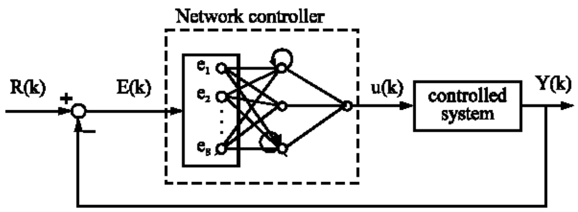

Fig. 6: The closed loop structure of dynamic neural network controller with local feedbacks

The gradient of the cost function with respect to the weights is calculated by using the following equation:

$$\frac{\partial j(k)}{\partial w_{ij}(k-1)} = \sum_{j=1}^{q}\left(\frac{\partial j_j(k)}{\partial y_j(k)}.\frac{\partial y_j(k)}{\partial u(k-1)}\right)\frac{\partial u(k-1)}{\partial w_{ij}(k-1)} = -\sum_{j=1}^{q}\{e_j(k)\operatorname{sgn}\left(\frac{y_j(k)-y_j(k-1)}{u(k-1)-u(k-2)}\right)\}e_j(k-1) \quad (17)$$

Since, there is no explicit relationship between the input and output of the system, in the above gradient the following approximation is used:

$$\frac{\partial y_j(k)}{\partial u(k-1)} \cong \frac{\Delta y_j(k)}{\Delta u(k-1)} = \frac{y_j(k)-y_j(k-1)}{u(k-1)-u(k-2)} \quad (18)$$

Moreover, it is prefered to use the sign of the above expression instead of the value itself in weight adjustemnt algorithm, since the effects can be compensated for in the weight adjustment rule. This prevents the algorithm from getting stuck in the steps in which the above value is small and accelerates and eases the calculations of the learning of the weights.

$$\Delta w_{ij}(k-1) = -\eta_{ij}\frac{\partial J_j(k)}{\partial W_{ij}(k-1)} = \eta_{ij}\sum_{j=1}^{q}\left\{e_j(k)\operatorname{sgn}\frac{\Delta y_j(k)}{\Delta u(k-1)}\right\}e_j(k-1), \quad i=1,2,3, \quad j=1,2,\ldots,q \quad (19)$$

The same procedure may be used for adjusting the parameters of the network (Mehrafrooz and Yazdizadeh, 2007).

Stability of the closed loop system: Among many research works in the field of application of neural network in control systems, only few consider the stability issue. Hourfar and Salahshoor (2009) have introduced a novel technique for controlling a continuous stirred reactor based on the very well known feedback linearization technique. In their proposed method the plant is modeled by an adaptive neural network. Mehrafrooz and Yazdizadeh (2007) in their research work show the stability of such a system for the first time. The use of Lypunov theorem for analysis and proof of the stability of nonlinear systems is a popular and well established method (Yazdizadeh *et al*., 2009). In order to prove the stability of the proposed method, the following Lyapunov function is defined:

$$V(k) = \frac{1}{2}\sum_{j=1}^{q} e_j^2(k) \quad (20)$$

and therefore for a sampling time k we have:

$$\Delta V(k) = V(k+1) - V(k) = \frac{1}{2}\sum_{j=1}^{q}\left[e_j^2(k+1) - e_j^2(k)\right] \quad (21)$$

and according to:

$$\frac{\Delta e_j(k)}{\Delta w_{ij}(k)} = \left(\frac{\partial e_j(k)}{\partial w_{ij}(k)}\right)^T \quad (22)$$

we have:

$$e_j(k+1) = e_j(k) + \Delta e_j(k) = e_j(k) + \left(\frac{\partial e_j(k)}{\partial w_{ij}(k)}\right)^T \times \Delta w_{ij}(k) \quad (23)$$

