Adaptive Control Based on Neural Network System Identification

Hassan E. A. Ibrahim
Dept. of Electrical and Computer Control Engineering
Arab Academy for Science and Technology, Cairo, Egypt.
e-mail: hibrahim_eg@yahoo.com

Abstract:- In adaptive control and system identification the self tuning regulator has wide range of applications. Neural network and artificial intelligence have big role in this area. This paper presents adaptive neural network control based on self tuning regulator (STR) scheme. The paper presents neural network block for on line system identification and discrete PID block controller. Analysis for the whole scheme is presented and simulated for different systems. Adequate desired performance is obtained by comparison with the nominal methods for using self tuning regulator.

Key-Words:-
Adaptive Control, Self Tuning Regulator, System Identification, Neural Network, Neuro Control

1. Introduction

The purpose of adaptive controllers is to adapt control law parameters of control law to the changes of the controlled system. Many types of adaptive controllers are known. In [1] the adaptive self-tuning LQ controller is described. The design and implementation of adaptive linear optimal controller (LQ) based on a pseudo-space model into PLC. For identification of the controlled system an algorithm based on artificial neural network is used. In [2] an adaptive neural network control scheme for thermal power system is described. The online tuning algorithm and neural network architecture are described. The performance of the controller is illustrated via simulation for different changes in process parameters. Performance of neural network controller is compared with conventional proportional-integral control scheme for frequency control in thermal power systems. In [3] neural network adaptive force controller is proposed for a hydraulic system. The dynamic model of this system is highly non-linear and very complex to obtain. The neural network parameters are updated online according to an adaptation algorithm obtained via stability analysis. The performance of the proposed neural network controller is validated on an experimental plant.

In [4] the paper presents adaptive controller with online identification block for brushless dc motor. The paper presents two different neural networks schemes for identification and control for such system. The first scheme is designed to control the rotor angular speed by establish three hidden layer feedforward neural network. The second scheme is designed to control the stator current by using predetermined control law as a function of the estimated states. The second scheme has established three feedforward neural network trained online using Levenberg-Marquardt algorithm. The neural network control strategy adapts to the uncertainties of the motor load dynamics and nonlinearities. System simulation concluded that, neuro controller in conjunction with adaptive control scheme is flexible and reliable which can have wide utilization and application in the real life

In [5] authors propose a new adaptive control scheme, composed of a neural identifier and a nonlinear controller and applied it to a linear induction motor (LIM). In order to compare the performance of LIM, they use $\alpha-\beta$ and $d-q$ models. A neural identifier of triangular form is proposed for both models as a nonlinear block controllable form (NBC). Then, a reduced order observer is designed in order to estimate measured variables. Learning law for neural network weights ensure that the identification error converges to zero exponentially.

The effect of sampling period on the identification process is very important; it has big effect on the identification accuracy and stability. It is a kind of trading off i.e. rapid sampling causes a problem of stability and makes advantage of disturbance cancelation and good overshoot. Short sampling causes a problem of aliasing and makes advantage of good numerical stability.

For classical methods of identifications like RLS (recursive least square method) or LSM (least square method), short sampling period done for real time system identification fails even though the disturbance have been taken into account. This fact happens due to the quantization in A/D converter. So, the quantization, noise effect and other nonlinearities accompanied by real planets make online identification more complex than it could be expected. [6, 7] show that a possible solution of this problem is using of an identification method based on neural networks.

The schemes of these types of controller are separated into the two main parts: identification and controller. In this work, identification based on neural network approach is used and the control algorithm is based on self tuning regulator as an adaptive controller using PID controller as an adaptive controller.

2. On-Line Identification

On-line identification of process parameters is the key element in adaptive control. The basic idea of on line identification is to compare the output of estimated system with the output of model during some time. The model is describable as a parameter vector. The aim is to adjust parameter until the model output is similar to the observed system output. The classical Recursive Least Square (RLS) identification method [8] and gradient method compares only actual model output to system output, while the identification method based on neural network approaches compares output over some interval time defined by length of a training set.

