Implementation of Adaptive Neural Networks Controller for NXT SCARA Robot System

Wessam M. F. Abouzaid
Electrical Engineering Department
Faculty of Engineering, Kafrelsheikh University
Kafrelsheikh, Egypt
eng_wessam_fikry1@yahoo.com

Elsayed. A. Sallam
Computers and Control Engineering Department
Faculty of Engineering, Tanta University
Tanta, Egypt
sallam@f-eng.tanta.edu.eg

Abstract—**Several neural network controllers for robotic manipulators have been developed during the last decades due to their capability to learn the dynamic properties and the improvements in the global stability of the system. In this paper, an adaptive neural controller has been designed with self learning to resolve the problems caused by using a classical controller. A comparison between the improved unsupervised adaptive neural network controller and the P controller for the NXT SCARA robot system is done, and the result shows the improvement of the self learning controller to track the determined trajectory of robotic automated controllers with uncertainties. Implementation and practical results were designed to guarantee online real-time.**

Keywords—Nonlinear MIMO Systems; ADALINE; trajectory design; NXT SCARA Model.

I. INTRODUCTION

Nonlinear control system consumes an increasingly essential placement in the region of process control engineering [1]. Robotic Manipulators are widely used in different fields of industry: for the purpose of saving time, effort, and sometimes life.For these reasons and due to the vast applications of robotic manipulators, the design of controllers to optimize the tracking and speed performance of robots has become a necessity and an important research area [2]. Therefore, the simple P controller implemented in a dynamic model of flexible manipulator. (Adaptive, NN, and fuzzy logic have been used for tip position control, as they are very effective, but more complex and difficult to design and analyze) in [3]. Henceforth, intelligent control has a special cerebrate on Neuro-control that can be used successfully to fix problematic real control that is usually nonlinear, noisy and complex. Four degrees of freedom are controlled by neural networks with fuzzy controller for driving trajectory robot and handing processing with certain job [4].

Also, neural networks are able to deal with nonlinearities and are used to improve an adaptive controller with unknown systems [5]. It is regarded as a powerful control technique due to its ability to learn and adapt [6]. By the 1980's, various adaptive approaches have been provided to a stable operation and asymptotic follow [7].

When designing a controller to control the trajectory tracking performance of a manipulator, some of the problems encountered during the controller design process are: the inverse kinematics and dynamics problems. There is always a difference between the conventional mathematical model used for the manipulator and the real robot which would generate a considerable error between the desired and actual trajectories. Therefore, a conventional model-based controller is ineffective in controlling the robot in real time. Each joint of the manipulator is controlled by individual controller[8], [9], [10]. This is because such methodologies do not count for the coupled interaction between the joints which result in the generation of coupling disturbance torques between the different joints of the manipulator [2].

Adaptive linear networks (ADALINEs) are usually among designs pertaining to files prediction. Using this method is not a lot program in order to forecast nonlinear technique dynamics because it offers gradual throughout convergence [11]. It presents that offer even fast training, small memory requirements, simple usage and shorter computational times [12].

This paper proposes a design and improvement adaptive MIMO (Multi Input Multi Output) with regard to real-time following management of manufacturing NXT SCARA (Selective Compliance Assembly Robot Arm), applying adaptive neural network. NXT means NEXT (Second Version of the Set of Lego Mindstorm) [13].The trajectory tracking controller problems of robot manipulators has been attracting widely attention [14]. The neural network will adjust the weights on line to enhance execution as it controls the NXT SCARA [15]. The designed controller is self-learning (Unsupervised) adaptive neural network based on modifying the ADALINE equation to concatenate between the inputs and outputs of the NXT SCARA system. The performance of this adaptive neural network based control strategy was compared with that of the conventional computed torque P

(Proportional)control method. The obtained simulation results proved the superiority of the designed adaptive neural network-based controllers in compensating for both structured and unstructured uncertainties and demonstrated the effectiveness of the self learning controller [2].

II. CONTROLLER DESIGN

Artificial neural system proved to be a helpful instrument in managing applications, for example, pattern recognition, signal processing, image processing and various complex controls and mapping problems, etc. Neural networks have been applied successfully to identify and control dynamic systems because of their learning capacity and ability to tolerate incorrect or noisy data [11]. However, the parallel distribution of the signals through the human neurons makes them capable of carrying out perceptual tasks such as face recognition much faster than the serially operated electronic circuits. Therefore, the strength of the human neurons lies in their parallel computing and signal distribution capabilities as well as their self-memorizing and learning characteristics [2].

One ANN (Adaline Neural Network) could be used instead of two P controllers. The second aim is to develop a neural network controller, which determines the correct control action to stabilize the system, and also can learn from experience [15].

