Inverse Control for a Magnetic Levitation System Using the Neural Network

Abd-El meged Mohamed[1], Gaber Elsady[2], Ashraf Hemeda[3], Asmaa Fawzy[4*]

Abstract—For overcomes several shortcomings of the inverse control design to controlling nonlinear systems using the neural networks as the controller based self tuning regulator. The magnetic levitation parameters are estimated online 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. The proposed algorithm is successfully verified using simulations. Then, this paper compared its result with the outcome of using proportional-plus-integral feedback (PI) self tuning regulator.

Keywords — STR, RBFNN, ARX, LMS, magnetic levitation

1. INTRODUCTION

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]. In many practical applications it is, however, difficult to determine the parameters of the controller, since the dynamics of the process and its disturbances are unknown. The parameters of the process thus have to be estimated. For stationary processes it is possible to determine the unknown parameters through identification. The experiments and their evaluations can, however, be rather time consuming. It is thus desirable to have a regulator which tunes its parameters on-line.

Enzeng 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 [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 (magnetic levitation) so the output tracks the reference input.

Sabahi [10] 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 [11] 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 [12] 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. Saad [13] 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.

Vali U. and M Yasir Amir [14] proposed Direct Inverse Control scheme of a simple nonlinear dynamic system. The idea is to train a neural network as an inverse of plant so as to cancel out the plant dynamics and to make plant follow the reference input.

A lot of research effort in control system field has been focused on the control of a Magnetic Levitation System (MLS). They are widely used in various fields such as frictionless bearings, high-speed Maglev passenger trains, levitation of wind tunnel models, vibration isolation of sensitive machinery, levitation of molten metal in induction furnaces and the levitation of metal slabs during manufacture. MLS are generally highly nonlinear and open loop unstable systems. The inherent nonlinearities of systems make the modeling and control problems very challenging.

Sakslli [15] compared the closed loop control performance of interval type-2 fuzzy PID controller with the type-1 fuzzy PID and conventional PID controller's counterparts for the Magnetic Levitation Plant. Dragan [16] presented analysis of different training types for nonlinear autoregressive neural network, used for simulation of magnetic levitation system. It is verified that NARX neural network can be successfully used to simulate real magnetic levitation system if suitable training procedure is chosen, and the best two training types, obtained from experimental results. suster [17] Designed control algorithm

1engineering faculty, Aswan university, Egypt; 2engineering faculty, Assiut university, Egypt;3,4energy engineering faculty, Aswan university, Egypt
*Correspondence to (e-mail: rashwan_asmaa@yahoo.com).

together with simulation model of the Magnetic levitation is implemented into control structure with purpose of control on steady state defined by a reference trajectory, which is verified in Matlab/Simulink language.

In this paper a technique is proposed that gives a good control for the magnetic levitation. An online control algorithm is structured using the radial basis function neural network (RBFNN). The magnetic levitation 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 magnetic levitation and thus the output tracks the reference trajectory. This scheme is exposed to several types of disturbances for wide range of operating conditions.

2. System Description

Magnetic levitation system considered in the current analysis is consisting of a ferromagnetic ball suspended in a voltage-controlled magnetic field. Fig. 1 shows the schematic diagram of magnetic levitation system.

Coil acts as electromagnetic actuator, while an optoelectronic sensor determines the position of the ferromagnetic ball. By regulating the electric current in the circuit through a controller, the electromagnetic force can be adjusted to be equal to the weight of the steel ball, thus the ball will levitate in an equilibrium state. But it is a nonlinear, open loop, unstable system that demands a good dynamic model and a stabilized controller [18].

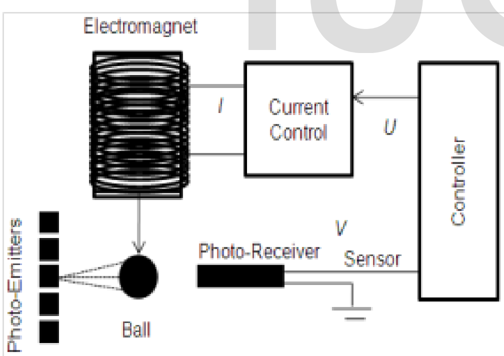


Fig. 1. Schematic Diagram of Magnetic Levitation System.