On the other hand, we have:

$$\Delta w_{ij}(k) = -\eta_{ij}\frac{\partial J(k)}{\partial w_{ij}(k)} = -\eta_{ij}\sum_{j=1}^{q} e_j(k)\frac{\partial e_j(k)}{\partial w_{ij}(k)}, \quad \Delta e_j(k) = -\eta_{ij}\sum_{j=1}^{q}[e_j(k).\frac{\partial e_j(k)}{\partial w_{ij}(k)}](\frac{\partial e_j(k)}{\partial w_{ij}(k)}) \quad (24)$$

By defining:

$$\frac{\partial e_j(k)}{\partial w_{ij}(k)} = e_{jw} \quad (25)$$

we have:

$$\Delta V(k) = \sum_{j=1}^{q}\left\{\eta\sum_{j=1}^{q}[e_j(k).e_{jw}].e_{jw}[-e_j(k) + \frac{1}{2}\eta_{ij}\sum_{j=1}^{q}[e_j(k).e_{jw}]e_{jw}]\right\} = \sum_{j=1}^{q}\left\{\eta(\sum_{j=1}^{q}[e_j(k).e_{jw}])^2\left(\frac{-e_j(k)e_{jw}}{\sum_{j=1}^{q}[e_j(k)e_{jw}]} + \frac{1}{2}\eta_{ij}e_{jw}^2\right)\right\} \quad (26)$$

It is now clear that since we have

$$\sum_{j=1}^{q}\left([e_j(k).e_{jw}]\right)^2 > 0$$

therefore, V(k)<0 and the system is stable provided that:

$$\eta_{lij}(-\lambda_j + \frac{1}{2}\eta_{lij}e_{jw}^2) < 0 \quad (27)$$

where:

$$\lambda_j = \frac{e_j(k)e_{jw}}{\sum_{j=1}^{q}[e_j(k)e_{jw}]} \quad (28)$$

in which λ_j can be negative or positive. Having assumed $\lambda_j>0$, the stability condition is:

$$0 < \eta_{lij} < \frac{2\lambda}{e_{jw}^2} \quad (29)$$

and when $\lambda_j<0$ the condition will be:

$$\frac{2|\lambda|}{e_{jw}^2} < \eta_{lij} < 0 \quad (30)$$

On the other hand, we have:

$$\begin{aligned}\frac{\partial e_j(k)}{\partial w_{ij}(k)} &= e_{jw} = -\frac{\partial y_j(k)}{\partial u(k)}.\frac{\partial u(k)}{\partial w_{ij}(k)} \\ &= \frac{-1}{\frac{\partial u(k)}{\partial y_j(k)}}.e_j(k) = \frac{1}{\frac{\partial u(k)}{\partial e_j(k)}}.e_j(k) = \frac{e_j(k)}{\sum_{i=1}^{3}w_{ij}}\end{aligned} \quad (31)$$

By defining:

$$e_{max}(k) = \max\left(\frac{e_j^2(k)}{\sum_{i=1}^{3}w_{ij}}\right) w_{min}(k) = \min\left(|w_{ij}(k)|\right) \quad (32)$$

and according to the equation:

$$e_{jw} = \frac{e_j(k)}{\sum_{i=1}^{3}w_{ij}} \quad (33)$$

and Eq. 37, we have:

$$\frac{2|\lambda|}{e_{jw}^2} = \frac{\left|\sum_{i=1}^{3}w_{ij}(k)\right|}{\sum_{j=1}^{q}\frac{e_j(k)}{\sum_{i=1}^{3}w_{ij}}} < \frac{w_{min}(k)}{e_{max}(k)} \quad (34)$$

and the learning rate η, should be in the following interval:

$$-\frac{w_{min}(k)}{e_{max}(k)} < \eta < \frac{w_{min}(k)}{e_{max}(k)} \quad (35)$$

RESULTS AND DISCUSSION

Conventional PID controllers are widely used in industrial plants. It is very hard to convince engineers in industry to substitute the conventional controllers by modern and more advanced techniques like Neuro-PID controllers instead. Toward this, we first selected one of the challenging issues in the steam power plant. In order to show the performance of the proposed method in a real life physical system, we extracted almost the exact model of the system under investigation. In order to compare the results of the proposed method with the conventional method, we applied a conventional PID controller to the system as well. As it is shown here, due to the nonlinear behaviour of the proposed neural network based method and also due to adaptive characteristic of the proposed method, the time domain response of the system is improved in all important aspects.