A. ADALINE Library

The Adaline is used to approximate 'almost linear' functions.The Adaptive NN is developed by using the neural network toolbox. The particular obstruct diagram with the tool kit is shown in Fig. (1), that have an ADALINE and GMLP (Generalize multilayered perceptron) blocks of Simulink. These blocks posses same interface.So, the possibility of using these networks was quick and easy [15].
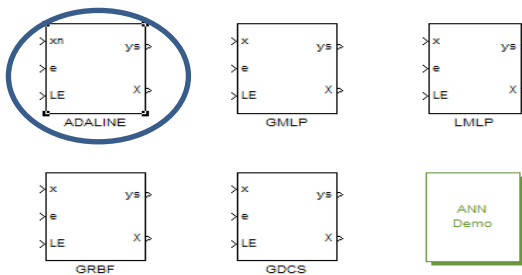
Fig.1. Block diagram of library toolbox of Adaptive Neural Network

ADALINE is one of the essential models utilized for information expectation. In such neural network, the present output system is subject to the past output and on both the current and past inputs. So, the ADALINE have output feedback andremembering input/output data. The low computational intricacy makes this technique proper for online

system to identify the real time adaptive control applications [11].

An ANN can form the organization of the data that gets within learning time. The adaptive NN operates in real time; its computation may be carried out in parallel [12]. Fig. (2) represents the inputs and outputs of ADALINE, the error computes as, $e = y - y_s$, e is error signal, y is adaptively output and y_s is the output of Adaline learned function.
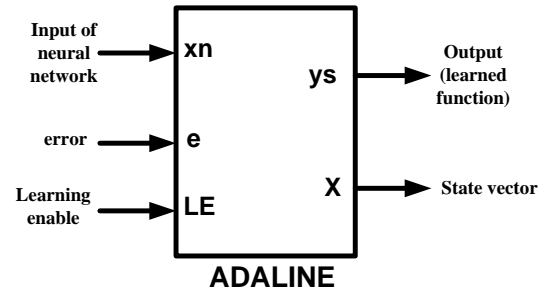

Fig.2. The inputs and outputs of ADALINE

The weights updated using gradient descent that minimizes the error squared among measurements and estimate values. It is used to adaptively approximate

$$y = W * X + k \qquad (1)$$

$$y_s = 0.1 * W_0 + W_s * X(t) \qquad (2)$$

X is a state vector, K is bias vector, and W_o is the initial value of weights.

The weights are updated according to the equation

$$W(t + T) = W(t) - \zeta * (\frac{dy_s}{dw}) * e(t) - \zeta * \gamma * W(t) \qquad (3)$$

Where $W(t)$ is weight matrix, $e(t)$ is an error vector at time t,

ζ is learning rate, γ is stablizing factor and $(\frac{dy_s}{dw})$ is jacobian matrix , T is sample time.

B. Unsupervised Online Learning Using ADALINE Toolbox

By modifying the ADALINE equation (3) of updating weights to connect the two inputs of NXT SCARA with the two outputs. The Simulink of unsupervised adaptive control is shown in Fig. 3.
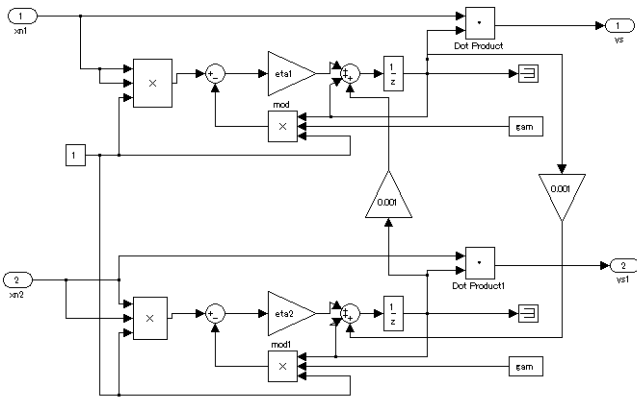
Fig.3. Simulink of unsupervised adaptive NN

Where x_{n1}, x_{n2} are the first and second inputs of Adaline controller; respectively, LE is learning enable equal 1, and y_s, y_{s1} are outputs of Adaline controller; respectively.

In unsupervised adaptive control, there is no target to learn about it, the error signal is connected to the inputs of the controller since the controller is trained on line as shown in Fig. 4. The block diagram, Fig.5. showed the system controlled by Adaline controller.
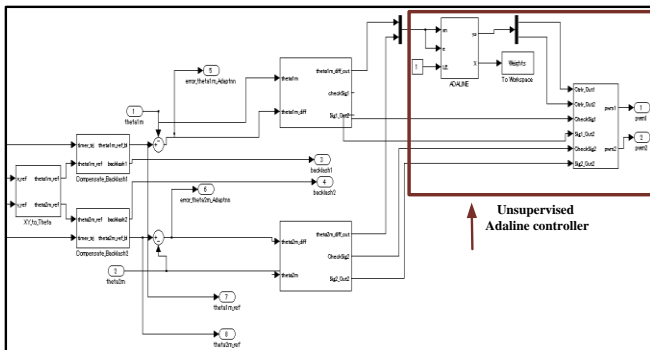

