Self-tuning based on radial basis function neural network of a linear system

Gaber Elsaady[1], Abd-El Meged Mohamed[2], Ashraf Hemeida[3], Asmaa Fawzy[4]
[1]*engineering faculty, Assiut university, Egypt* [2]*engineering faculty, Aswan university, Egypt*
[3,4]*energy engineering faculty, Aswan university, Egypt*

Abstract— This paper introduces the inverse control design using neural network based self tuning regulator (STR). The control algorithm performs equally very well to both minimum phase and non-minimum phase of linear plants. The controller is the radial basis function neural network (RBFNN) and acts as inverse of the plant. The plant parameters are estimated online using the system identification method where uses the Autoregressive with exogenous input (ARX) model which depends on the input and output values of the plant. The difference between the output of the plant and the reference signal is used to adjust coefficients of ARX model. These coefficients of ARX are used to update the weights of the RBFNN. The weight update equations are derived based on the least mean squares principle (LMS). The output tracks the reference trajectory though the self tuning regulator (STR) structure exposed to different types of disturbances for wide range of operating conditions. Then, the algorithm compared with the self tuning of proportional-plus-integral feedback (PI) controller. The proposed algorithm is successfully verified using simulations for both minimum and non-minimum phase of linear plants counter to using PI.

Index Terms— STR, RBFNN, ARX, LMS, Single-phase full-converter drive, flexible transmission

1. INTRODUCTION

The development of a control system involves many tasks such as modeling, design of the control law, implementation and validation. The self-tuning regulator (STR) attempts to automate several of these tasks.
The Self Tuning Regulator is composed of two loops: the inner loop which contains the process and an ordinary feedback controller, the outer loop which is composed by a recursive parameter estimator and design calculations and this loop adjusts the controller parameters.
Indirect adaptive algorithm: two steps: (1) estimate process model parameters; (2) update controller parameters as if estimates were correct [1]. This is illustrated in fig. (1).

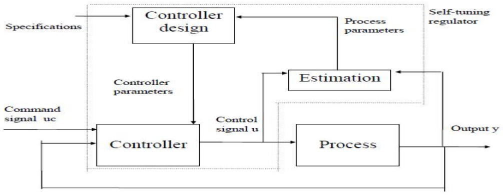


Fig. 1: Block diagram of a self-tuning regulator

Adaptive control schemes are used for the control of plants, where the parameters of the plant are not known exactly or slowly time varying. Some reasons for using Adaptive control such as variations in process dynamics and variations in the character of the disturbances [1],[2]. Enzeng ,Shuxiang and others [3] present a neural network based self tuning PID controller for autonomous underwater vehicle, the control system consists of neural network identifier and neural network controller, and the weights of neural networks are trained by using Davidon least square method, also[4].

Neural network (NN) is a good structure for control the nonlinear plants and has many types [5],[6]. Kumar and Ray [7] used neural network for modeling the retention process and as controller. In this paper, we used the RBFNN as a controller. This type is faster one and uses least number of neurons at hidden layer [8],[9]. The inverse control means that the controller (RBFNN) acts the inverse of the plant so the output tracks the reference input [10].

The DC motors have been extensively used in control systems. The main advantages of dc motors are easy speed or position control and wide adjustable range Therefore, DC motors are often used in a variety of industrial applications such as robotic manipulator, where a wide range of motions are required to follow a predetermined speed or position trajectory under variable load [11],[12],[13]. K.sabahi [14] used a new adaptive and nonlinear control based on neural network approaches, this method has been named feedback error learning (FEL) approaches, that classical controller is used for training of neural network feedforward controller.

Pal and Naskar [15] proposed a simple self-tuning scheme for PI-type fuzzy logic controllers (FLCs) for a real time water pressure control system. This scheme is improved performance of the system even at load change and set point variations. Kota and Goud [16] used PID controller and fuzzy logic controller for control separately excited dc motor. Fuzzy self-tuning PID has better dynamic response curve, shorter response time, small overshoot, and small steady state error compared to the conventional PID controller. Fawaz and others [17] presented a simulation and hardware implementation of a closed loop control of a separately excited dc motor using a self-tuning PID controller. It gives very acceptable results in the reduction of overshoot, stability time and the steady-state transient response of the controlled plant. Saad [18] proved that the proposed Neural Network (NN) self-tuning PID controller is more efficient to control the robot manipulator to follow the desired trajectory compared to classical tuning method of PID controller. Alfonso and others [19] introduced a new self-tuning algorithm is developed for determining the Fourier Series Controller coefficients with the aim of reducing the torque ripple in a Permanent Magnet Synchronous Motor (PMSM), thus allowing for a smoother operation. This algorithm adjusts the controller parameters based on the component's harmonic distortion in time domain of the compensation signal.

An application of the multiple models adaptive control based on switching and tuning to a flexible transmission system presented by Alireza K. and Dore L [20]. This approach has been considered in order to assure high control performance in the presence of large load variation on the system.

In this paper a new technique is proposed that gives a good control for the linear plants. An online control algorithm is structured using the radial basis function neural network (RBFNN). The plant parameters are estimated on line and are used to update the weights of the RBFNN. The weight update equations are derived based on the least mean squares principle. The RBFNN virtually models the inverse of the plant and thus the output tracks the reference trajectory. This scheme is exposed to several types of disturbances for wide range of operating conditions. The self tuning regulator (STR) meets the aforementioned disturbances separately and simultaneously.

2. Proposed Structure

The proposed structure is shown in Fig. (2), in the estimator block the Autoregressive with exogenous input (ARX) is used to model the plant. The estimates are updated online for any changes in real-time. These estimates are fed to the weight update block for the RBFNN.