Application of the proposed controllers to neka power plant model: NEKA Power Plant is a large steam power plant in Northern Province Mazandaran in Iran. There is a number of processes and control loop in large scale system like a steam power plant. Among them, one may refer to polishing plant that is responsible for chemical process for water and steam closed circle in the power plant. Although, some other important variables are controlled in this section, tank level control is one of the challenging issues (El-Kouatly and Salman, 2008; Al-Gallaf, 2002).

As shown in Fig. 7, the system consists of a tank and a pump which pumps the fluid into the tank from above. This method of filling the tank is called as non-gravitational filling. The pump has a variable speed rate and its speed rate depends on the voltage it receives. The term bV represents the input flow to the tank where b is a constant that shows characteristic of the pump and V is the voltage applied to the pump.

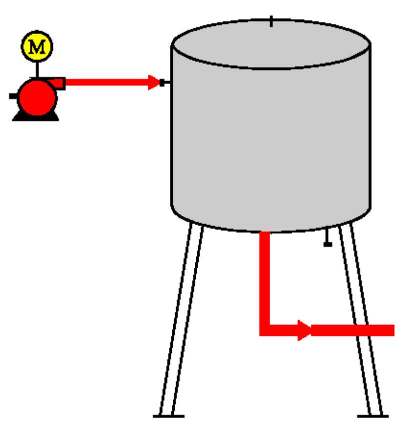

Fig. 7: Tank and the Pump in water level control system

The rate at which the fluid exits the tank, due to the gravity force, is proportional to the square root of the height of the fluid and is represented by $a\sqrt{h}$ in which a is a constant coefficient and depends on the characteristics of the output pipe. The change of the tank fluid level depends on the difference between the input and output flow, therefore:

$$\frac{dvol}{dt} = bV - a\sqrt{h} \tag{36}$$

where, vol represent the volume of the fluid in the tank. On the other hand:

$$\frac{dvol}{dt} = A\frac{dh}{dt} \tag{37}$$

where A is the area of the base of the tank. Therefore:

$$\frac{dh}{dt} = \frac{bV}{A} - \frac{a\sqrt{h}}{A} \tag{38}$$

It can be seen in Eq. 37 that the term $\sqrt{h}$ makes the system non-linear:

Implementation of the proposed controllers for NEKA power plant water tanks: The tanks have a circular base with a radius of 11 m and are 11 m high. Each tank is filled by using a control valve and the water exits each tank at the rate of 55 m^3 h^{-1} by a pump (Mehrafrooz and Yazdizadeh, 2007). To apply the controllers, it is assumed that the desired level of the fluid in the tank is 5 m. The pump starts to work at t_0. The equation of the non-linear state for the system is given by:

$$\frac{dh}{dt} = \frac{b.c}{A} - \left(\frac{a\sqrt{h} + F}{A}\right)u(t - t_0) \tag{39}$$

where, b depends on the input flow and c is the control signal applied to the control valve with a magnitude of 4 to 20 mA and A, h, a and F are the area of the base of the tank, height of the tank, characteristics coefficient of the output pipe and the output flow that is pumped by the pump, respectively. It is clear that at t_0 the pump and the gravitational force causes the fluid level to drop. These parameters are defined as follows:

$$\begin{aligned} &A = \pi * (5.5)^2 = 95\ m^2 \\ &F = 55\ m^3\ h^{-1} = 0.0153\ m^3\ sec^{-1} \\ &a = 9\ m^{2.5}\ sec^{-1} \\ &b = 2\ m^3\ A.s \end{aligned} \tag{40}$$