Fig.4. The Simulink module of system with unsupervised Adaline controller
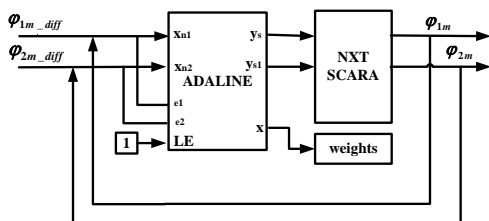

Fig.5. Block diagram of controlled system

We used Matlab m-files, codes and Simulink modules to control the NXT SCARA system by Adaline neural network toolbox in unsupervised method. The parameter block of Adaline is shown in Fig. 6 with learning rate of 0.05.
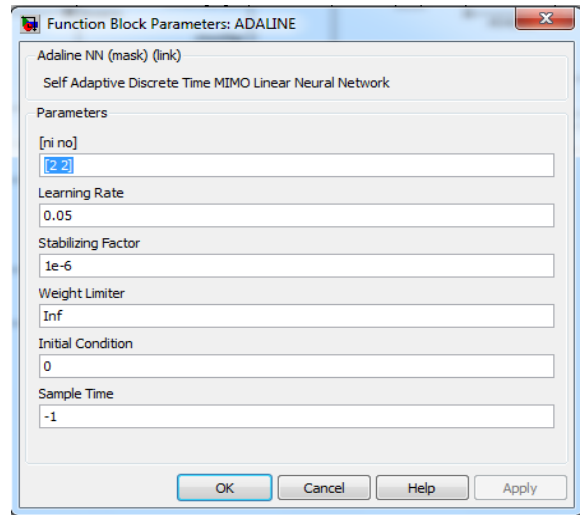

Fig.6. The parameters of adaptive controller

And the training weights are presented in Fig. 7, the weights are changed and updating to get the final values and stable weights as w1=0, w2=60.4, w3=-18.77, w4=0, w5=-18.77, and w6=68.6.
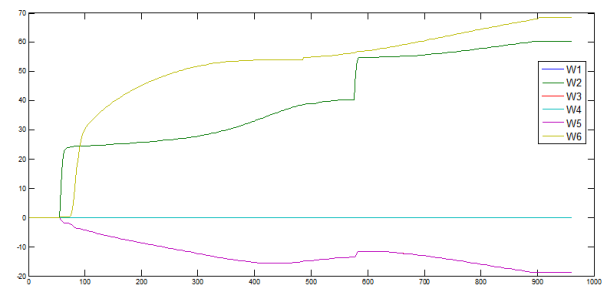

Fig.7. The training weights of unsupervised control

III. NXT SCARA SYSTEM

The targets of the modeling are creating a unified device model with information about that structure and the functionality of this device. There is a kinematic structure that defines the appearance and movement capabilities of the robot[16]. The system consists of 3 DC motors for link1, link2, and pen; respectively. The first and second motors are used to move links for drawing the required trajectory as shown in, the units of the axis are centimeters in x and y, the x axis equal to 20cm, and the y axis is 25cm as shown in Fig. 8.This dependent on the maximum longest of the two links of robot and corresponded with the circular movement of the DC motors.
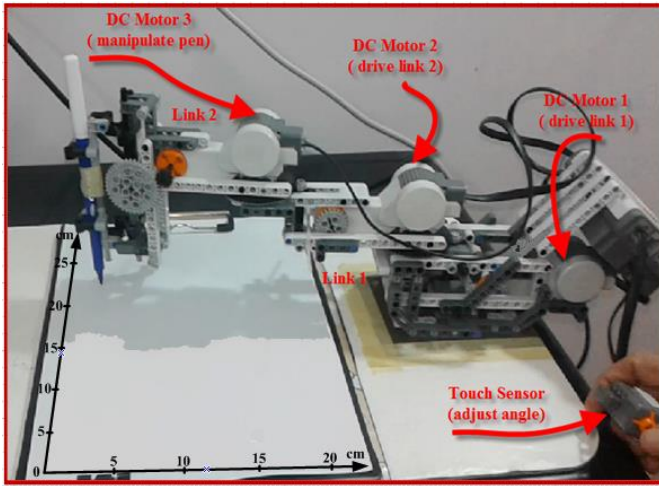
Fig.8. NXT SCARA system with paper sheet

Figure9 represents the block of the inputs and the outputs of SCARA system, where PWM1, PWM2 and PWM3 are the control signal of link1, link2 and pen; respectively. Also, φ_{1m}, φ_{2m} are angle of motor1 and motor2. φ_1 is the angle of link1; φ_2 is the angle of link2.
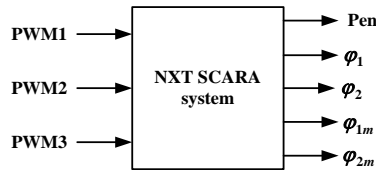

Fig.9. Block diagram of inputs/ outputs system

Where$\varphi = \varphi_m/g$and g is gear reduction ratio.

