TECHNOLOGY HORIZONS JOURNAL

Vol. 2 (3), 10 March 2018, pp. 57-62

Adaptive Control Based on Neural Network System

Malaz Gaafer Mohammed[1], Eltahir Mohamed Hussein[2]

1Control Engineering, Alneelain University, Khartoum, Sudan

2College of Engineering, Sudan University of science and technology, Khartoum, Sudan

Corresponding author E-mail: sanhourym@hotmail.com

Received: 30 December 17

Accepted:15February 18

ABSTRACT

In adaptive control and system identification the self-accuracy regulator has wide range of application Neural network and artificial intelligence are playing big role in this area This paper presents adaptive neural network control based on Self Tuning Regulator (STR) scheme. An On-line identification of process parameters is the key element in adaptive control, the basic idea of online identification is to compare the output of estimated system with the output of model during some time, and 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. This paper presents neural network block for online system identification and discrete PID block controller. Modified Zeigler –Nichole method for designing PID controller has been presented and simulated using MATLAB. Analysis of 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.

***Keywords:- Adaptive Control; Self Tuning Regulator; System Identification; Neural Network; Neuro-Control*;**

I. INTRODUCTION

The control of a process variable requires three basic functional units- the measuring equipment, the controller, and the final controlling equipment - as well as the knowledge of how to make proper use of the individual belonging elements. Usually, these control loop components are separate devices that must be supplied with auxiliary energy. Self-operated regulators take over all the tasks required in a control loop.

They integrate measuring sensor, controller as well as control element all in one system. Neuro controller in conjunction with adaptive control scheme is flexible and reliable which can have wide utilization and application in the real life. The purpose of adaptive controllers is to adapt control law parameters of control law to the changes of the controlled system for identification of the controlled system an algorithm based on the artificial neural network is used

II. METHOD

On-line identification of process parameters is the key element in adaptive control. The basic idea of online identification is to compare the output of the 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 [1] 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 the length of a training set. Figure. 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[2].

The main idea of online 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)

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

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

$$\gamma(k)=\varphi(k)\theta(k) \qquad (2)$$

$$\varphi(k)=\left[u(k-1)..u(k-1-m)y(k-1)...-y(k-1-n)\right]^T \qquad (3)$$

Is the vector of measured inputs and outputs and

$$\theta(k)=\left[b_1(k)....b_m(k)a_1(k)....a_n(k)\right]^T \qquad (4)$$

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

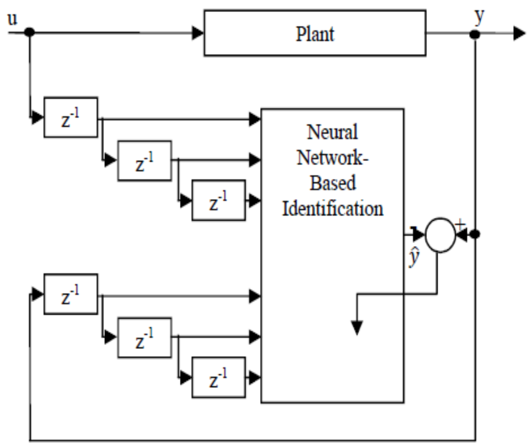

Figure. 1 The principle of the system using neural network

The Levenberg– Marquardt method [9] can be used for training of the neural network. The new vector of parameters is in each step given by equation (5).

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

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

$$J=\begin{bmatrix}\frac{\partial\varepsilon_1}{\partial\omega_1} & \cdots & \frac{\partial\varepsilon_1}{\partial\omega_j}\\ . & . & .\\ . & . & .\\ . & . & .\\ \frac{\partial\varepsilon_i}{\partial\omega_1} & \cdots , & \frac{\partial\varepsilon_i}{\partial\omega_j}\end{bmatrix} (6)$$

-

$$\varepsilon(k)=\gamma(k)-\gamma^{\wedge}(k) \qquad (7)$$

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 to 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.

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[3- 4] for the number of hidden layers, number of neurons in each layer and type of activation function for each layer. Figure. 1 shows schematic diagram of 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 (2to 7) at every time instant k, that is described as in Figure. 1, where ε is the chosen measured error (required accuracy). The properties of the forgetting factor λ and it's big effect on the RLS algorithm accuracy speed and even convergence to the right values of the estimated parameters have taken a lot of concentration and studies previously, these studies are summarized and presented in [1].

Most of the 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 increased speed of parameter convergence at the expense of increased noise influence.

In [1], 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 the robust convergence of the estimated parameters to their right values and zero error at the iteration number 200. Figure. 2 shows the system identification results during online mode for random input for both original system (plant) and neural network system identification block.