Dynamic behavior of magnetic levitation system can be modeled by the study of electromagnetic and mechanical sub systems. Electromagnetic force produced by current is given by the Kirchhoff's voltage law[18][19];

$$E(t) = V_R + V_L = iR + \frac{dL(x)i}{dt} \qquad (1)$$

Where $E(t)$ is applied voltage, $i(t)$ is current in the coil of electromagnet, R is coil's resistance and L is coil's inductance.

In mechanical part, free body diagram of ferromagnetic ball suspended by balancing the electromagnetic force $f_{em}(x,i)$ and gravitational force f_g. Net force f_{net} acting on the ball is given by Newton's 3rd law of motion while neglecting friction, drag force of the air etc.

$$f_{net} = f_g - f_{em}$$

$$m\ddot{x} = mg - C(\frac{i}{x})^2 \qquad (2)$$

Where m is mass of ball, x is position of the ball, g is gravitational constant and C is magnetic force constant.

Equation 1 indicates that $L(x)$ is a nonlinear function of balls position x. Various approximations have been used for determination of inductance for a magnetic levitation system. If we take the approximation that inductance varies with the inverse of ball's position, that is

$$L(x) = L_1 + \frac{L_0 x_0}{x} \qquad (3)$$

Where L_1 is the constant inductance of the coil in the absence of ball, L_0 is the additional inductance contributed by the presence of the ball, x_0 is the equilibrium position. Substituting equation (3) into (1) results in

$$e_m(t) = iR + L\frac{di}{dt} - (\frac{L_0 x_0 i}{x^2})\frac{dx}{dt} \qquad (4)$$

A conservation of energy argument shows that $C = L_0 x_0 / 2$. The ball's velocity is v so;

$$v = \frac{dx}{dt}$$

Combining the above with $x_1 = x$, $x_2 = v$, $x_3 = i$ and $u = e_m$, the system equations in state-space form are [20]

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = g - \frac{C}{m}(\frac{x_3}{x_1})^2 \qquad (5)$$

$$\dot{x}_3 = -\frac{R}{L}x_3 + \frac{2C}{L}(\frac{x_2 x_3}{x_1^2}) + \frac{u}{L}$$

The system was linearized around a point $x_1 = x_{10}$, which results in state vector as;

$$X_0 = [x_{10} \quad x_{20} \quad x_{30}]^T \qquad (6)$$

At equilibrium, time rate derivative of x must be equal to zero i.e. $x_{20} = 0$. Also equilibrium current can be evaluated from equation (2) and it must satisfy the following condition;

$$x_{30} = x_{10}\sqrt{\frac{gm}{C}} \qquad (7)$$

Thus we can write the linearized model in state space form as under;

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \dfrac{2Cx_{30}^2}{mx_{10}^3} & 0 & -2\dfrac{Cx_{30}}{mx_{10}^2} \\ 0 & 2\dfrac{Cx_{30}}{Lx_{10}^2} & -\dfrac{R}{L} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \dfrac{1}{L} \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \qquad (8)$$

The linear system described in equation 8 is unstable and controllable. Therefore, the pole placement method is used to determine a value of $K \in R^{1\times n}$ that will produce a desired set of closed-loop poles. Ackermann's formula can be used for pole placement. Ackermann's formula (1972) is a direct evaluation method. It is only applicable to SISO systems [21].

3. SELF TUNING REGULATOR WITH NEURAL NETWORK

Fig.2 is the proposed structure. Autoregressive with exogenous input (ARX) is used to identify the dc motor and found the model. The model coefficients are updated online depending on magnetic levitation parameters variation. These coefficients are fed the weight update block which trains the controller whether RBFNN or PI controller using the least mean square LMS algorithm.

Fig.2: Proposed self-tuning magnetic levitation regulator structure

3.1 ARX model

The process is modeled by an ARX model [22], whose output is given by

$$y(t) = \sum_{i=1}^{n} a_i y(t-i) + \sum_{j=1}^{m} b_j x(t-j) \qquad (9)$$

Or in terms of q^{-1} operator

$$y(t) = \frac{B(q^{-1})}{A(q^{-1})} q^{-d} x(t) \qquad (10)$$

3.2 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 nodes 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 $w_1, \dots, w_{n_0}$.