Fig. 1 shows the principle of identification of system using neural network. A very fast algorithm for training neural networks is the Levenberg-Marquardt (LM) algorithm. The main idea of on-line identification is that according to the measured input to the identified system $u(t)$ and the corresponding system output $y(t)$ we are able to find the vector of system parameters θ. For computing of the identified system output we can use the linear model in equation (1) which represents the standard form of the actual process (plant).

$$F_m(z^{-1}) = \frac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} \quad (1)$$

The above model can be written in vector form as follows, where equation (2) represents the estimated output $\hat{y}(k)$ at sample k from the model (neural network) which will be compared with the plant output $y(k)$ at the same sample k and $\varphi(k)$ represents the vector of all inputs and outputs samples for prespecified sampling period, and finally $\theta(k)$ represents the vector of all the plant parameters which is needed to be identified.

$$\hat{y}(k) = \varphi(k)\theta(k) \quad (2)$$

Where

$$\varphi(k) = [u(k-1) \dots u(k-1-m) - y(k-1) \dots - y(k-1-n)]^T \quad (3)$$

is the vector of measured inputs and outputs and

$$\theta(k) = [b_1(k) \dots b_m(k)\ a_1(k) \dots a_n(k)]^T \quad (4)$$

is the vector of estimated system parameters and k denotes discrete time.

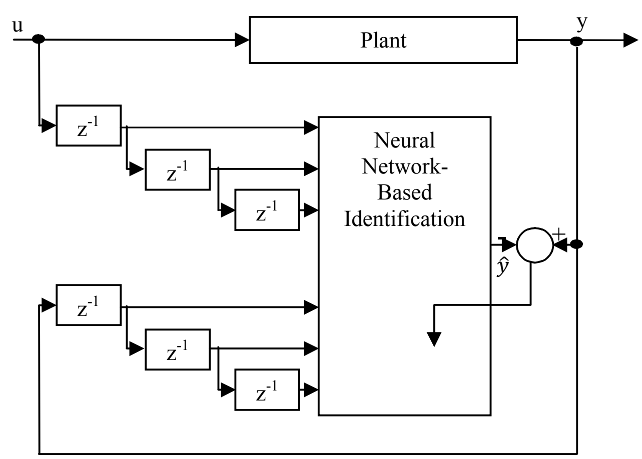

Fig. 1 The principle of identification of system using neural network

As it has been mentioned above, the Levenberg– Marquardt [7] method can be used for training of the neural network. The new vector of parameters is in each step given by equation (5).

$$\theta(k+1) = \theta(k) - (J^T J + \lambda I)^{-1} J^T \varepsilon(k) \quad (5)$$

Where j is the jacobian matrix as shown in equation (6).

$$J = \begin{bmatrix} \frac{\partial \varepsilon_1}{\partial w_1} & \cdots & \frac{\partial \varepsilon_1}{\partial w_j} \\ \vdots & \ddots & \vdots \\ \frac{\partial \varepsilon_i}{\partial w_1} & \cdots & \frac{\partial \varepsilon_i}{\partial w_j} \end{bmatrix} \quad (6)$$

$$\varepsilon(k) = y(k) - \hat{y}(k) \quad (7)$$

Where ε is the vector of errors, it measures the difference between the output of the system and the output of the model for all training sets. The parameter i denotes the number of training patterns and j denotes the number of estimated parameters.

One of the most important reasons why neural network approach, when used as an identification block, is preferable than the RLS and LSM approaches is that, it can adapt itself for different systems orders and even for nonlinearity which could be included in some systems which leads to better controller performance i.e. the controller could be more reliable and faster.

3. Simulation Results Of On-Line Identification

The Recurrent neural network with tap delay is chosen and implemented using MATLAB neural network toolbox, the network could be designated in different ways, many assumptions exits in the literature [9,10] for the number of hidden layers, number of neurons in each layer and type of activation function for each layer. Fig. 1 shows schematic diagram for the recurrent neural network for single input, single output system with delays on each. Initially, the training input vectors and target vectors of the recurrent network have been found from many simulation results for the original plant model represented in equation (1). The network has been designed with input bias at each neuron; the gain and the system stability have been checked and designed during the training process. The neural network has been trained and checked for specified accuracy during the off-line mode.