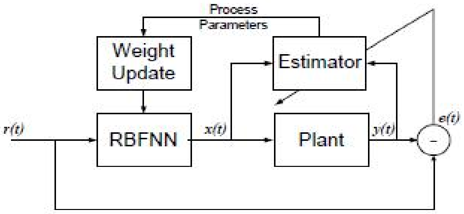


Fig. 2 Proposed self-tuning regulator structure

2.1 ARX Model

The ARX model is by far the most widely applied linear dynamic model. The popularity of the ARX model comes from its easy-to-compute parameters. The ARX model can be obtained by choosing $A(q)$ and $B(q)$ arbitrary polynomial.

$$A(q)y(k) = B(q)u(k) + v(k)$$

$$B(q) = b_1 q^{-1} + b_2 q^{-2} \ldots + b_m q^{-m}$$

$$A(q) = 1 + a_1 q^{-1} + a_2 q^{-2} + \ldots + a_n q^{-n}$$

$$y(t) = \frac{B(q^{-1})}{A(q^{-1})} q^{-d} u(t)$$

2.2 Radial Basis Functions Neural Networks

A single input single output radial basis function neural network (SISO RBFNN) is shown in Fig. (3). It consists of an input node $r(t)$, a hidden layer with n neurons and an output node $x(t)$. Each of the input node is connected to all the nodes in the hidden layer through unity weights (direct connection). While each of the hidden layer nodes is connected to the output node through some weights $w_1, \ldots, w_{n_0}$.

Each neuron finds the distance d of the input and its center and passes the resulting scalar through nonlinearity. So the output of the hidden neuron is given by [8], [21]

$$\phi(d) = \exp(-\tfrac{1}{2} d^2) = \exp(-\tfrac{1}{2} \|r(t) - c_i\|_{\Sigma}^2)$$

$$d = \sqrt{\left(\frac{r(t) - c_1}{\beta_1}\right)^2 + \ldots + \left(\frac{r(t) - c_{n_0}}{\beta_2}\right)^2}$$

c_i is the center of i^{th} hidden layer node where $i = 1,2,\ldots,n_0$, Σ is the norm matrix and $\phi(.)$ is the nonlinear basis function. Normally this function is taken as a Gaussian function of width β. The output $x(t)$ is a weighted sum of the outputs of the hidden layer, given by

$$x(t) = \sum_{i=1}^{n_0} w_i \phi(\|r(t) - c_i\|_{\Sigma})$$

As we see the radial basis function (RBF) network utilized a radial construction mechanism. This gives the hidden layer parameters of RBF networks a better interpretation than for the multilayer perceptron network MLP, and therefore allows new, faster training methods.

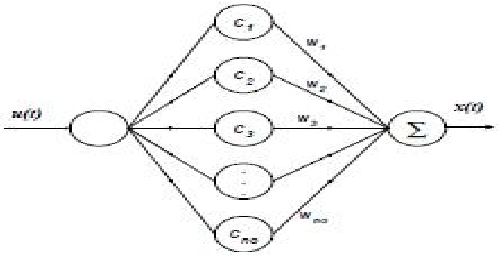


Fig. 3: A general RBF network

2.3 Parameters estimation for self-tuning of RBFNN

The training algorithm is based on a recursive scheme (least mean square) of estimating the parameters of the ARX model and the weights of the RBFNN. The parameters (the coefficients of the ARX model and the weights of the RBFNN) are updated by minimizing the performance index I given by [9]

$$I = \tfrac{1}{2} e^2(t)$$

$$e(t) = r(t) - y(t)$$

Where $r(t)$ is the reference input signal and $y(t)$ is the output of the plant. The coefficients of the ARX model and the weights of the RBFNN are updated in the negative direction of the gradient as,

$$\theta(K+1) = \theta(K) - \mu \frac{\partial I}{\partial \theta(K)}$$

And

$$W(K+1) = W(K) - \mu \frac{\partial I}{\partial W(K)}$$

Where $\theta = [a_1 \ldots a_n \; b_1 \ldots b_m]$ is the parameter vector, $W = [w_1 \; w_2 \; \ldots \; w_{n_0}]$ is the weight vector for RBFNN and μ is the learning parameter. The variable K is used to show the iteration number of training.

Keeping the regressions of the variables in the system in a regression vector ψ as $\psi(t) = (-y(t-1) \; \ldots -y(t-n) \; x(t-d) \; \ldots \; x(t-m-d))$ and finding partial derivatives.

$$\frac{\partial I}{\partial \theta} = \frac{1}{2} \frac{\partial e^2(t)}{\partial \theta}$$

$$= e(t) \frac{\partial}{\partial \theta} (r(t) - y(t))$$

$$= e(t) \frac{\partial}{\partial \theta} \left(r(t) - \frac{B(q^{-1})}{A(q^{-1})} q^{-d} x(t) \right)$$

$$= e(t) \frac{\partial}{\partial \theta} \left(r(t) - \left(-a_1 q^{-1} - \ldots a_n q^{-n} \right) y(t) - \left(b_1 q^{-1} + \ldots + b_m q^{-m} \right) q^{-d} x(t) \right)$$

$$\frac{\partial I}{\partial \theta} = -e(t) \psi(t)$$

The final parameter update equation will be,

$$\theta(K+1) = \theta(K) + \mu e(t) \psi(t)$$

The partial derivatives for the weights are derived as follows,

$$\frac{\partial I}{\partial W} = \frac{1}{2} \frac{\partial e^2(t)}{\partial W}$$

$$\frac{\partial I}{\partial W} = -e(t) B(q^{-1}) q^{-d} \phi(t)$$

The final weight update equation will be,

$$W(K+1) = W(K) + \mu e(t) B(q^{-1}) q^{-d} \phi(t)$$

But the final coefficients update equation of PI will be,

$$PI(K+1) = PI(K) + \mu e(t) B(q^{-1}) (-x(t-1) + (t_s + 1) x(t))$$

Where $PI = [k_p \quad k_i]$ is the vector for parameters proportional-integral (PI) and t_s is the sample time.

3. Simulation Results

The proposed STR structure is tested for both minimum and non-minimum phase linear plants like as the Single-phase full-converter drive and the flexible transmission respectively.