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, 20]

$$\phi(d) = \exp(-\tfrac{1}{2} d^2) = \exp(-\tfrac{1}{2} \left\| r(t) - c_i \right\|_{\Sigma}^2) \qquad (11)$$

$$d = \sqrt{\left(\frac{r(t)-c_1}{\beta_1}\right)^2 + \dots + \left(\frac{r(t)-c_{n_0}}{\beta_2}\right)^2} \qquad (12)$$

c_i is the center of i^{th} hidden layer node where $i = 1,2,\dots,n_0$, 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

$$x(t) = \sum_{i=1}^{n_0} w_i \phi(\left\| r(t) - c_i \right\|_{\Sigma}) \qquad (13)$$

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.

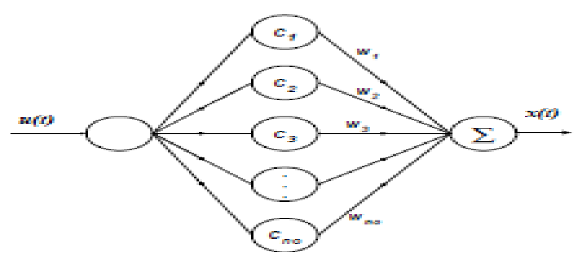


Fig. 3: A general RBF network

3.3 parameters estimation for self-tuning of RBFNN/PI

The parameters of the magnetic levitation model are estimated online and are used to update the coefficients of the controller (weights of the RBFNN / parameters of PI). The weight/coefficient update equations are derived based on a recursive scheme (least mean squares principle). This previous parameters are updated by minimizing the performance index I given by [9]

$$I = \tfrac{1}{2} e^2(t) \qquad (14)$$

ISSN 2229-5518

$$e(t) = r(t) - x(t) \qquad (15)$$

Where $r(t)$ is the reference input signal and $x(t)$ is the output position of the ball of the magnetic levitation model. The coefficients of the ARX model and the weights of the RBFNN/parameters of the PI are updated in the negative direction of the gradient as,

$$\theta(K+1) = \theta(K) - \mu \frac{\partial I}{\partial \theta(K)} \qquad (16)$$

$$W(K+1) = W(K) - \mu \frac{\partial I}{\partial W(K)} \qquad (17)$$

$$PI(K+1) = PI(K) - \mu \frac{\partial I}{\partial PI(K)} \qquad (18)$$

Where $\theta = \begin{bmatrix} a_1 & \ldots & a_n & b_1 & \ldots & b_m \end{bmatrix}$ is the parameter vector, $W = \begin{bmatrix} w_1 & w_2 & \ldots & w_{n_0} \end{bmatrix}$ is the weight vector for RBFNN, $PI = \begin{bmatrix} k_p & k_i \end{bmatrix}$ vector for the parameters proportional-integral (PI) 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 $\psi(t) = \begin{pmatrix} -x(t-1) & \ldots & -x(t-n) & e\,(t-d) & \ldots & e\,(t-m-d) \end{pmatrix}$ and finding partial derivatives.

$$\frac{\partial I}{\partial \theta} = \frac{1}{2} \frac{\partial e^2(t)}{\partial \theta} \qquad (19)$$

$$= e(t) \frac{\partial}{\partial \theta} (r(t) - w(t)) \qquad (20)$$

$$= e(t) \frac{\partial}{\partial \theta} \begin{pmatrix} r(t) - \left(-a_1 q^{-1} - \ldots a_n q^{-n}\right) x(t) \\ -\left(b_1 q^{-1} + \ldots + b_m q^{-m}\right) q^{-d} e_m(t) \end{pmatrix} \qquad (21)$$

$$\frac{\partial I}{\partial \theta} = -e(t)\psi(t) \qquad (22)$$

The final parameter update equation will be,

$$\theta(K+1) = \theta(K) + \mu e(t)\psi(t) \qquad (23)$$

The partial derivatives for the weights are derived as follows,

$$\frac{\partial I}{\partial W} = \frac{1}{2} \frac{\partial e^2(t)}{\partial W} \qquad (24)$$