The neural network output is very close to the plant output which means small accepted error. Figure. 3 shows the error signal between the plant output and the neural network system identification block output during online 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. Figure. 4 shows the system identification results during online 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.

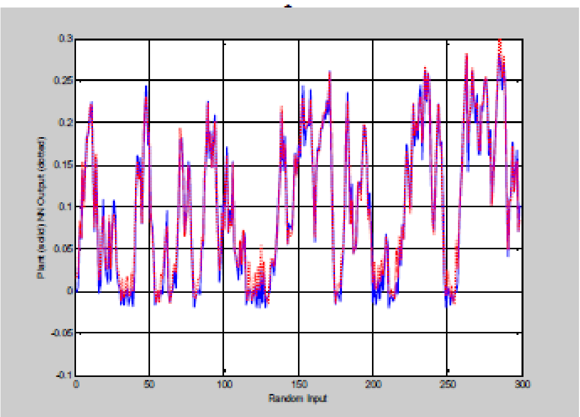

Figure. 2 neural network system identification for random input

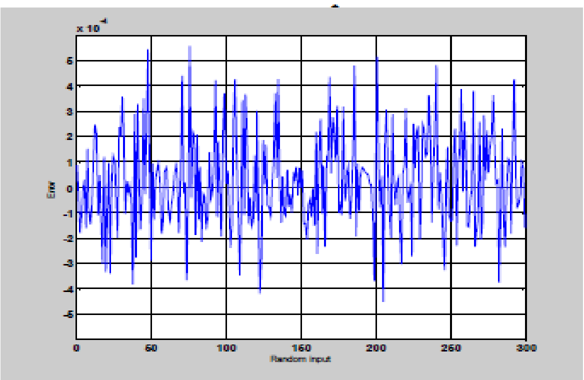

Figure. 3 Error for random input

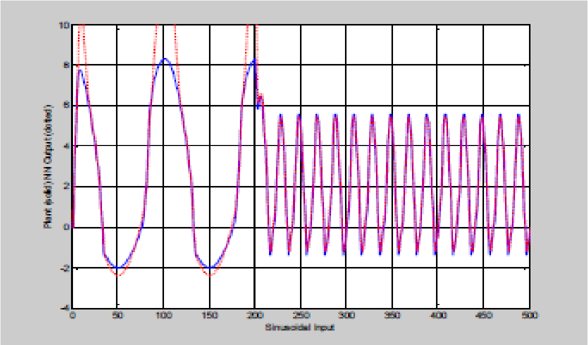

Figure. 4 neural network system identification for sinusoidal input

The idea of adaptive controllers (self tuning controller's scheme STR) is to combine an online 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 Figure. 5. The system (adaptive PID controller based on STR scheme, with neural network identification block) presents adaptive PID controller designed using Zeigler – Nichole method.

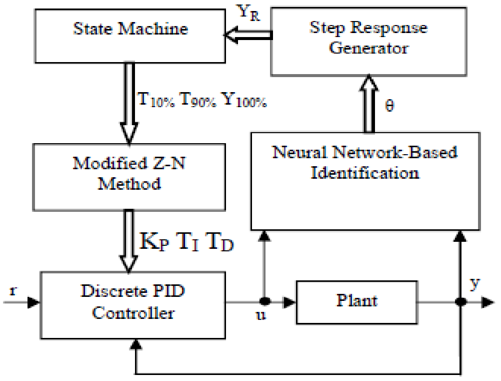

Figure.5 Recursive parameter estimate

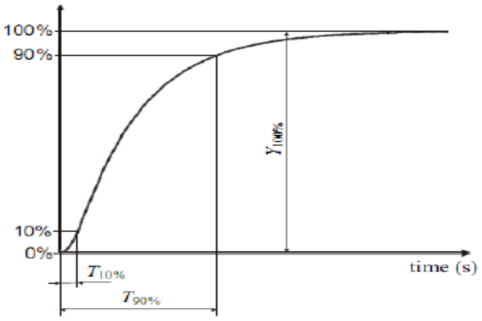

Figure 6 The characteristics points used for a tuning of the adaptive heuristics controller based on modified Ziegler-Nichols open loop method

The characteristics points used for a tuning of the adaptive heuristics controller based on modified Ziegler-Nichols open loop method the step response generator generates the sequence YR of a step response of the estimated model θ. Then, the state machine finds characteristic points T10%, T90% and Y100% in the sequence YR Figure. 6. These values are used to design PID discrete controller according to [3-5]. As shown in equations (8, 9).

$$L = 0.8 * T_{10\%}, R = \frac{Y_{100\%}}{T_{90\%} - L} \qquad (8)$$

$$K_p = \frac{0.8}{R * L}, T_1 = 3 * L, T_D = 0.5 * L \qquad (9)$$

Where K_p , T_1 and T_D are the parameters that of the PID controller needed to find the PID gain.