3.1 Single-phase full-converter drive system

The dc motor dynamic are given by the following equations:

$$K_b w(t) = -R i_a(t) - L \frac{d i_a(t)}{dt} + v_a(t)$$

$$K_m i_a(t) = J \frac{dw(t)}{dt} + b w(t)$$

Where $w, v_a, i_a, R, L, b, J, K_m$ and K_b the rotor speed, terminal voltage, armature current, armature resistance, armature inductance, damping constant, rotor inertia, torque constant and back emf constant, respectively. Fig. (4) describes the block diagram of the DC motor, using the superposition principle to obtain the output (w) with two inputs (v_a, T_d) :

$$W(s) = \frac{K_m V_a(S) + (LS + R) T_d(s)}{JLS^2 + (Lb + JR)S + (Rb + K_m K_b)}$$

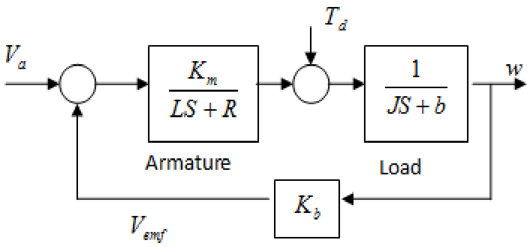


Fig. 4: The block diagram of the dc motor

ZOH (zero order hold) is used to convert a continuous-time dc motor model to a discrete-time one, the output equation can be written in the form of difference equation.

If the armature circuit of a dc motor is connected to the output of a single-phase controlled rectifier, the armature voltage can be varied by varying the delay angle of the converter, α, as [22]:

$$V_a = \frac{2V_m}{\pi}\cos\alpha \qquad for \quad 0 \le \alpha \le \pi$$

The proposed (STR) structure exposed to different disturbances as dc motor does in life. The dc motor meets to changing in its parameters due to increased temperature. This paper studies these variations where the motor model parameters are estimated online and are used to update the weights/coefficients of the RBFNN/PI at the same instant.

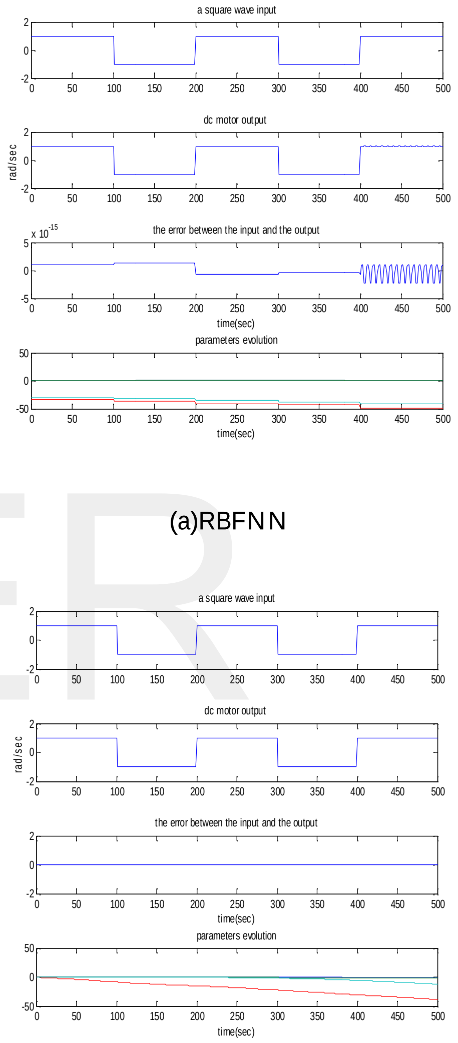
Fig.(5) shows the efficient response of the RBFNN/PI to the square input and the error signal between the reference and the RBFNN/PI output, in the case no disturbance.

The changing of temperature has an impact upon the parameters of a dc motor as an armature resistance and rotor inertia. Fig.(6) shows the effect of variance for the armature resistance at specified period $t \ge 300$, (a) in RBFNN case, the resistance has the range $2 \le R \le 14.8$. The output system takes the same trajectory until the armature resistance became 14.9Ω. But, (b) PI case, the resistance has the range $2 \le R \le 15$. The output system mimics the trajectory exactly until the armature resistance became 15Ω. But the error between the response of the system and the reference signal is not smaller than radial basis regulator.

Fig.(7) shows the effect of variance for the rotor inertia at specified period $t \ge 300$; (a) the parameters evolution turn into large values when the rotor inertia increases, so the inertia has the range $0.02 \le J \le 0.08$ when RBNN used. (b) The speed of the dc drive follows the excitation signal when used PI; however, the ARX model parameters value fluctuate sharply and this is not good.

In the life, we must take account of the disturbance torque. Fig.(8) illustrate the efficient output while this disturbance reach to maximum value $Td = -1.5\ N.m$ with radial basis self tuning. Likewise, PI self tuning have over this disturbance until reach to maximum value $Td = -1.7\ N.m$ at specified period $300 \le t \le 400$.