$$= e(t) \frac{\partial}{\partial W} \left(r(t) - \frac{B(q^{-1})}{A(q^{-1})} q^{-d} e_m(t) \right) \qquad (25)$$

$$\frac{\partial I}{\partial W} = -e(t) B(q^{-1}) q^{-d} \phi(t) \qquad (26)$$

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,

$$PI(K+1) = PI(K) + \mu e(t) B(q^{-1}) q^{-d} \left(-e_m(t-1) + (t_s + 1) e_m(t)\right) \qquad (27)$$

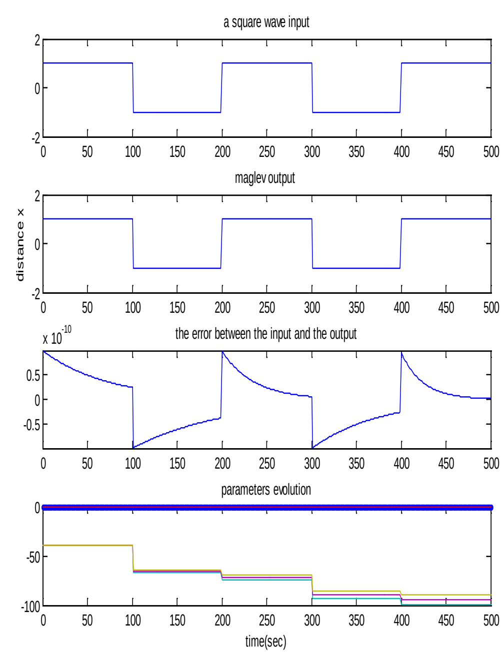
t_s is the sample time.

4. SIMULATION RESULTS

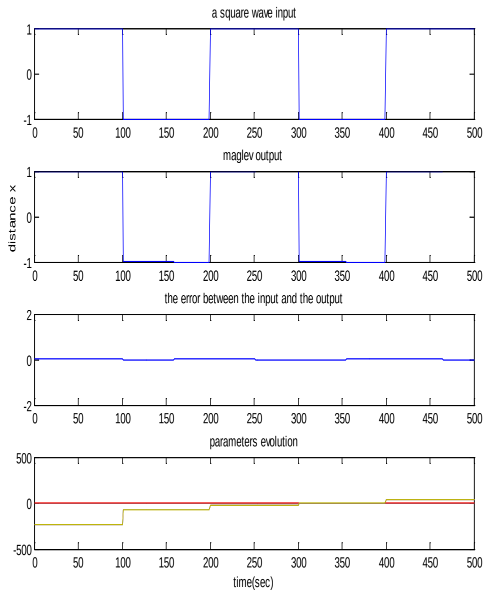
The offered self-tuning regulator (ST R) structure designed to overcome the nonlinearity of the system and to improve the system's response performance. The magnetic suspension has many uncertainties, e.g. magnetic field distribution, variation of coil inductance, etc.

The performance was compared in terms of the response to square wave input fig.4. The magnetic levitation system consists of a steel ball of mass 50 g. a good controller should be able to control the position of the levitated ball even when the weight of the ball changes fig.5. We notice amounts of parameters of the ARX model increase directly proportional with growth of mass with the PI against radial basis self tuning and maintains the track relatively.

Fig.6 interprets that there isn't modification in the response of the system as well as values of model parameters when the coil resistance varies. Likewise, the parameters values expand when the inductance of the electromagnet increases when using the RBFNN; But with the PI affects on the trajectory and the parameters value that explained in fig.7. Fig.8, the sinusoidal output doesn't follow the track exactly but PI case is the worst. Finally, the controller exposed to noise at the position output at specific time period $250 \le t \le 300$ and the output of self-tuning regulator (STR) structure follows the track precisely at the RBFNN case but the output of PI self-tuning regulator (STR) structure pursue the track with some error fig.9.