From the description above, the implementation of the neural network system identification block consists of just iterating the five equations (2 to 7) at every time instant k, that is described as in Fig. 1., where ε is the chosen measured error (required accuracy). The properties of the forgetting factor λ and its big effect on the RLS algorithm accuracy speed and even convergence to the right values of the estimated parameters has taken a lot of concentration and studies previously, these studies are summarized and presented in [8]. Where, most of authors, used to choose descent random value for λ in the range of (0-1), by experience authors conclude that choosing high value for λ (closet to 1) yields to slowest speed (but provide the best robustness toward noise) and decreasing values of λ result in increasing speed of parameter convergence at the expense of increased noise influence. In [8], the author presented new algorithm called adaptive RLS, to choose the optimum value of λ which give the best convergence to the system parameters. This algorithm has been used and implemented in our work for on-line identification.

Intensive simulation has been done using MATLAB, for the RLS algorithm. The simulation has been done for different processes have different orders such as the process shown in equation (1).

The simulation has been rerun for different values of λ and every time the simulation shows the estimated parameters and the instantaneous error. The optimum value for λ is (0.53) which shows robust convergence of the estimated parameters to their right values and zero error at the iteration number 200.

Fig. 2 shows the system identification results during on-line mode for random input for both original system (plant) and neural network system identification block. It is clear that the neural network output is very close to the plant output which means small accepted error.

Fig. 3 shows the error signal between the plant output and the neural network system identification block output during on-line mode for random. As shown the error signal tends to zero value after 20 input. This means small accepted value.

The neural network block has been tested for another input to make sure of its accuracy and performance. Fig. 4 shows the system identification results during on-line mode for sinusoidal input for both original system (plant model) and neural network system identification block. It is clear that the neural network output is very close to the plant output which means small accepted error.