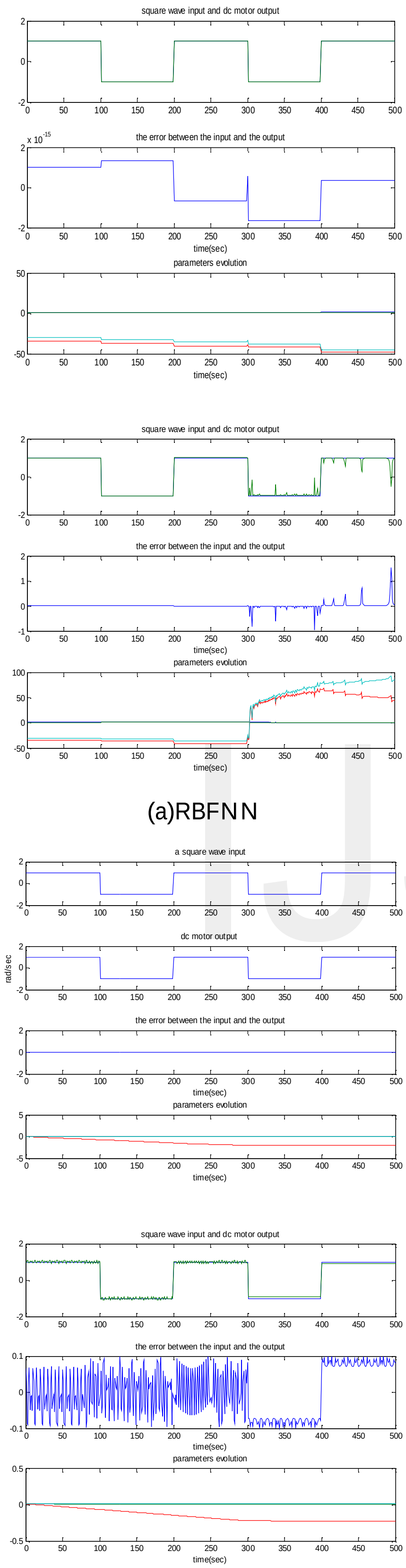
The previous mentioned disturbances are applied synchronized. fig.9.a, at case 1 when $R = 10\ \Omega$, $J = 0.05\ Kg.m^2$ at $t \ge 300$ and $Td = -1\ N.m$ at $220 \le t \le 310$; and case 2, when $R = 10\ \Omega$, $J = 0.05\ Kg.m^2$ at $t \ge 300$ and $Td = -1\ N.m$ at $300 \le t \le 400$.fig.9.b, at case 1 when $R = 4\ \Omega$, $J = 0.04\ Kg.m^2$ at $t \ge 300$ and $Td = -.4\ N.m$ at $220 \le t \le 310$; and case 2 when $R = 3\ \Omega$, $J = 0.03\ Kg.m^2$ at $t \ge 300$ and $Td = -.3\ N.m$ at $300 \le t \le 400$. The PI self tuning controller doesn't work well with this kind of disturbance.



(a)RBFNN

(b) PI

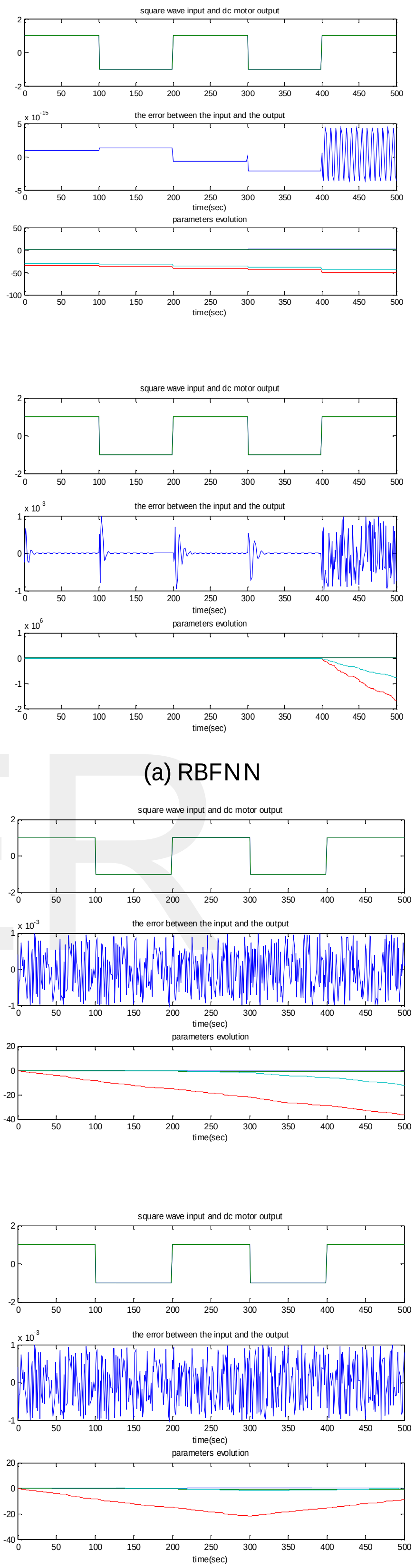
Fig. 5: Tracking trajectory for self tuning DC motor and the error signal