Simulation Results

From the description above, the implementation of the adaptive PID controller based on the neural network online identification shown in Figure. 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)

$$F_m(Z^{-1}) = \frac{b_1}{a_1 + a_2 z^{-1} + a_3 z^{-2} + a_4 z^{-3}} \qquad (10)$$

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 parameter (a1 = 0.909, a2 = 0.206, a3 = 1, b1 = 0.0909) Figure. 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 mode, which gives error value of 50%.

Figure. 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 the better adaptive control system. Figure. 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%.

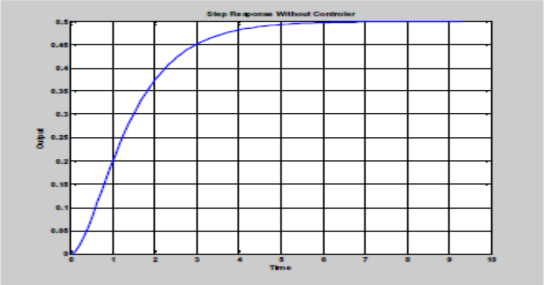

Figure.7 2nd order plant without control

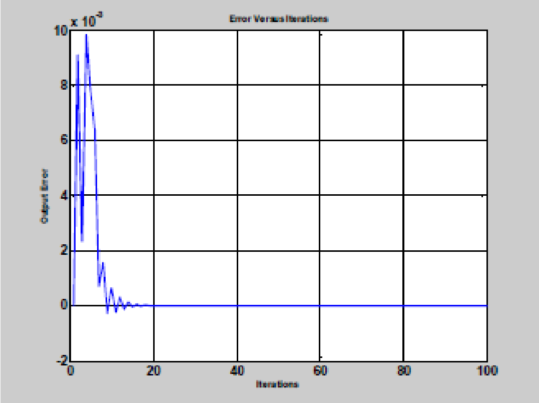

Figure.8 Error versus Iteration

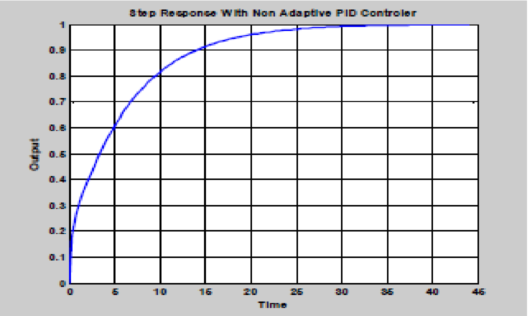

Figure.9 Step response for 2nd order plant with Adaptive PID (STR Scheme) control

To check the adaptive PID controller system, first, we started o the following (a1 = 0.909, a2 = 0.02, a3 = 1, b1 = 0.0909), and then first check the controller performance before adopting its parameters. Figure. 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 online 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 an 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. Figure. 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 the better adaptive control system.

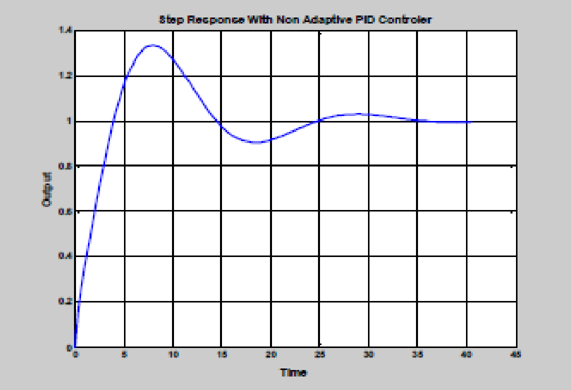

Figure.10 Step response for 2nd order plant with PID control after changing plant parameters

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 (a1 = 0.909, a2 = 0.02, a3 =1, b1 = 0.0909), and then first check the controller performance before adopting its parameters. As shown in Figure. 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.