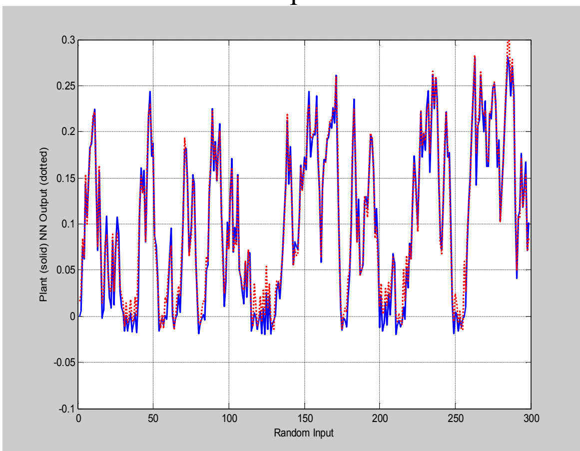

Fig. 2 neural network system identification for random input

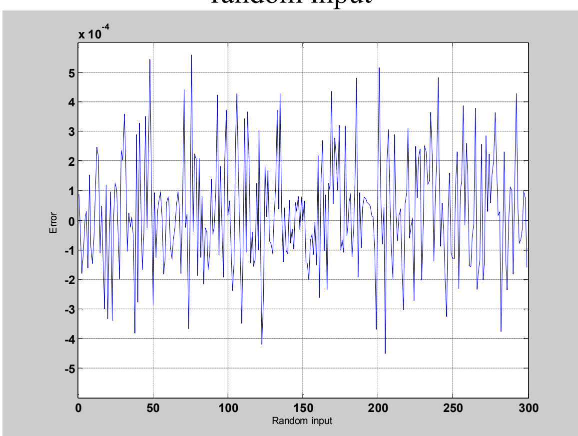

Fig. 3 Error for random input

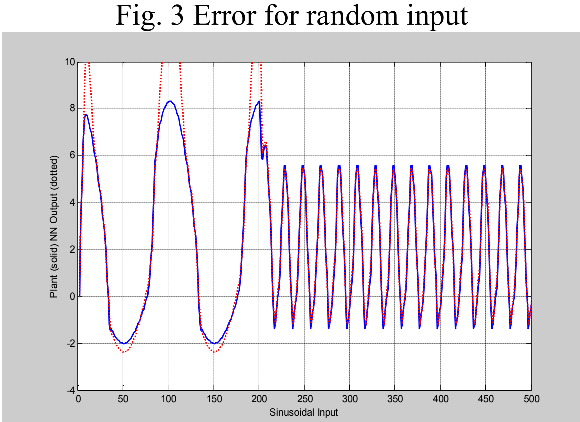

Fig. 4 neural network system identification for sinusoidal input

4. Adaptive Control

The idea of adaptive controllers (self-tuning controller's scheme STR) is to combine an on-line identification with on-line control law synthesis. Many of control law synthesis approaches are based on two methods – pole placement and inversion of dynamics. Both of the methods are numerically sensitive to the good estimation of the plant.

The requirement for correctly computed system parameters vector θ is not often fulfilled during controlling of real system with a higher order. Therefore, we use simple synthesis based on modified Zeigler –Nichole method. The basic architecture of adaptive controller (STR) we discussed is shown in Fig. 5.

The system (adaptive PID controller based on STR scheme, with neural network identification block) presents adaptive PID controller designed using Zeigler –Nichole method.

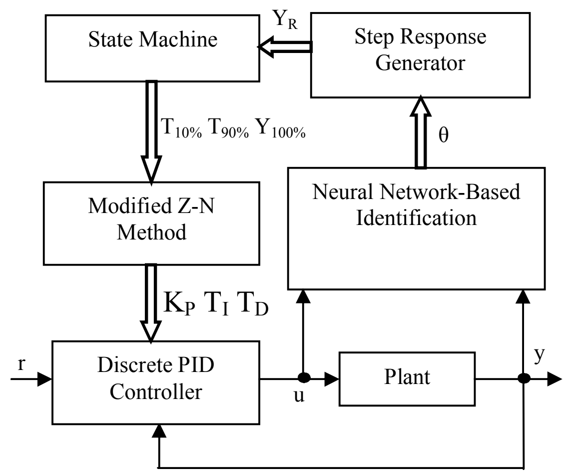

Fig.5 Block diagram of Recursive parameter estimate

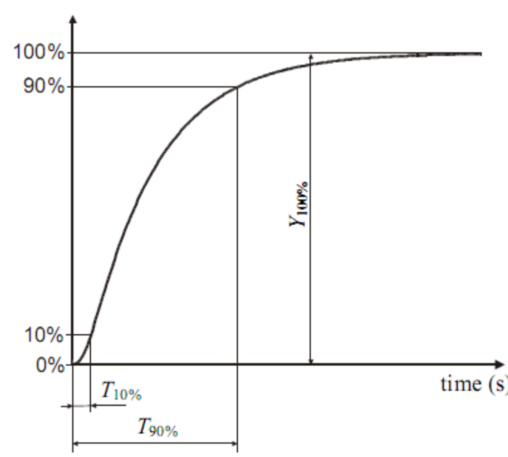

Fig.6 the characteristics points used for a tuning of the adaptive heuristics controller based on modified Ziegler-Nichols open loop method

ISBN: 978-1-61804-069-5

The step response generator generates the sequence Y_R of a step response of the estimated model θ. Then, the state machine finds characteristic points $T_{10\%}$, $T_{90\%}$ and $Y_{100\%}$ in the sequence Y_R Fig. 6. These values are used to design PID discrete controller according to [9,11]. As shown in equations (8, 9).

$$L = 0.8 * T_{10\%}, \quad R = \frac{Y_{100\%}}{T_{90\%} \; L} \tag{8}$$

$$K_P = \frac{0.8}{R * L}, \quad T_I = 3 * L, \quad T_D = 0.5 * L \tag{9}$$

Where K_P, T_I and T_D are the parameters that of the PID controller needed to find the PID gain.