A. NXT SCARA Modeling

The NXT SCARA robot has an ability to perform high-speed operations and several different tasks [17]. SCARA was designed in Japan; it is used for small components like electronic component. Though the real robot is the final target, it is very useful to perform simulations [18]. NXT SCARA was considered as two-link planar robot arm with a coordinate system shown in Fig. 10. m_1, m_2 are link$_1$ length and link$_2$ length; respectively.
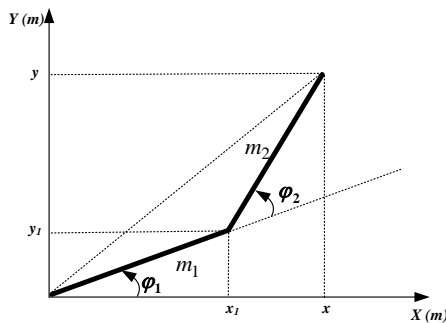

Fig.10. System of two – link planar robot

B. Kinematics and Inverse Kinematics

Kinematic is a component of examining motion. In general, the motion was described via trajectories, velocities as well as accelerations. To measure the trajectories, the robot joint parts can be used by the angle of rotation joint or by the distance of translational joint [19]. Direct kinematics was represented by calculating the position of the end point robot from the joint angles. By joint variables from the position of the robot endpoint, Inverse kinematic is calculated. [20] The inverse kinematic problem is not simple because its equations are nonlinear; their solution is difficult [21] Then, the position (x,y) has a geometricrelation with (φ_1, φ_2) as follows:

$$d_1 = m_1 \begin{bmatrix} \cos\varphi_1 \\ \sin\varphi_1 \end{bmatrix} \qquad (4)$$

$\mathbf{d}_1$ is directing the first segment of the simple mechanism

$$d_2 = m_2 \begin{bmatrix} \cos(\varphi_1 + \varphi_2) \\ \sin(\varphi_1 + \varphi_2) \end{bmatrix} \qquad (5)$$

$\mathbf{d}_2$ the second segment.

$\mathbf{x}$ has described the position of the robot endpoint for solving the inverse kinematics, then calculating the joint angles from the position automatic robot endpoint. The actual cosine rule is employed.

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} m_1\cos\varphi_1 + m_2\cos(\varphi_1 + \varphi_2) \\ m_1\sin\varphi_1 + m_2\sin(\varphi_1 + \varphi_2) \end{bmatrix} \qquad (6)$$

$$x^2 + y^2 = m_1^2 + m_2^2 - 2m_1m_2\cos(180° - \varphi_2) \qquad (7)$$

The actual perspective in the second shared of the two-segment manipulator had been worked out since the inverse trigonometric purpose, the elbow link of the planar manipulator is used to generate a time-varying disturbance to the base link [22].

$$\cos\varphi_2 = \frac{x^2 + y^2 - m_1^2 - m_2^2}{2m_1m_2} = D \quad (8)$$

$$\varphi_2 = \pm\cos^{-1}(D) \quad (9)$$

$$\sin\varphi_2 = \pm\sqrt{1 - D^2} \qquad (10)$$

$$\varphi_2 = \tan^{-1}\frac{\pm\sqrt{1-D^2}}{D} \quad (11)$$

$$\varphi_1 = \tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{m_2\sin\varphi_2}{m_1 + m_2\cos\varphi_2}\right) \qquad (12)$$

IV. TRAJECTORY MAKING

The trajectory is referred to the pervious time of position, velocity and acceleration for every degree of freedom by moving the tool from its initial position to the final position in a certain amount of time [23] for calculating the trajectory, the motion was described by a required duration and a final aim point. The trajectories are designed by m-files Matlab codes in equations or by determined points, to draw the required basic trajectory. The basic trajectory $\eta$is varying with time that wholly dividedη into parts, and then connecting them with each other. This basic trajectory consists of straight line or circular arc.In general, permissive pathways specified of contain intermediate items; the manipulator is going for relaxation by way of position. Be able to pass through a point without stop, and so require generalizing the way in which is suitable for

cubic to the path constraints[24]. All these items become a few sought after sharing perspectives by request of the inverse kinematics. Then, considering the problem of cubic computing that connect the via-point values for each joint together in a smooth way. Using basic trajectory showing in Fig. 11. to make the procedure of a reference trajectory.
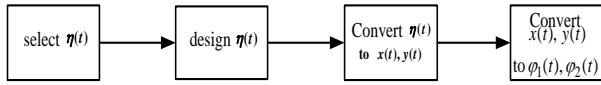