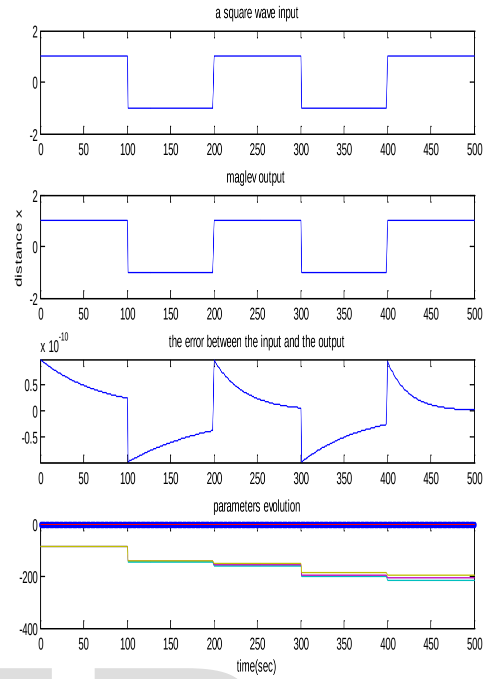
(a) RBFNN



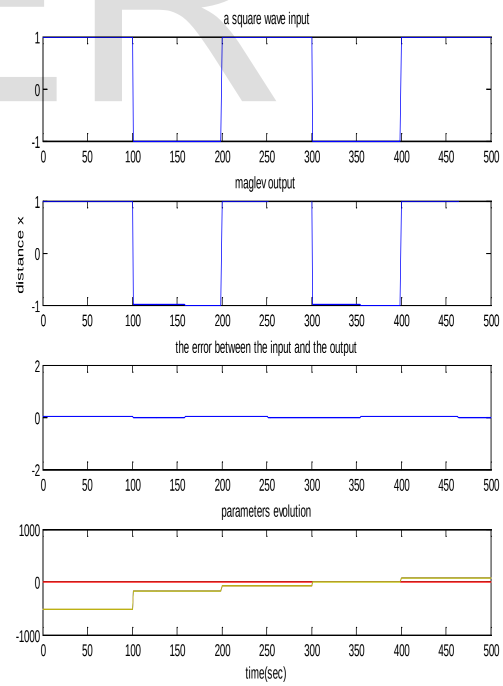
(b)PI

Fig.4: the output of self-tuning magnetic levitation system for a square wave reference

0



(a) RPFNN

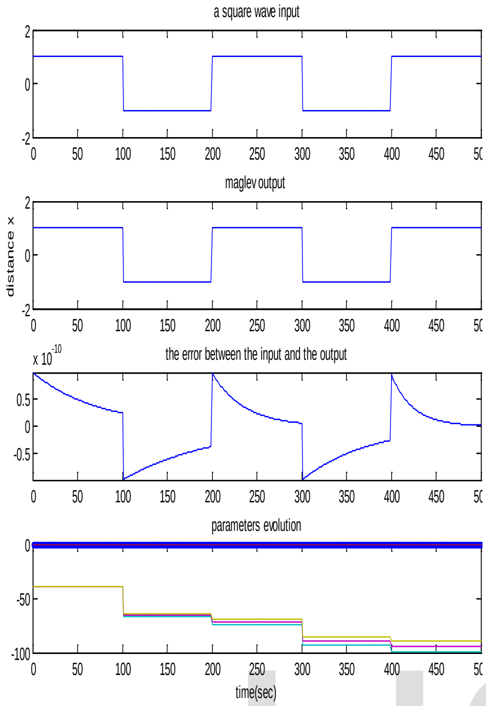


(b)PI

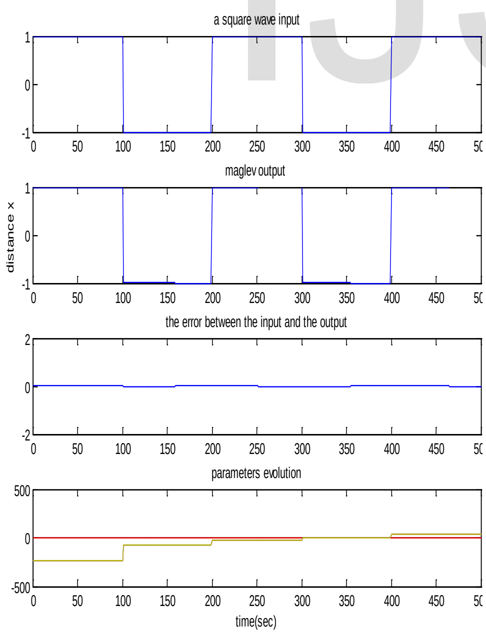
Fig.5: The effect of variance for the ball mass (a) $m = .11kg$ (b) $m = .11kg$