5. Simulation Results

From the description above, the implementation of the adaptive PID controller based on neural network on line identification shown in Fig. 5 has been implemented and simulated using MATLAB.

Intensive simulation has been done using MATLAB. The simulation has been done for different plants have different orders follows the standard form shown in equation (1).

The following simulation results have been presented for a plant shown in equation (10)

$$F_m(z^{-1}) = \frac{b_1}{a_1 + a_2 z^{-1} + a_3 z^{-2} + a_4 z^{-2}} \tag{10}$$

The assumed desired control requirements have been chosen to get 30 sec settling time, critical damping performance (i.e. damping ratio = 1) and zero steady state error for all cases of changing the plant parameters or order.

First the simulation has been run for the following plant parameters (a_1 = 0.909, a_2 = 0.206, a3 = 1, b_1 = 0.0909), Fig. 7 shows the step response for the plant without control (i.e. the PID controller is switched off), the figure shows the plant output versus the time, as shown the output reaches its final value (0.5) at time 6 sec in a critical damping modes, which gives error value of 50%.

Fig. 8 shows the error between the plant output and the neural network output versus the iteration number. As shown the error reaches zero at iteration number 18 which means fast identification which leads to better adaptive control system.

Fig. 9 shows the plant step response after the PID controller adapted its parameters to fit the design requirements, the figure shows the plant output versus the time, as shown the output reaches its final value (1) at time 30 sec in a critical damping mode (i.e. zero overshoot), which gives error value of zero%.

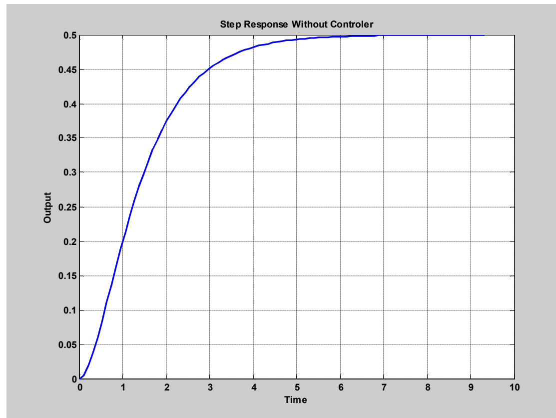

Fig.7 Step response for 2nd order plant without control

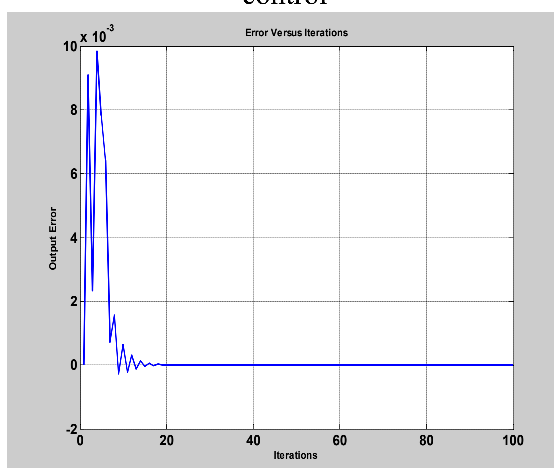

Fig.8 Error Versus iteration

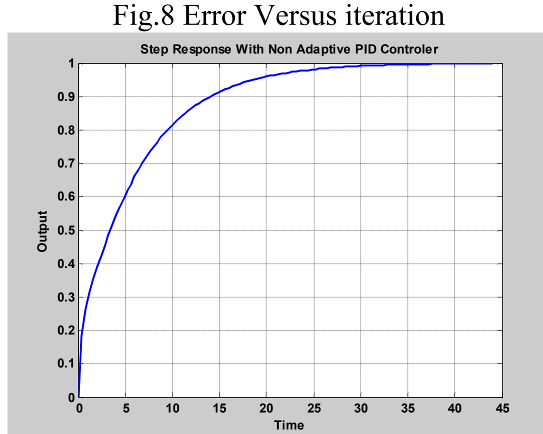

Fig.9 Step response for 2nd order plant with Adaptive PID (STR Scheme) control

ISBN: 978-1-61804-069-5

To check the adaptive PID controller system, first, we started to change the plant parameters to the following (a_1 = 0.909, a_2 = 0.02, a3 = 1, b_1 = 0.0909), and then first check the controller performance before adapting its parameters.