Fig.11.Making treatment of a reference point trajectory with simple trajectory

Generating a trajectory between two configurations, while we wish to define the start and end velocities of the trajectory. The cubic polynomial trajectory is giving a continuity in positions and velocities at the start and finish points, times, and providing discontinuities in the acceleration. The derivative of acceleration is called the jerk that may decrease tracking accuracy. In such case, we now have six demands: one each pertaining to primary and final options, primary and final velocities, and primary and final accelerations. Consequently, we require a fifth order polynomial [24].

$$\eta(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0 \quad (13)$$

$$\dot{\eta}(t) = 5a_5 t^4 + 4a_4 t^3 + 3a_3 t^2 + 2a_2 t + a_1 \quad (14)$$
$$\ddot{\eta}(t) = 20a_5 t^3 + 12a_4 t^2 + 6a_3 t + 2a_2 \quad (15)$$

If t_0 is start time, t_1 is finish time and $\eta_0, \dot{\eta}_0, \ddot{\eta}_0, \eta_1, \dot{\eta}_1, \ddot{\eta}_1$ are position, velocity, acceleration at t_0 and t_1; respectively. Then, calculate the coefficient $a_5, a_4, a_3, a_2, a_1, a_0$ as shown below

$$
\begin{bmatrix} a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} =
\begin{bmatrix}
t_0^5 & t_0^4 & t_0^3 & t_0^2 & t_0 & 1 \\
5t_0^4 & 4t_0^3 & 3t_0^2 & 2t_0 & 1 & 0 \\
20t_0^3 & 12t_0^2 & 6t_0 & 2 & 0 & 0 \\
t_1^5 & t_1^4 & t_1^3 & t_1^2 & t_1 & 1 \\
5t_1^4 & 4t_1^3 & 3t_1^2 & 2t_1 & 1 & 0 \\
20t_1^3 & 12t_1^2 & 6t_1 & 2 & 0 & 0
\end{bmatrix}^{-1}
\begin{bmatrix} \eta_0 \\ \dot{\eta}_0 \\ \ddot{\eta}_0 \\ \eta_1 \\ \dot{\eta}_1 \\ \ddot{\eta}_1 \end{bmatrix} \quad (16)
$$

There are two motions of a robot arm called CP Continuous Path motion and PTP Pose To Pose motion, CP motion is a blind of motion between targets, with the loss of path accuracy at the target transitions, and PTP motion is starting and stopping between configurations. The total trajectory is deriving from concatenating CP trajectory and PTP trajectory [8].

V. NXT SCARA CONTROLLER DESIGN

It is a tedious job to control the robot manipulator for tracking the desired trajectory. By using the adaptive controllers and with indication of the tasks assigned to the robot, this can define the movement of the links and the effectors of the robot [25].The overall structure of a robot can be visible as a combination of sub-devices work together to fit the functionality of the robot; it consists of a controller and a multiple of sensors and actuators. The functional structure of the robot is formed by the sub-devices. Fig. 12 represents the components of a robotic closed loop system, and Fig.13. showed the Simulink module of NXT SCARA system and the adaptive controller.
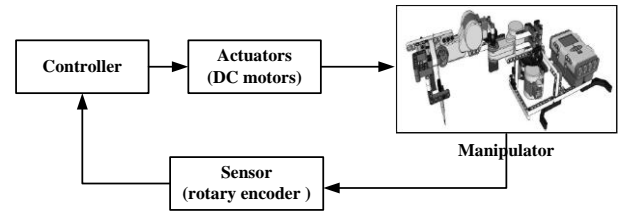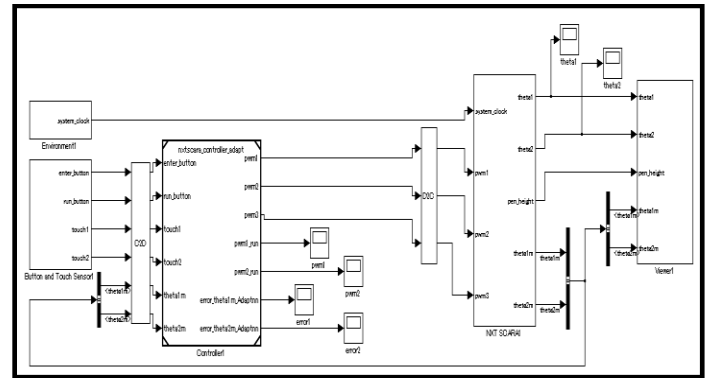