(a)RBFNN

(b) PI

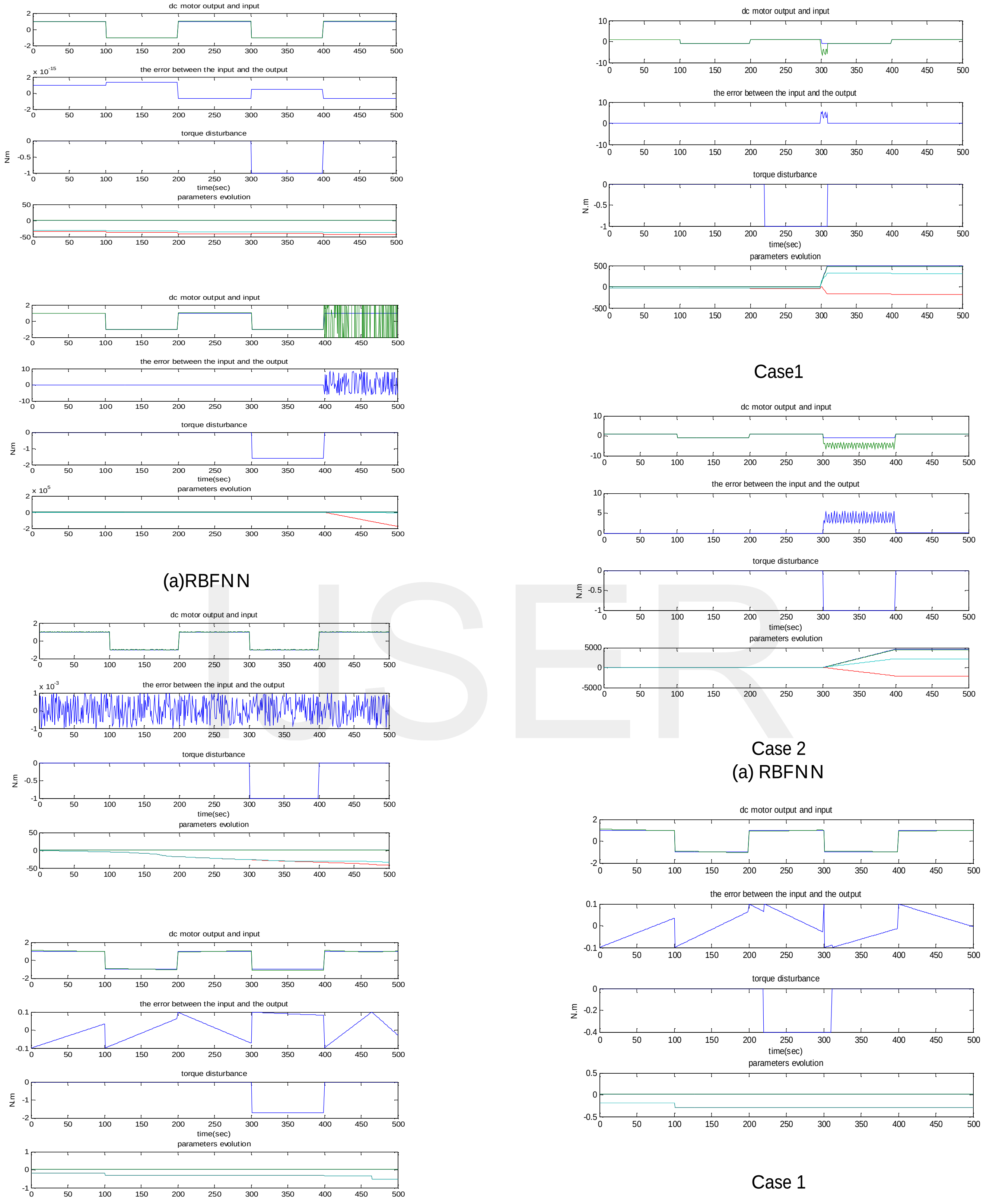
Fig. 6: The effect of variance for the armature resistance at $t \geq 300$ at (a) $R = 10\Omega$ and $R = 14.9\Omega$ (b) $R = 15\Omega$ and $R = 16\Omega$



(a) RBFNN

(b) PI

Fig. 7: The effect of variance for the rotor inertia at $t \geq 300$ (a) $J = 0.07$ and $J = 0.09\ Kg.m^2$ (b) $J = 0.06$ and $J = 1\ Kg.m^2$



(a)RBFNN

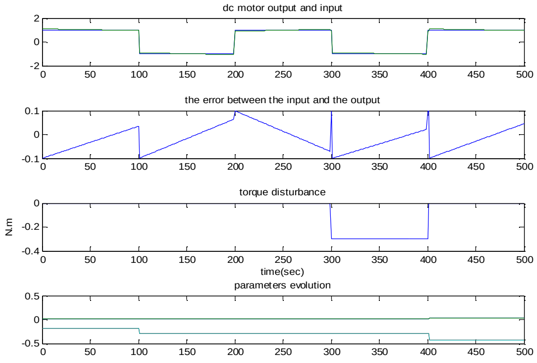
(b)PI

Case1

Case 2
(a) RBFNN

Case 1

Fig. 8: The effect of variance for the torque disturbance at $300 < t < 400$ (a) $Td = -1$ and $Td = -1.6\ N.m$ (b) $Td = -1$ and $Td = -1.7\ N.m$



Case 2
(b) PI
Fig. 9: The effect of disturbances synchronized on the speed of self tuning DC motor system.

3.2 The flexible transmission system

The flexible transmission system consists of three horizontal pulleys connected by two elastic belts. The schematic diagram and the photo of the system are shown, respectively, in Fig.10. The first pulley is driven by a dc motor whose position is controlled by local feedback. Since the dynamic of this feedback loop is much faster than that of the mechanical parts, it can be neglected in the analysis of the system. The objective is to control the position of the third pulley which may be loaded with small disks. The system input is the reference for the axis position of the first pulley. A PC is used to control the system. The system has a pure time delay equal to two sampling periods and an unstable zero.

The discrete-time plant is described by the following transfer operator:

$$H\left(q^{-1}\right)=\frac{q^{-d}B\left(q^{-1}\right)}{A\left(q^{-1}\right)}$$

Where q^{-1} is the backward shift operator, d is the plant pure time delay and

$$B(q^{-1}) = b_1 q^{-1} + ... + b_{nB} q^{-nB}$$
$$A(q^{-1}) = 1 + a_1 q^{-1} + ... + a_{nA} q^{-nA}$$