ISSN 2229-5518

0



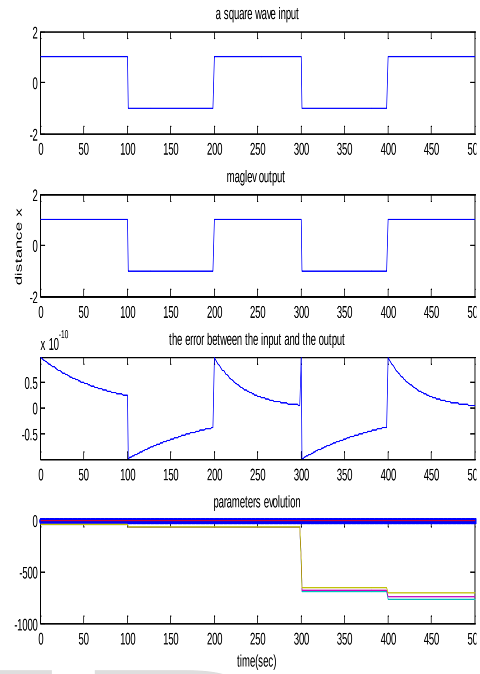
(a)RBFNN



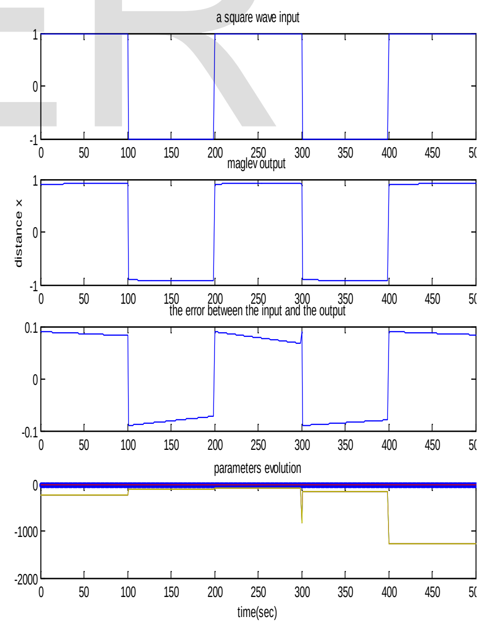
(b) PI

Fig.6: The effect of variance for the coil resistance $R = 3\Omega$ at $t \geq 300$