Fig.12.Components of a robotic system

Fig.13.Simulink of unsupervised controller and NXT SCARA system

VI. SIMULATION RESULTS

All simulations are using MATLAB and SIMULINK which are used widely in control applications. By using 'to workspace' blocks, the data is issued to the Matlab. Data are estimated and validated. The comparison between the classic P controller, and the improved unsupervised self- learning Adaline controller. Results show that the controller action by the unsupervised adaptive control provides a faster speed and better robustness of robot manipulators with uncertainties. Fig. 14 and15 show the error between the P controller output, and the unsupervised controller output.
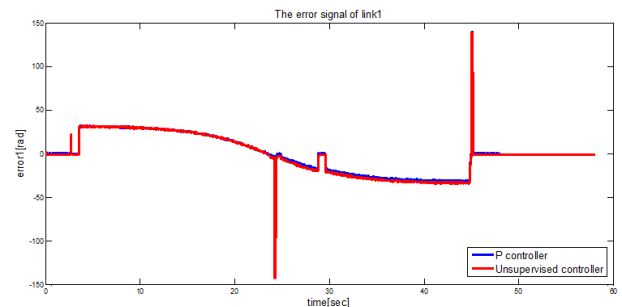

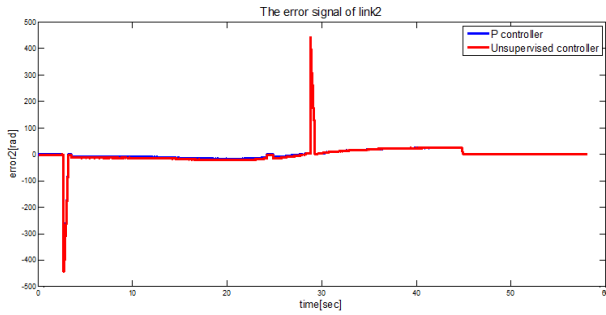Fig.14. Error of motor1 angle of the circle trajectory

Fig.15. Error of motor2 angle of the circle trajectory

From the figures, the error goes to zero, the weights of the network get together with final values, and the error is converged to zero; therefore, the network is trained. Calculate the root mean square error from (17) to get the value of error for the controllers as in table (1). It is clear that the error resulting from the unsupervised adaptive is less than the P controller.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y - y_s)^2} \qquad (17)$$

TABLEI.   THE ERROR VALUESOF CONTROLLERS

The response of an output PWM control signal of controllers is presented in Fig. 16 and 17. The designed controller is overlapping the oscillation from P controller.
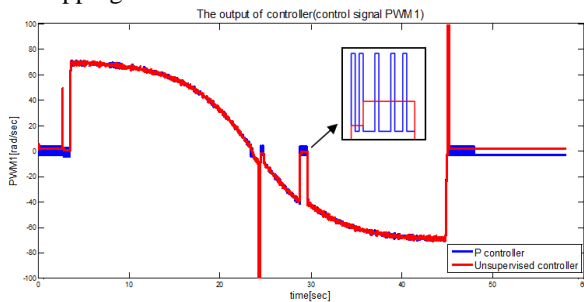

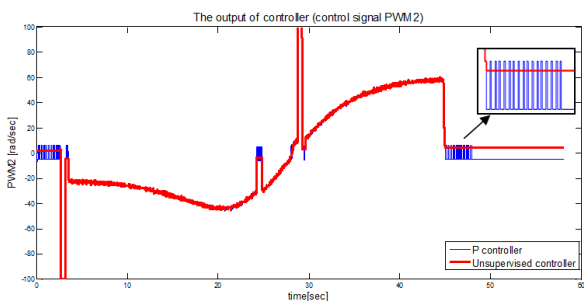Fig.16. Control signal *PWM1* for the circle trajectory

Fig.17. Control signal *PWM2* for the circle trajectory

The output of the NXT SCARA system is φ_{1m} and φ_{2m} for the circle trajectory shown in Fig.18 and 19. The unsupervised controller follows the reference signal in superiority.
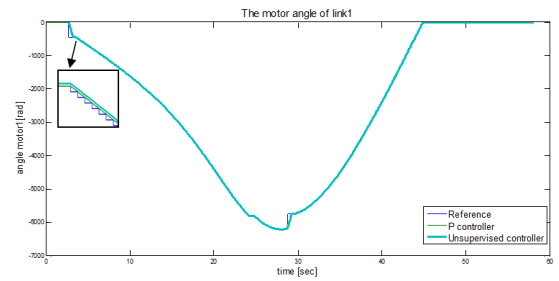

Fig.18. The output of NXT SCARA φ_{1m}
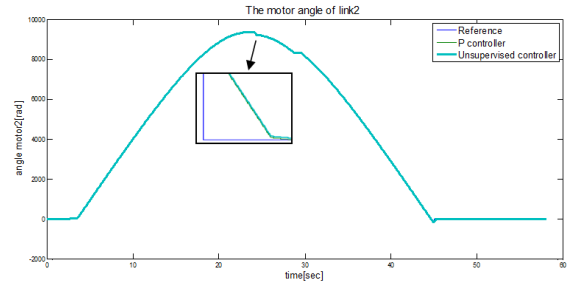

Fig.19. The output of NXT SCARA φ_{2m}

VII. IMPLEMENTATION RESULTS

Utilizing the Lego Mindstorm robot unit is one of the best and simplest methods

Controller	P controller	Unsupervised NN
Error1	22.79	21.49
Error2	38.83	37.55

for building robotics with the latest technologies, servo motors, microcontroller computational force, and remote controllers. Using Lego Mindstorm elements gives an easy way to reconstruct the design in the improvement stage [26].