Fig. 10 shows the plant step response before the PID controller start adapting its parameters to fit the new changing in the plant parameters (i.e. the neural network block for on line identification is switched off). The figure shows the plant output versus the time, as shown the output reaches its final value (1) at time 36 sec in an under damping mode with overshoot of 35%, and gives error value of zero%. So, the comment on this result says that the performance of the PID controller is not bad, but it deviated from the desired design requirements, where the change in the plant parameters was not too large.

Fig. 11 shows the error between the plant output and the neural network output versus the iteration number. As shown the error reaches zero at iteration number 17 which means fast identification which leads to better adaptive control system.

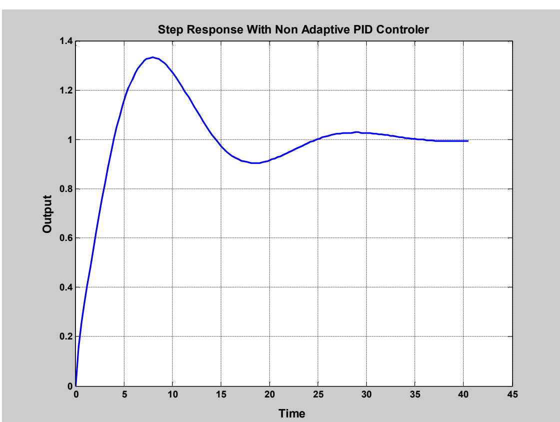

Fig.10 Step response for 2nd order plant with PID control after changing plant parameters

Fig. 12 shows the plant step response after the PID controller adapted its parameters to fit the design requirements, the figure shows the plant output versus the time, as shown the output reaches its final value (1) at time 23 sec in a critical damping mode (i.e. zero overshoot), which gives error value of zero%.

To continue checking the adaptive PID controller system, second, we started to change the plant order and parameters (i.e. new plant) to the following (a_1 = 0.909, a_2 = 0.02, a3 = 1, b_1 = 0.0909), and then first check the controller performance before adapting its parameters. As shown in Fig. 13, the step response of the plant step response before the PID controller start adapting its parameters to fit the new changing in the plant parameters (i.e. the neural network block for on line identification is switched off). The figure shows the plant output versus the time, as shown the output goes to infinity at time 170 sec which indicates unstable response and gives error value of infinity. So, the comment on this result says that the PID controller has failed to control such system and has to be redesigned to control the new plant, where the plant has been fully changed.