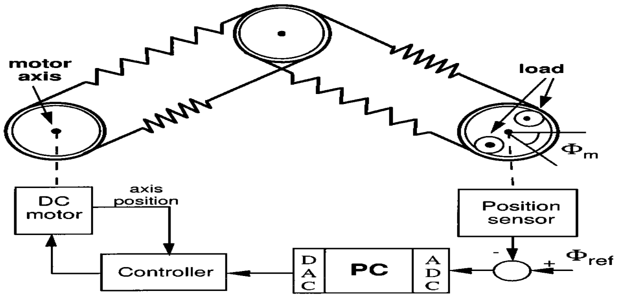
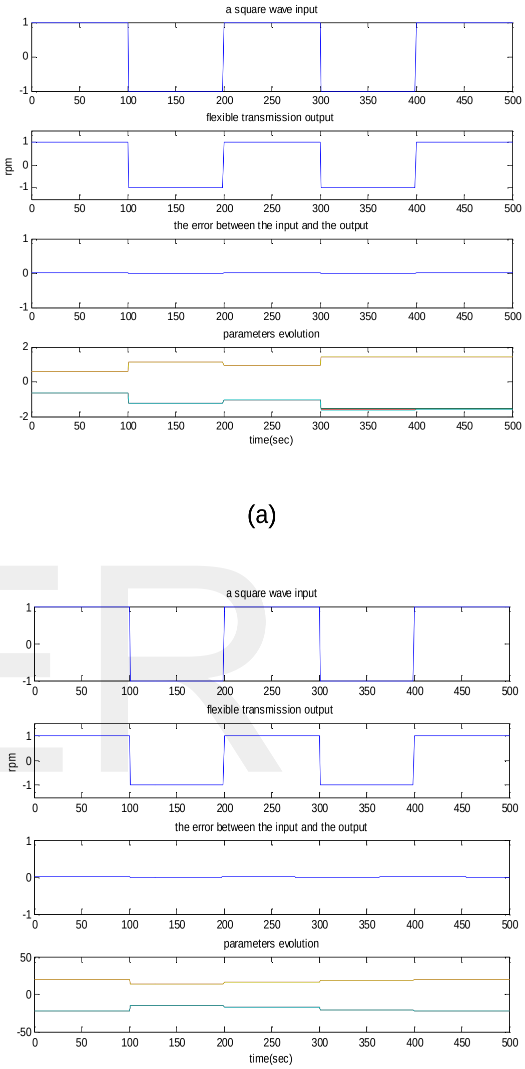


Fig. 10: Schematic diagram of the flexible transmission

Fig. 11 shows the efficient response of the self tuning flexible transmission adaptive control for the square wave excitation (reference) signal, the error signal which the variance between the reference signal and the output system also, the model parameters growth at the no load case.

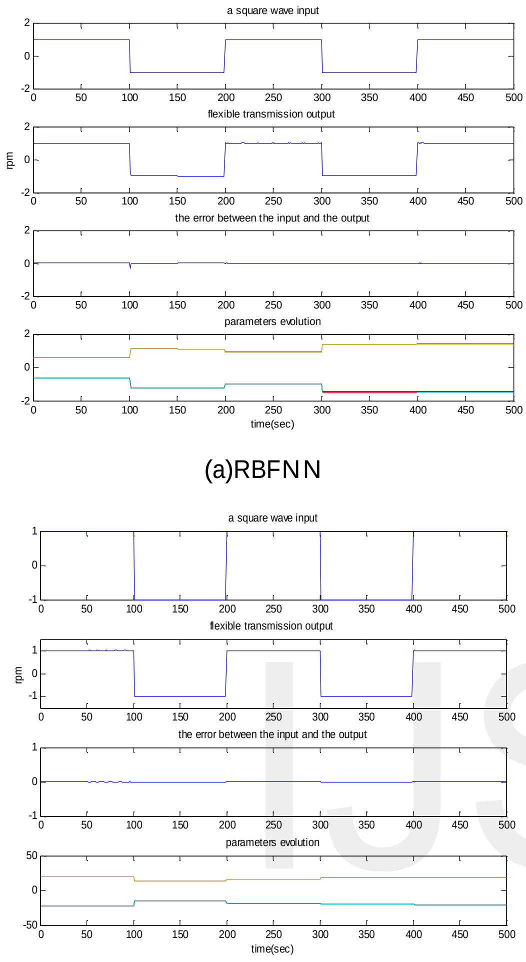


(a)

(b)

Fig. 11: Simulation Results for self-tuning flexible transmission system for no-load case

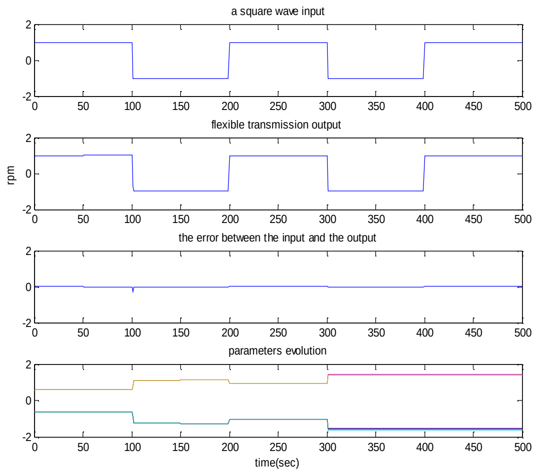
The self-tuning flexible transmission system is started without load on the third pulley and at the instants 150s and 350s 50% of the total load is added on the third pulley. Therefore the system without load becomes full loaded in two stages. Fig. 12 shows very good performances in tracking. The control system with the same synthesis parameters as well as the same reference signal is again considered. The plant is initially full loaded case and it passes to unloaded case in two stages (at 50s and 150s). The results illustrated in Fig. 13 show the good tracking performances of the system; likewise Fig. 14 that shows the transition from full load case to unload case direct (at 300s).



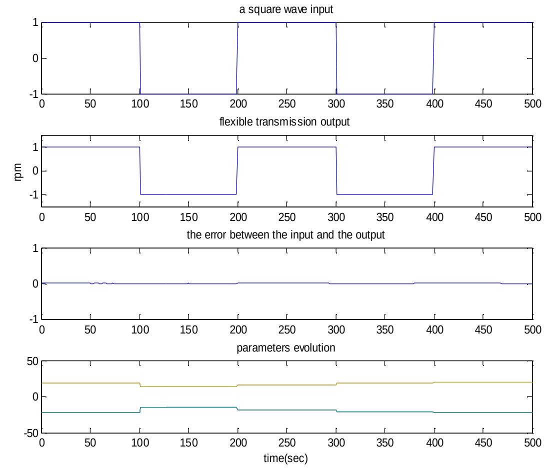
(a)RBFNN

(b)PI

Fig. 12: Simulation Results for self-tuning flexible transmission system for change load ($0\% \rightarrow 100\%$ in two stages at 150s, 350s).