After the design of the robot, the operating system which is running on the brick of NXT is of high importance. The joints must be stable enough for drawing a straight line, and with enough torque for pushing the pencil to the paper so that the line is well clear. Experiment results give our design results in better performance and simple structure. The simulation results have a good robustness[27]. A real-time experimental result of different controllers' performance compared to NXT SCARA robot is reported. The study focused on tracking trajectories[28]and the result experiment showed that the unsupervised ADALINE ANN controller is adapting the weights to give the desired value. Fig. 20. represents the complete system hardware.

Fig.20. The complete system hardware

In Fig. 21, a and b represent XY of the simulation result and experimental result for spiral trajectory;respectively, and we note that there is no difference between simulations and experimental.
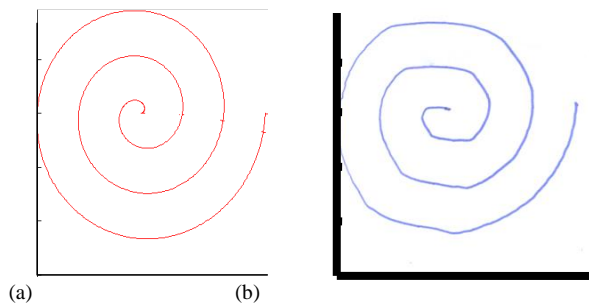

(a)   (b)

Fig.21. Spiral trajectory (a) simulation result, (b) experimental result

Also, Fig. 22, a and b shows the XY simulation result and experimental result for Matlab logo; respectively. The experimental results showed that derived similar results as simulation results.
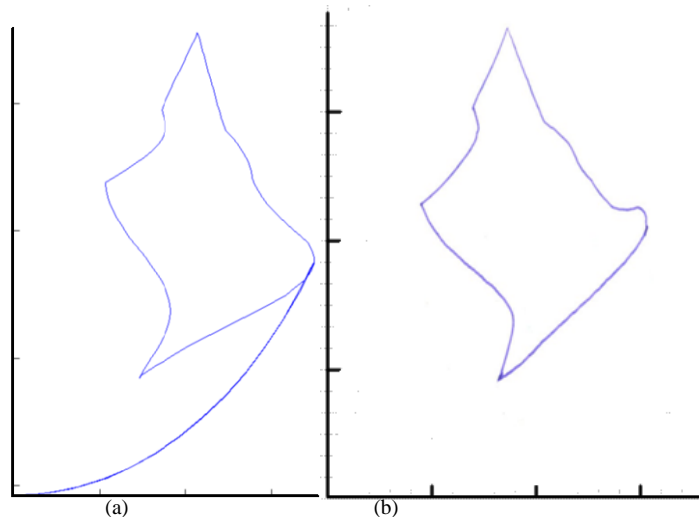

(a)   (b)

Fig.22. Matlab logo trajectory (a) simulation result, (b) experimental result

VIII. CONCLUSION

The performance of adaptive neural network based control strategy was compared with that of the conventional computed torque control method through carrying out several simulations of the robotic arm. The obtained simulation results proved the superiority of the unsupervised adaptive neural network-based controllers over a conventional computed torque rejection controller in compensating for both structured and unstructured uncertainties. The unsupervised learning control is able to be used for controlling the NXT SCARA system in real time. From experimental results, the designed controller was capable of controlling current dynamics of the NXT SCARA effortless and also reduces the tracking error and give less RMSE value. The neural network controller created is based on the unsupervised control which gives the online forecast and control quickly and exactly. The error signal is fed back into the Adaline when disturbance happens during control operation to fit the network weights.

REFERENCES

[1] E. A. H. Sallam. and W. M. F. Abouzaid, "NXT* SCARA Model Based on an Adaptive Neural Network Controller," CiiT International Journal of Artificial Intelligent Systems and Machine Learning, vol. 7, No.02, February 2015.

[2] Mahmoud M. Al Ashi, " Trajectory Tracking Control of A 2-DOF Robot Arm Using Neural Networks," A thesis for the degree of Master of Science in Electrical Engineering, February 2014.

[3] Sane Subhash, " Studies on Trajectory Tracking of Two Link Planar Manipulator," National Institute of Technology, Rourkela, A Thesis for the Degree of Bachelor Of Technology in Mechanical Engineering and Master of Technology in Mechatronics and Automation, May, 2015.

[4] Yousif I. Mohammed, Safwan M. Hussein, "Modeling and Simulation of Industrial SCARA Robot Arm," International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249 – 8958, Vol. 4 Issue 4, April 2015.