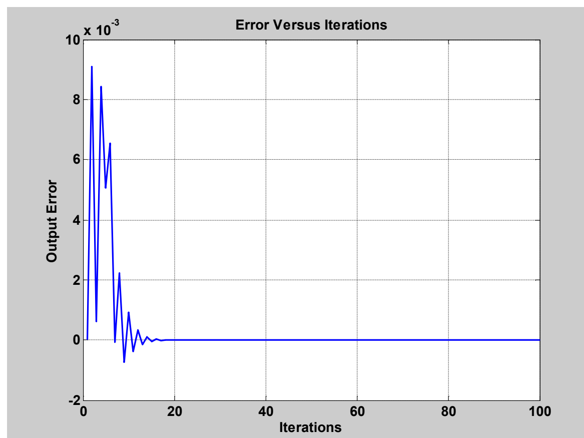

Fig.11 Error Versus iteration after changing plant parameters

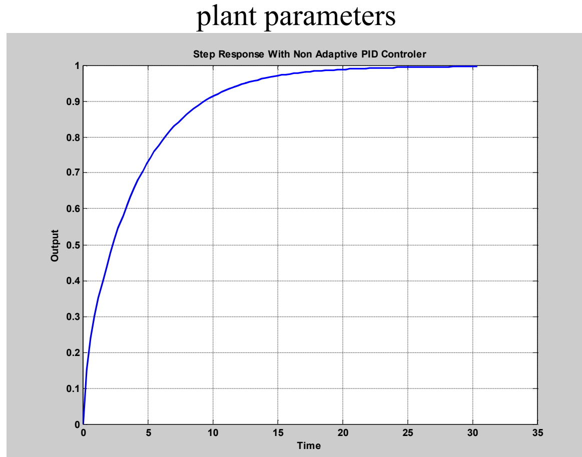

Fig.12 Step response for 2nd order plant with Adaptive PID (STR Scheme) control after changing plant parameters

ISBN: 978-1-61804-069-5

So, the first step is to turn on the neural network identification block to start identifying the plant parameters and dominant poles. Fig. 14 shows the error between the plant output and the neural network output versus the iteration number. As shown the error reaches zero at iteration number 19 which means fast identification which leads to better adaptive control system.

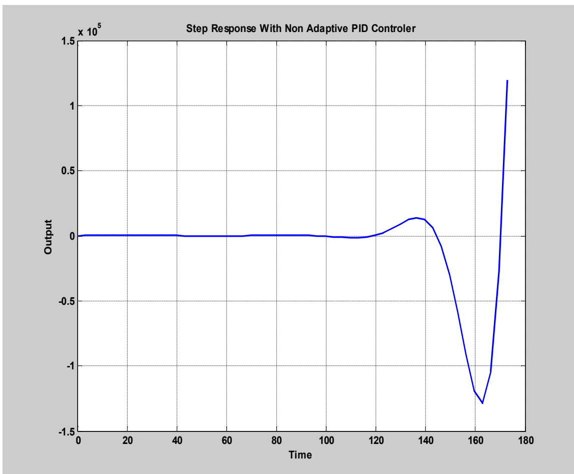

Fig.13 Step response for 2nd order plant with PID control after changing plant order

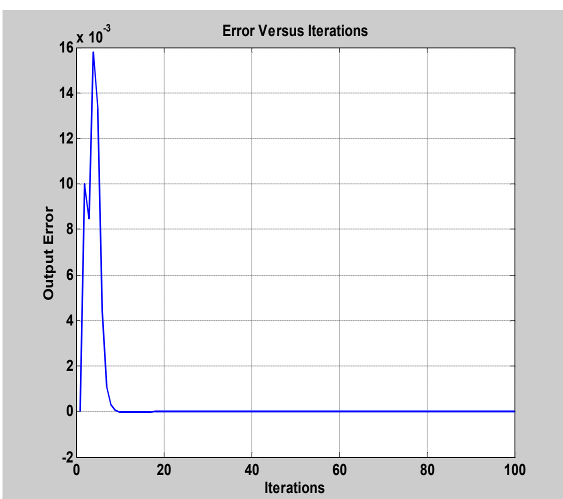

Fig.14 Error Versus iteration after changing plant parameters

Fig. 15 shows the plant step response after the PID controller adapted its parameters to fit the design requirements, the figure shows the plant output versus the time, as shown the output reaches its final value (1) at time 45 sec in a critical damping mode (i.e. zero overshoot), which gives error value of zero%.