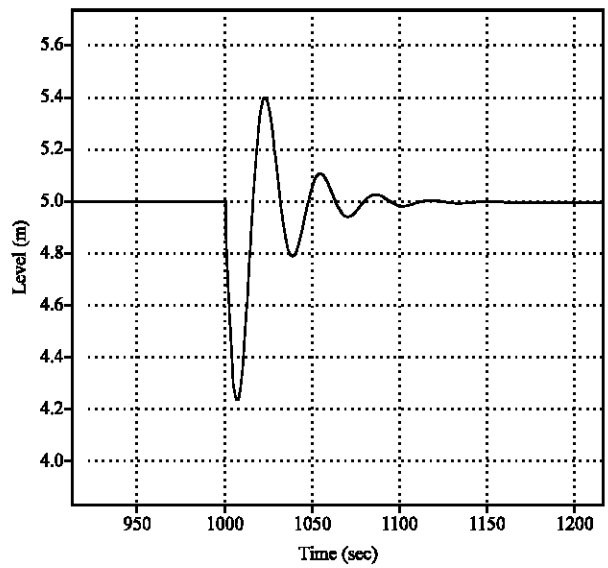

Fig. 8: Water level

At first, the system is controlled by a conventional PID controller. Proportional coefficients, derivation and the integration for this controller are 2, 3 and 0.2, respectively based on experiments.

Figure 8 shows the water level of the tank. As depicted in Fig. 8, when the pump starts, the water level drops about 8 cm that represents a significant amount of water according to the large volume of the tank.

To compare the results of the Generalized Neural Network Controller with the results of the conventional PID controller, the weight adjustment algorithm is run 1000 iterations to achieve the following values:

$$\begin{aligned} &W(1) = 4.09 \\ &W(2) = 3.38 \\ &W(3) = 2.54 \end{aligned} \tag{41}$$

The above weights are proportional, integral and derivative coefficients of the PID controller, respectively. It should be mentioned that the initial values are the same initial values used in the conventional controller.

The out put signal, namely, the water level of the tank is shown in Fig. 9.

Compared to the traditional method, the overshoot and undershoot are less. In other words the overshoot decreases from 1bout 40 to 30 cm and the undershoot decreases from 80 to 40 cm. In addition, rest time of the fluid is reduced from 150 to 100 sec.

In industrial plants the control signal is very important. It is shown in Fig. 10. From industrial point of view, this control command is acceptable to be applied to the actuator.