0



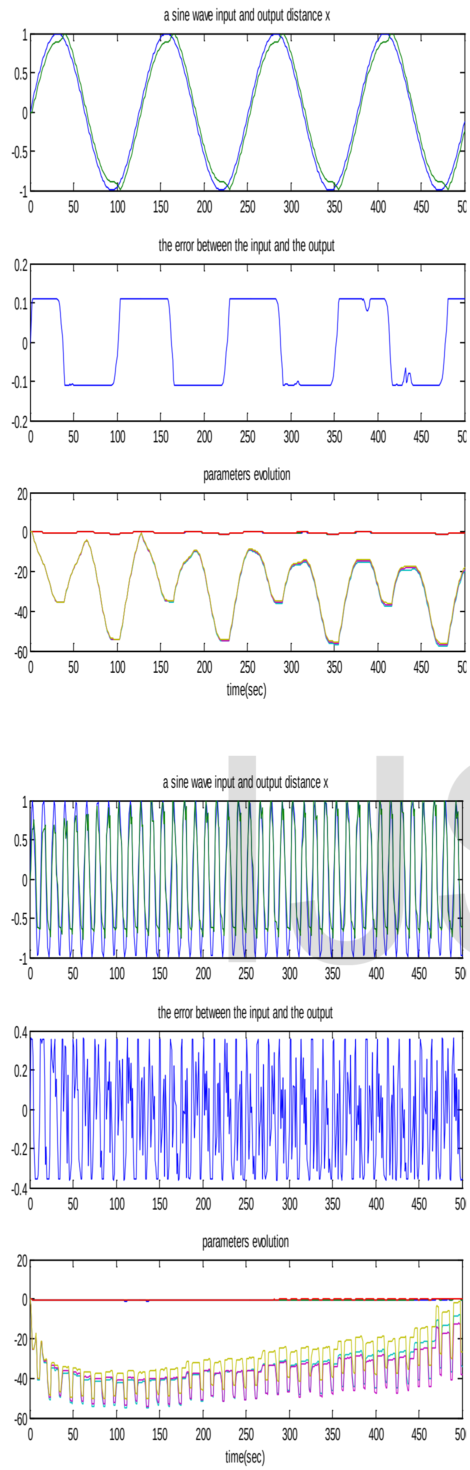
(a) RBFNN



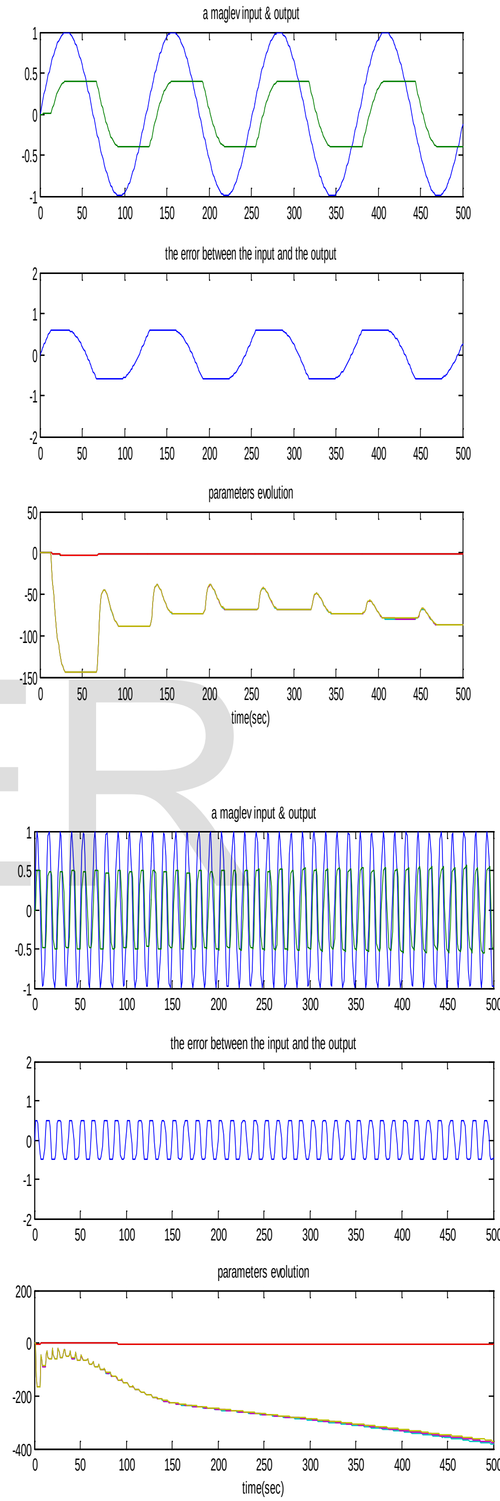
(b)PI

Fig.7:The effect of variance for the inductance of the electromagnet at $t \geq 300$ (a) $L = .1H$ (b) $L = .05H$