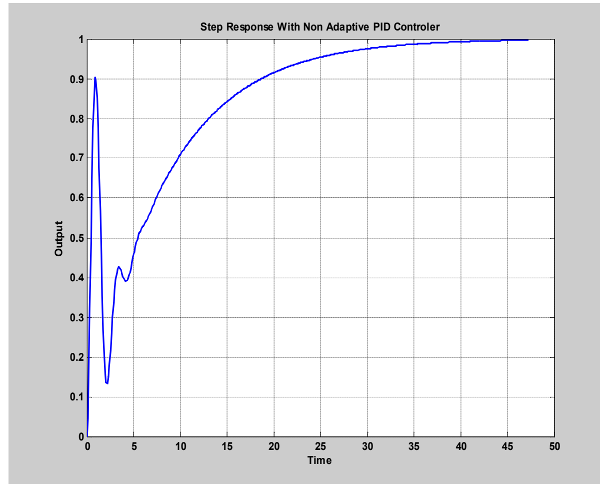

Fig.15 Step response for 2nd order plant with Adaptive PID (STR Scheme) control after changing plant order and recovering the PID parameters

6. Conclusion

Adaptive discrete PID controller analysis, as a self tuning regulator scheme, has been presented with artificial neural network as on line system identification in mind. On line system identification block using neural network has been presented. Recurrent neural network type for system identification has been presented and analyzed. Neural network has been designed and trained for identifying plant parameters. Modified zeigler –Nichole method for designing PID controller has been presented and simulated using MATLB. Analysis for the whole scheme model is presented and simulated using MATLAB for different plants order. The simulation has been done for different cases of changing plant parameters and changing plant order. Adequate desired performance is obtained with the required accuracy and response.

References:

[1] P. Pivonka, V. Lorenc, “Real-Time Implementation of Adaptive Optimal Controller Based on a Pseudo-Space Model into PLC”, 11th WSEAS International

Conference on Systems, Crete Island, Greece, PP 23-27, July 23-25, 2007.

[2] O. Kuljaca, S. Tesnjak, V. Koroman, "Performance Analysis of Adaptive Neural Network Frequency Controller for Thermal Power Systems", 9th WSEAS International Conference on Automatic Control, Modeling & Simulation, Istanbul, Turkey, PP 127-132, May 27-29, 2007.

[3] B. Daachi, A. Benallegue, "A Stable Neural Adaptive Force Controller For a Hydraulic Actuator", Proc. Instn Mech. Engrs, Vol. 217, Part I, J. Systems and Control Engineering, PP 303-310, 2003.

[4] A. Rubaai, R. Kotaru, M. D. Kankam, "A Continually Online-Trained Neural Network Controller for Brushless DC Motor Drives", IEEE Transaction On Industry Applications, Vol. 36, No. 2, March/April, PP 575-583, 2000.

[5] V. H. Benitez, E.N. Sanchez, A. G. Loukianov, "Neural Network Identification and Control for Linear Induction Motors", Journal of Intelligent & Fuzzy Systems, Vol. 16, IOS Press, PP 33-55, 2005.

[6] P. Pivonka, V. Veleba, p. Osmera, "Using of Neural Network Based Identification for Short Sampling Period in Adaptive Control", 11th WSEAS International Conference on Systems, Crete Island, Greece, PP 12-17, July 23-25, 2007.

[7] P. Pivonka, K. Svancara, "The Use of Short Sampling Period in Adaptive Control", WSEAS Transaction on Systems, ISSN 1109-2777, Vol. 10, No. 4, PP 1627-1636, 2005.

[8] H.E.A. Ibrahim, "Adaptive Recursive least square (RLS) Using Genetic Algorithm", ASJEE, ISSN 1687-8582, Vol. 2, PP 35-45, Dec. 2009.

[9] A. Goedtel, I. Silva, P. Serni, "Neural Network Based Estimation of Torque in Induction Motors for Real-Time Applications", Taylor & Francis, Journal of Electric Power Components and Systems, ISSN 1532-5008, PP 363-387, 2005.

[10] B. Daachi and A. Benallegue, "A Stable Adaptive Force Controller for a Hydraulic Actuator", Proc. Instn. Mech. Engrs Vol. 217 Part I: J. Systems and Control Engineering, PP 303-310, 2003, I08402© IMechE 2003.

[11] Astrom, K., J., Wittenmark, B., "Computer Controlled Systems Theory and Design", Prentice Hall, Inc., 1997.