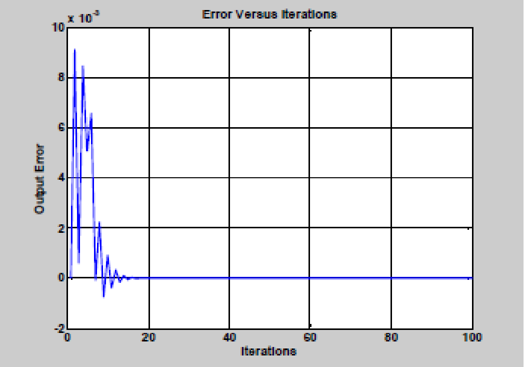

Figure.11 Error Versus iteration

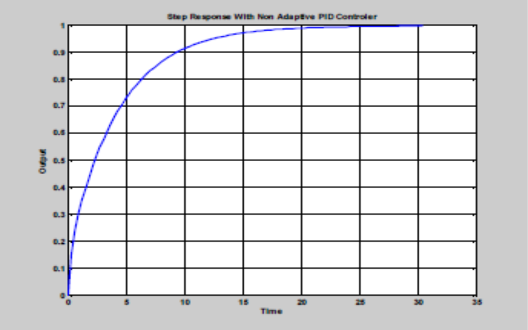

Fig.12 2nd order plant with Adaptive PID (STR Scheme) control after changing plant parameters

The first step is to turn on the neural network identification block to start identifying the plant parameters and dominant poles. Figure. 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 the better adaptive Control system.

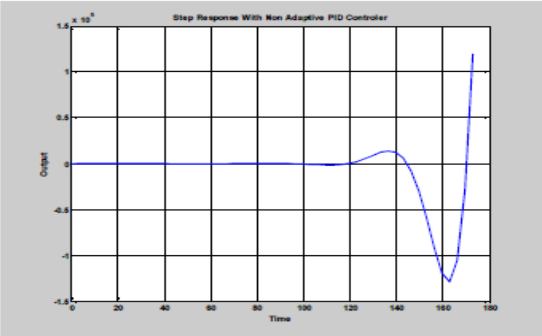

Figure.13 Step response for 2nd order plant with PID control after changing plant order

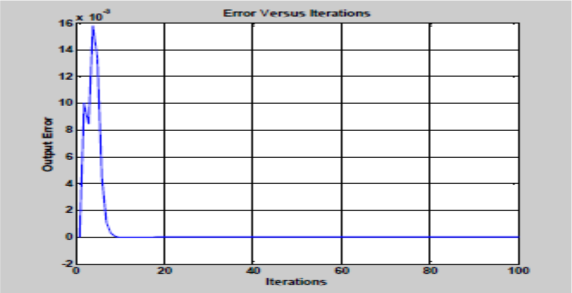

Figure.14 Error Versus iteration after changing plant parameters

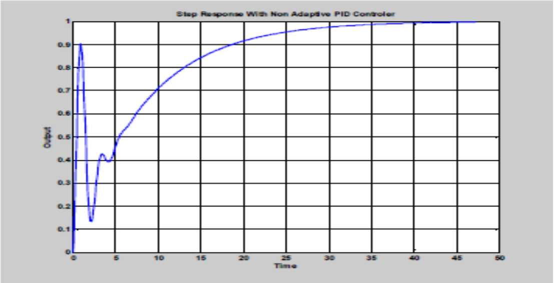

Figure.15 2nd order plant with Adaptive PID (STR Scheme) control after changing plant order and recovering the PID parameters. Figure. 15 shows the plant step response after the PID controller adapted its parameters to fit the design requirements, the figure shows the plant output versus the time, as shown the output reaches its final value (1) at time 45 sec in a critical damping mode (i.e. zero overshoot), that gives error value of zero%.

III. CONCLUSION

Adaptive discrete PID controller analysis, as a self-tuning regulator scheme, has been presented with the artificial neural network as online system identification in mind. Online system identification block using neural network has been presented. Recurrent neural network type for system identification has been presented and analyzed. The neural network has been designed and trained for identifying plant parameters. Modified Ziegler – Nichole method for designing PID controller has been presented and simulated using MATLAB. Analysis of 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.

REFERENCE

[1] H. E. Ibrahim, "Adaptive control based on neural network system identification," in *Proceedings of the 11th WSEAS international conference on Electronics, Hardware, Wireless and Optical Communications, and proceedings of the 11th WSEAS international conference on Signal Processing, Robotics and Automation, and proceedings of the 4th WSEAS international conference on Nanotechnology*, 2012, pp. 84-91.

[2] P. Pivonka and K. Svancara, "The Use of Short Sampling Period in Adaptive Control," *WSEAS Transactions on Systems*, vol. 4, pp. 1627-1636, 2005.

[3] A. Goedtel, I. N. D. SILVA, and P. J. A. SERNI, "Neural network based estimation of torque in induction motors for real-time applications," *Electric Power Components and Systems*, vol. 33, pp. 363-387, 2005.

[4] B. Daachi and A. Benallegue, "A stable neural adaptive force controller for a hydraulic actuator," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 217, pp. 303-310, 2003.

[5] K. J. Åström and B. Wittenmark, *Computer-controlled systems: theory and design*: Courier Corporation, 2013.