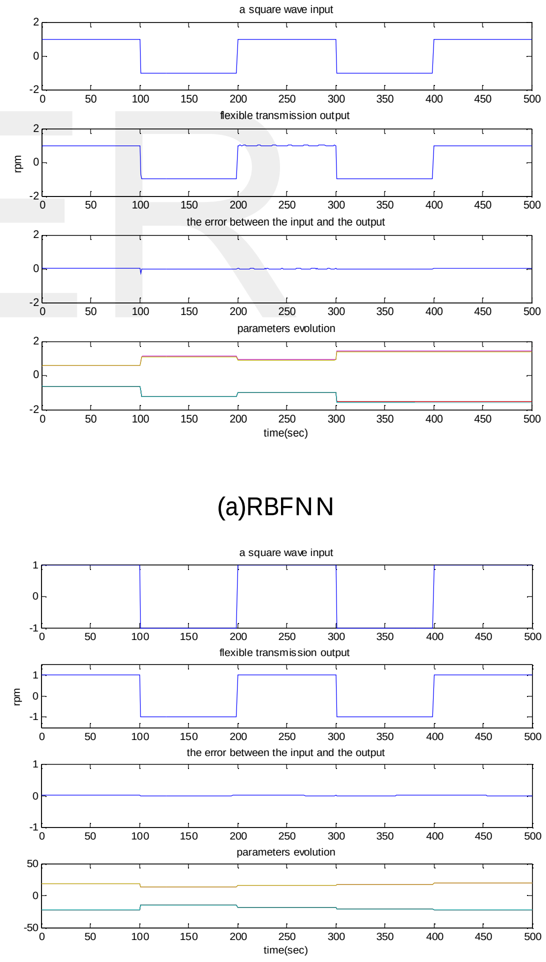


(a)RBFNN



(b) PI

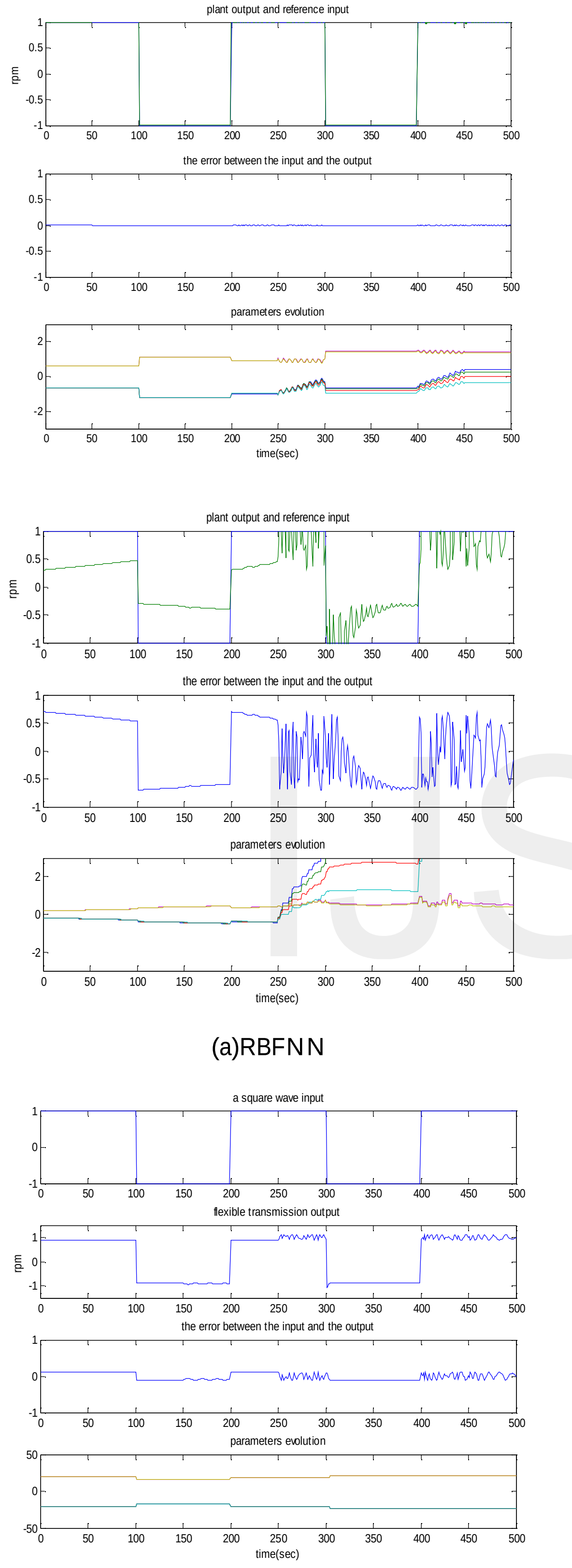
Fig. 13: Simulation Results for self-tuning flexible transmission system for change load ($100\% \rightarrow 0\%$ in two stages at 50s, 150s).



(a)RBFNN

(b) PI

Fig. 14: Simulation Results for self-tuning flexible transmission system for change load ($100\% \rightarrow 0\%$ at 300s).



(a)RBFNN

(b) PI

Fig. 15: Simulation Results for self-tuning flexible transmission system for $(0\% \rightarrow 100\%)$ case with output disturbance (e.g.(a) $0.05\sin(t)$ and $0.5\sin(t)$ (b) $0.05\sin(t)$)

Then, at the changed load case from unloaded state to full load state in two stages that previously mentioned, we suffix the sine wave signal to the position output of the system at specific time period $250 \le t \le 300$ and $400 \le t \le 450$ in fig .15. In summary, we see that the output of self-tuning regulator (STR) structure mimics the track accurately at first status but another trajectory misses when noise enlarged lightly.