[5] R. Jafari, and R. Dhaouadi, "Adaptive PID Control of a Nonlinear Servomechanism Using Recurrent Neural Networks," in Advances in

[6] W. Zuo, and L. Cai, "Adaptive-Fourier-Neural-Network-Based Control for a Class of Uncertain Nonlinear Systems," IEEE Transactions On Neural Networks, vol. 19, 2008.

[7] N. S. Bhattathiri, and P. A. Saraswathi, "ANN Control of Non-Linear and Unstable System and its Implementation on Inverted Pendulum " International Journal of Current Engineering and Technology, vol. 4, pp. 826-831, 2014.

[8] Y. Yamamoto, " NXT SCARA Model-Based Design Control of two-link planar robot arm, built with LEGO Mindstorms", first edition, 2008.

[9] Zhao-Hui Jiang, " Trajectory Control of Robot Manipulators Using a Neural Network Controller," Robot Manipulators Trends and Development, March, 2010.

[10] B.S.K.K. Ibrahim, and Ahmed M.A. Zargoun, "Modelling and Control of SCARA manipulator," ELSEVIER, Science Direct, Procedia Computer Science, vol. 42, pp 106-113, 2014.

[11] S. Sharma, V. Kumar, and R. Kumar, "Supervised Online Adaptive Control of Inverted Pendulum System Using ADALINE Artificial Neural Network with Varying System Parameters and External Disturbance," I.J. Intelligent Systems and Applications, vol. 8, pp. 53-61, 2012.

[12] A.P. Cheema1, Mohd. Sabir2, " Role of Artificial Neural Network for the Stabilization of Non-Linear System ," International Journal of Engineering and Mathematical Sciences, Vol. 1, Issue 1, pp.47-54, June 2012.

[13] Marwan A., Farrukh Nagi., Ksm Sahari, Hanim S., Fadi. I, "On-Line Adaptive Fuzzy Switching Controller for SCARA Robot " Wseas Transactions On Systems And Control, vol. 6, no. 11, pp. 404-416, 2011.

[14] M. Jing, Z. Wenhui, and Z. Haiping, "Adaptive Control for Robotic Manipulators base of RBF Neural Network," TELKOMNIKA, vol. 11, no. 1693-6930, p. 521~528, 2013.

[15] T. Callinan, "Artificial Neural Network identification and control of the inverted pendulum," 2003.

[16] L. Lindqvist, "Unified Infrastructure for Simulation, Communication and Execution of Robotic Systems," Master, Espoo, 2007.

[17] H. Ramli, A. Patar, H. Al-Assadi, M. A. Ayub, "A New Neural - Active Force Control Architecture in High Precision SCARA System," Latest Advances in Systems Science and Computational Intelligence, 2012.

[18] M. S. Alshamasin, F. Ionescu, and R. T. Al-Kasasbeh, "Kinematic Modeling and Simulation of a SCARA Robot by using Solid Dynamics and Verification by MATLAB/Simulink," European Journal Of Scientific Research, vol. 37, pp. 388-405, 2009.

[19] F. Flacco, A. De Luca, I. Sardellitti and N. G. Tsagarakis, "On-line estimation of variable stiffness in flexible robot joints," The International Journal of Robotics Research, 2012.

[20] T. Bajd, M. Mihelj, J. Lenarčič, A. Stanovnik, and M. Munih, "Robotics", vol. 43, Springer, 2010.

[21] B. Krose, and P. van der Smagt, "An Introduction to Neural networks," Eighth edition, 1996.

[22] Y. ZHU, "Adaptive Output Feedback Control Of Nonlinear Systems," Doctor Of Philosophy, Graduate College of the Oklahoma State University, 2004.

[23] J. J. Craig, "Introduction to Robotics Mechanics and Control," third edition, 2005.

[24] M. W. Spong, S. Hutchinson, and M. Vidyasagar, "Robot Modeling and Control," first edition, 2006.

[25] Prabu D, S. Kumar and R. Prasad, "Advanced Dynamic Path Control of the Three Links SCARA using Adaptive Neuro Fuzzy Inference System," in Robot Manipulators Trends and Development, 2010.

[26] Akos Hamori, Janos Lengyel, Barna Resko, " 3DOF drawing robot using LEGO-NXT," 15th IEEE International Conference on Intelligent Engineering Systems, pp 293 – 295, 2011.

[27] M. Jing, Z. Wenhui and Z. Haiping, "Adaptive Control for Robotic Manipulators base on RBF Neural Network," Telkomnika, vol. 11, no. 1693-6930, p. 521~528, 2013.

[28] A. Marwan, F.Nagi, K. Sahari, and S. Hanim, "Real-Time Tracking Controllers For SCARA Robot," in Proceedings of International Conference on Engineering and Information Technology "ICEIT2012", Toronto, Canada 2012.