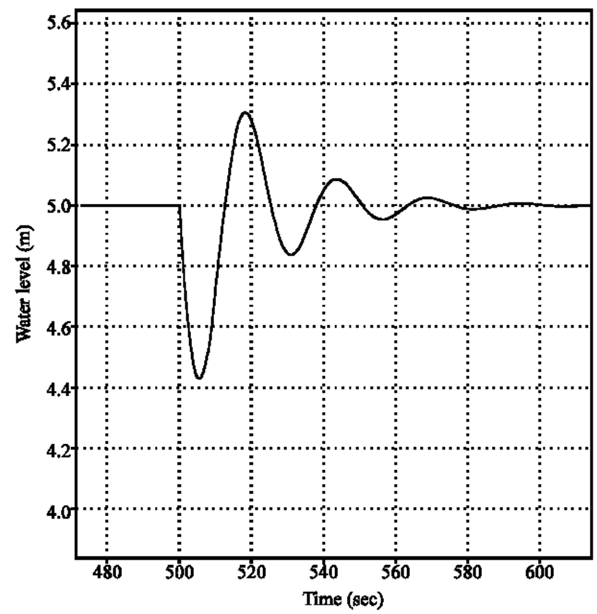

Fig. 9: The output signal curve of the generalized PID controller

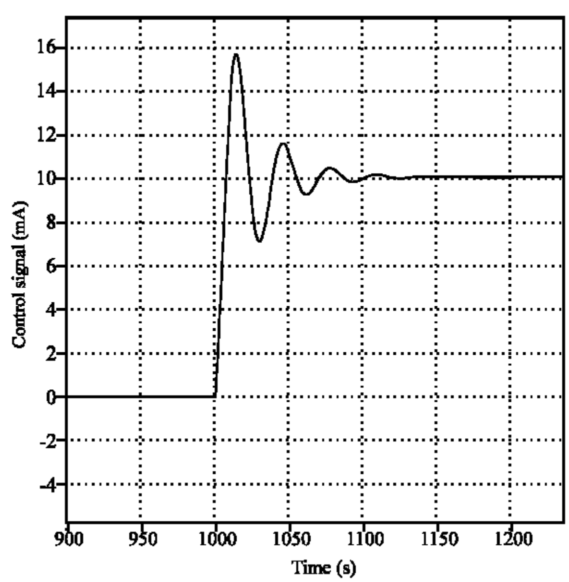

Fig. 10: The control valve signal

Figure 11 shows the error which is the difference between the output signal (real fluid level) and the desired output (5 m). The magnitude of the error decreases as the number of iterations in weight adjustment algorithm increases. As shown in Fig. 4, the error reduction rate slows down as the number of iteration reaches to 500. Therefore, in case that we need more speed for the system we may decrease the number of iteration to 500.

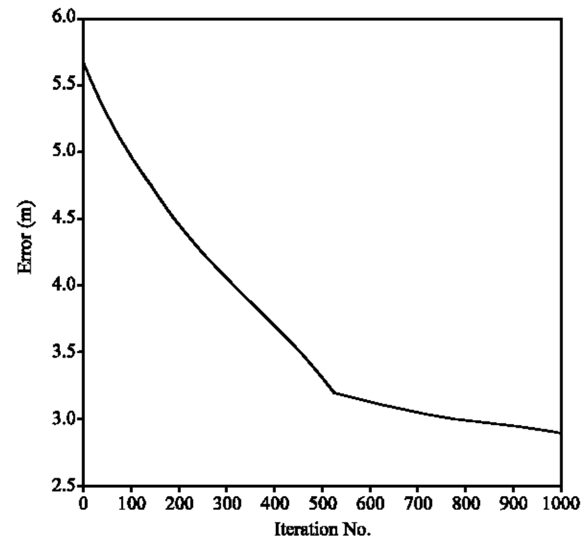

Fig. 11: Errors for different number of iterations

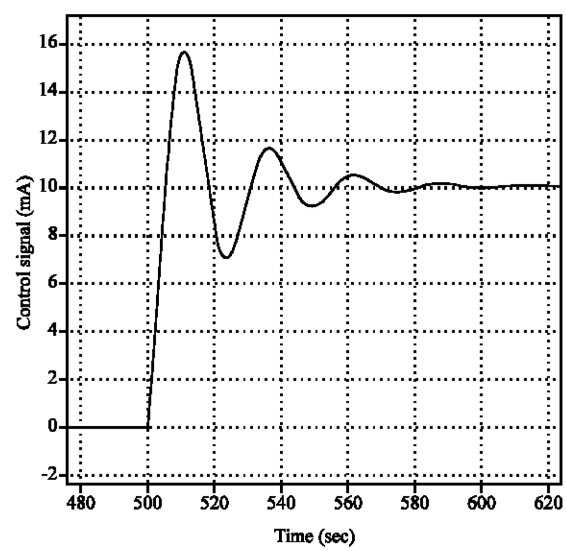

Fig. 12: Control signal curve for the neural network PID controller

In order to show the performance of the proposed algorithm, even with only 500 iteration, the system is simulated under this condition. The curve for the control signal applied to the control valve is depicted in Fig. 12. It can be seen that the control valve is completely opened (20 mA) about 10 sec after the pump starts working and finally reaches to more steady state condition at 12 mA(10+4). From industrial point of view, this control command is a very usual type of command which is applied to the actuator.

CONCLUSION

In this study, a neural network based PID controller is proposed. The input-output representation of the network matches the PID governing equation. The proposed method is generalized to multi input-multi output case which is not applicable in conventional PID controller. The stability of the proposed method is shown by using Lyapunov technique. To show the effectiveness of the proposed method, it is applied to an industrial plant, namely water level control in a tank in NEKA steam power plant. The simulation results show very good performance of the controller in the sense that it is more accurate and due to the system performance enhancement control valve life time is increased and energy consumption is reduced.