4. Conclusions

A direct STR type of adaptive control is introduced and makes use of the RBFNN. Training algorithm is developed for the new STR. The proposed method is a very simple structure that updates itself online. The exact model of the plant needs not to be known and just the estimates are enough to drive the RBFNN as the process inverse. In the proposed technique the inverse of the plant is not obtained by another filter that might be the unstable inverse of the non-minimum phase plant, but modeled by RBFNN that only requires the estimate of the zeroes of the plant. The restrictions for the non-minimum phase and unstable linear plant are relaxed, thus outperforming others of its kind. Simulation results depict satisfactory tracking behavior for both minimum and non-minimum phase linear plants.

The proposed STR structure is tested by exposed to different disturbances in used plants. The adaptive self-tuning regulator introduces a good solution for control the linear plant even if the model meets a different individual disturbances or synchronous disturbances.

The RBFNN is a fast neural network compared with others type due to using least mean squares principle as training algorithm. Its structure has 2 neurons in hidden layer. At last, the proposed algorithm is successfully verified using simulations for both minimum and non-minimum phase of linear plants.

Appendix A

The parameter of the DC motor

Emf constant Kb=.015

Friction coefficient= 0.2 $N.m/rad/\sec$

Moment of inertia J= 0.02 $Kg.m^2$

Armature resistance R= 2.0 Ohms

Armature inductance L= 0.5 Henrys

Torque constant Km=.015

5. References

[1] K. J. Astrom and B.Wittenmark. Adaptive Control. Addison Wesley, 1995.

[2] Won Seok Oh; Kim Sol ; Kyu Min Cho ; Kyungsang Yoo ; Young Tae Kim. Self-tuning speed controller for induction motor drives. Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), 2012 International Symposium.

[3] Enzeng Dong, Shuxiang Guo, Xichuan Lin, Xiaoqiong Li, Yunliang Wang A neural network-based self-tuning PID controller of an autonomous underwater vehicle, Mechatronics and Automation (ICMA), 2012 International Conference on 5-8 Aug. 2012, pp 898 – 903.

[4] Zhao Ximei , Sun Xianfeng. Neural-Network-Based Self-Tuning. PI Controller for Permanent. Magnet Synchronous Motor, Electrical Machines and Systems (ICEMS), 2011 International Conference

[5] Rahmouni, A.; Lachiver, G. Optimal speed tracking control of induction motor using artificial intelligence techniques . Power Electronics Specialist Conference, 2003. PESC '03. 2003 IEEE 34th Annual.

[6] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical system using neural networks. IEEE Transactions on Neural Networks, 1:4–27, 1990.

[7] Kumar Rajesh, A.K.Ray. Artificial neural network modeling and control of Retention process in the wet end, International Journal of Information Technology and Knowledge Management July-December 2010, Volume 2, No. 2, pp. 259-264.

[8] S. Haykin. Neural Networks:A Comprehensive Foundation II. Macmillan / IEEE Press, 1994, 1999.

[9] Syed S., Hhssain A.,Muhamad M. Radial Basis Functions Neural Network Based Self-Tuning Regulator. WSEAS Transaction On Systems, Issue 9, Volume 3, November 2004.

[10] Vali U.,M .Yasir. Multiple Layer Perceptron for Direct Inverse Control of a Nonlinear System. Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference.

[11] M.A. Elsharkawi, Siri weerasooriya. Development and implementation of self-tuning tracking controller for DC motors. IEEE Transactions on Energy Conversion, Vol.5,No.1, March 1990.

[12] Siri Weerasooriya, M. A. El-Sharkawi. Identification and control of a dc motor using back-propagation neural networks. IEEE Transactions on Energy Conversion, Vol. 6, No.4, December 1991.

[13] S.E. Gaber; H.A. Yousef, Efficiency optimized speed control of DC motors based on self tuning regulator. Industrial Electronics, 1993. Conference Proceedings, ISIE'93 - Budapest., IEEE International Symposium.

[14] K.sabahi. Application of ANN Technique for DC-Motor Control by Using FEL Approaches. pp.131-134, 2011 Fifth International Conference on Genetic and Evolutionary Computing, 2011.

[15] A.K.Pal, I.Naskar. Design of Self-Tuning Fuzzy PI controller in LABVIEW for Control of a Real Time Process. International Journal of Electronics and Computer Science Engineering. Volume 2, Number 2, P.P 538-545, 2013.

[16] Kota, B.V.S Goud. Fuzzy PID Control for Networked Control System of DC Motor with Random Design. International Journal of Computer Applications (0975 – 8887) Volume 52 – No. 7, August 2012.

[17] Fawaz, Abdul-Kareem, Thair. Speed Control of Separately Excited D.C. Motor using Self-Tuned Parameters of PID Controller. Tikrit Journal of Engineering Sciences Vol.20, No.1, March 2013, (1-9).

[18] Saad Zaghlul. Tuning PID Controller by Neural Network for Robot Manipulator Trajectory Tracking Al-Khwarizmi Engineering Journal, Vol. 8, No. 2, P.P. 91-28 (2013).

[19] Alfonso, Víctor, Manuel, Hugo, Edgar, Juvenal, Gilberto. A New Adaptive Self-Tuning Fourier Coefficients Algorithm for Periodic Torque Ripple Minimization in Permanent Magnet Synchronous Motors (PMSM). Sensors, Vol. 13, Pages 3831-3847, 2013.

[20] Karimi A., Landau I.D. "Robust adaptive control of a flexible transmission System using multiple models". IEEE Transactions on Control Systems Technology, vol. 8, no. 2, March 2000

[21] Oliver. Nonlinear System Identification From Classical Approaches To Neural Networks And Fuzzy Models,Springer-Verlage Berlin Heidelberg 2001.

[22] Mr. M.V.Sudarsan. Designing Of ANN Based Speed Controller For Phase Controlled DC Motor. International Journal of Engineering Science and Technology (IJEST), Vol. 3 No. 7 July 2011.