ISSN 2229-5518



(a) RBFNN

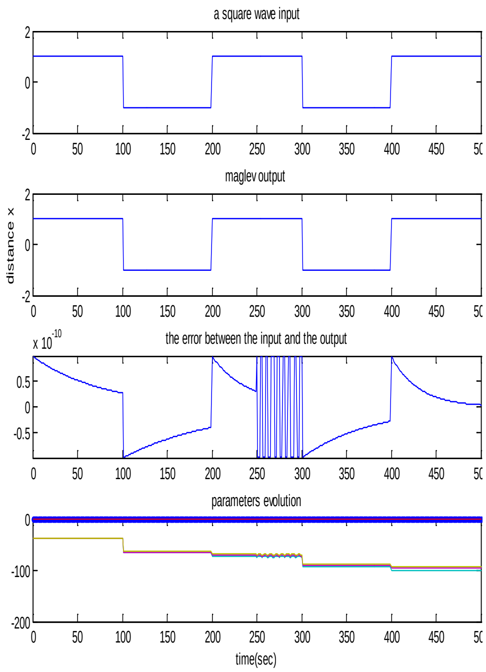


(b) PI

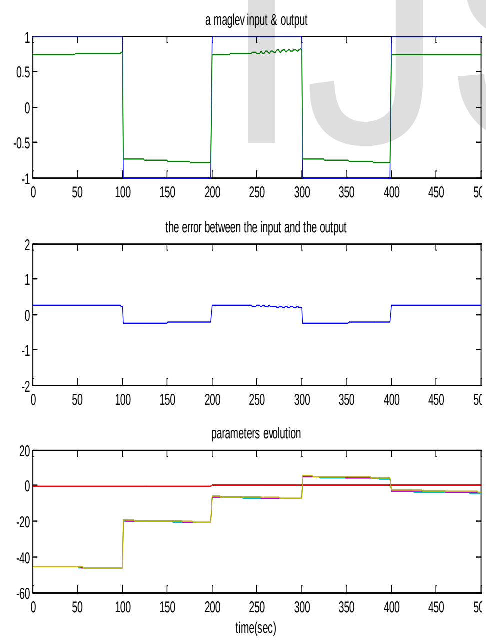
Fig.8: the output of self-tuning magnetic levitation system for a sine wave reference with different frequency

ISSN 2229-5518

0



(a) RBFNN



(b) PI

Fig.9: Simulation Results for self-tuning magnetic levitation system output disturbance at $250 \le t \le 300$ (a) $0.05\sin(t)$ (b) $0.01\sin(t)$

5. CONCLUSIONS

This paper introduces a very simple structure for control the magnetic levitation that updates itself online. MLS are generally highly nonlinear and open loop unstable systems. The exact model of the MLS needs not to be known and just the estimates are enough to drive the RBFNN as the process inverse.

The adaptive self-tuning regulator introduces a good solution for control the maglev even if the model meets different 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.

6. 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] 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.

[11] 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.

[12] 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.

[13] 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).

[14] 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.

[15] Ahmet Sakalli, Tufan Kumbasar, Engin Yesil, Hani Hagras " Analysis of the Performances of Type-1, Self-Tuning Type-1 and Interval Type-2 Fuzzy PID Controllers on the Magnetic Levitation System" 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) July 6-11, 2014, Beijing, China

[16] Antić, Dragan; Milovanović, Miroslav; Nikolić, Saša; Milojković, Marko; Perić, Staniša "Simulation Model of Magnetic Levitation Based on NARX Neural Networks". International Journal of Intelligent Systems & Applications; Apr2013, Vol. 5 Issue 5, p25.

[17] P. Šuster and A. Jadlovska. Modeling and control design of magnetic levitation system. Proceedings of 10th International Symposium Applied Machine Intelligence and Informatics, Herl'any, Slovakia, January, 26.-28., 2012, pp. 295-299.

[18] I. Ahmad and M. Akramjavaid. Nonlinear model and controller design for magnetic levitation system. Proceedings of 9th International Conference on Signal Processing, Robotics and Automation, University of Cambridge, United Kingdom, February, 20.-22., 2010, pp. 324-328

[19] YING-SHING SHIAO, "Design and Implementation of a Controller for a Magnetic Levitation System" Proc.Natl. Sci. Counc. ROC(D) Vol. 11, No. 2, 2001. pp. 88-94

[20] W. Barie and J. Chiasson, "Linear and Nonlinear state-space controllers for magnetic levitation," International Journal of Systems Science, vol. 27, number 11, pp. 1153-1163, 1996.

[21] Roland S. Advanced control engineering. first published 2001

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