REFERENCES

Al-Gallaf, E.A., 2002. Artificial neural network based nonlinear model predictive control strategy. Inform. Technol. J., 1: 173-179.

Astrom, K.J. and B. Wittenmark, 1989. Adaptive Control Stability Convergence and Robustness, Prentice Hall, Englewood Cliffs, New Jersey.

Astrom, K.J. and T. Hagglund, 1995. PID Controllers: Theory, Design and Tuning. 2nd Edn., Instrument Society of America, North Carolina.

Bao, J., J.F. Forbes and P.J. McLellan, 1999. Robust Multi-loop PID controller design a successive semi definite programming approach. Ind. Eng. Chem. Res., 38: 3407-3419.

Barzamini, R., A.R. Yazdizadeh, H.A. Talebi and H. Eliasi, 2009. Adaptive control of a double-electromagnet suspension system. J. Applied Sci., 9: 1201-1214.

Chen, S., S.A. Billings and P.M. Grant, 1990. Nonlinear system identification using neural networks. Int. J. Control, 49: 1013-1032.

Chen, B.S., Y.M. Chiang and C.H. Lee, 1995. A genetic approach to mixed h2/h optimal pid control. IEEE Control Syst. Magazine, 15: 51-56.

Chen, C.L. and F.Y. Chang, 1996. Design and analysis of neural fuzzy variable structural pid control systems. Proceedings of the IEE Control Theory and Applications, March 1996, Taiwan Univ., Taipei, pp: 200-208.

Cheng-Ching, Y., 1999. Auto Tuning of PID Controllers. Springer-Verlag, London.

Cong, S. and B. Li, 2005. A Novel PID-Like Neural Network Controller. Departmentof Automation University, P.R. China.

El-Kouatly, R. and G.A. Salman, 2008. A radial basis function with multiple input and multiple output neural network to control a non-linear plant of unknown dynamics. Inform. Technol. J., 7: 430-439.

Farivar, F., M.A. Shoorehdeli, M. Teshnehlab and M.A. Nekoui, 2009. Hybrid control of flexible manipulator. J. Applied Sci., 9: 639-650.

Hourfar, F. and K. Salahshoor, 2009. Adaptive feedback linearization control of nonlinear processes using neural network based approaches. J. Applied Sci., 9: 3219-3235.

Mehrafrooz, A. and A. Yazdizadeh, 2007. Adaptive neuro-pid controller design with application to nonlinear water level in neka power plant. M.Sc. Thesis, Power nad Water University of Technology.

Otero, F., 2004. Advanced control techniques. Cornerstone Controls.

Shi, C.T. and B. Chen, 2001. Multi objective PID control design in uncertain robotic systems using neural network elimination scheme. IEEE Trans. Syst. Human Cybernetics, Part A Syst. Humans, 31: 637-657.

Shu, H. and X. Guo, 2004. Decoupling control of multivariable time-varying systems based on PID neural network. Asian Control Conf., 1: 682-685.

Shu, H. and Y. Pi, 2005. Decoupled temperature control system based on PID neural network. Proceedings of the ACSE 05 Conference, Dec. 19-21, CICC, Cairo, Egypt, pp: 1-5.

Yazdizadeh, A. and K. Khorasani, 1998. Identification of nonlinear systems using embedded dynamic neural networks. Proceedings of the IEEE International Conference on Neural Networks, (ICNN'1998), Anchorage, Alaska, USA., pp: 378-383.

Yazdizadeh, A. and K. Khorasani, 2000. Identification of a two link flexible manipulator using adaptive time delay neural network. IEEE Trans. Man Cybernet., 30: 165-171.

Yazdizadeh, A. and K. Khorasani, 2002. Adaptive time delay neural network structure for nonlinear system identification. Neurocomputing, 47: 207-240.

Yazdizadeh, A., S.M. Noorbakhsh and R. Barzamini, 2009. A new lyapunov-based design scheme for adaptive friction compensation. J. Applied Sci., 9: 1668-1676.