



EXPERIMENTAL INVESTIGATION OF AN ADAPTIVE NEURO-FUZZY CONTROL SCHEME FOR INDUSTRIAL ROBOTS

Abdallah Farrage*, **Abdel Badie Sharkawy,**
Ahmed S. Ali, M-Emad S. Soliman, and Hany A. Mohamed

Staff in Mechanical Engineering Department, Assiut University, Assiut, Egypt

Received 22 March 2014; revised 14 April 2014; accepted 5 May 2014

ABSTRACT

This paper presents the application of an adaptive fuzzy logic controller with feed-forward component (AFLCF) to the Selective Compliance Assembly Robot Arm (SCARA Robot). The feed forward torque component is computed on-line using an artificial neural network (ANN) which has been trained off-line. This feed-forward component is designed to deliver the ideal torque component to the robot drivers. The feedback fuzzy logic control (FLC) component is made to keep the stability of the closed loop system. As the FLC is dependent in its rule base, here, a compact rule base is used. It consists of only four rules per each degree of freedom (DOF). The FLC ensures closed loop stability in the sense of Lyapunov and is valid for second order nonlinear systems. Furthermore, adaptability of the FLC has been achieved to enhance the tracking performance. The theoretical background of this control algorithm has been published in[1].

Using SCARA robot as the testing platform, here, experimental results are presented for the following five controllers: the conventional PD controller, PD controller tuned by fuzzy system (PDT), the FLC, Adaptive FLC (AFLC), and finally the AFLCF. The controllers are tested experimentally at the same initial conditions to make fair comparison between their performances. Results show that the investigated AFLCF has outperformed the other controllers.

Keywords: PD Control, PD controller tuned by fuzzy system (PDT), Fuzzy Logic Controller (FLC), adaptive fuzzy logic controller (AFLC), adaptive fuzzy controller with feed forward component (AFLCF), Artificial Neural Network (ANN).

1. Introduction

Robots are very important tools for industrial automation but their control is a quite challenging problem. The difficulties of control problem in the field of industrial robots are caused by the dynamics governing robot motion. This can be attributed to nonlinearities, uncertainties and strong coupling of the robot dynamics. For this reason, feedback controllers should be applied to accommodate system uncertainties, coupling, and parameters variations.

* Corresponding author.

E-mail address: abdallahfraag@yahoo.com

There are many types of controllers which have been applied in the field of robotics. The classical controllers (PD, PID) are widely used in practice [2][3], however their tracking performance are in question, since they are dependent on fixed gains, which cannot tackle the changing environment.

FLC has demonstrated excellent robustness in both simulations and experimental applications[4][5]. A customary for this phenomenon is that fuzzy sets, with their membership property are less sensitive to error than crisp sets. Plain FLC however, requires large number of parameters which are needed to be adjusted. Simplicity of the rule base and adaptability of control gain is thus needed to make FLC applicable.

Adaptive control has been studied for many decades to deal with changed unknown parameters. An adaptive controller usually supposes modifications of the plant model, and learning control assumes the modifications of the control strategy based on past experience[6][7]. Adaptive control system can adjust itself to accommodate new situations, such as change in the observed dynamic behavior of a plant. Tuning of the controlled system can be made by different methods such as gradient steepest descent [8], ANN[9][10][11], genetic algorithm[12], particle swarm optimization (PSO)[13]. In essence, adaptive control monitors the input-output behavior of the system to identify it. In addition, the perfect knowledge of the inertia parameters can be relaxed via adaptive controller.

In this paper, an AFLCF is experimentally investigated. The theoretical basis of this controller can be found in [1]. The controller consists of adaptive fuzzy feedback control component which achieve closed loop stability and ANN feed forward control component which identifies the robot dynamics during motion and delivers the ideal torque needed to robot motion. Experimental comparison is made with conventional PD controller, PDT, FLC, and AFLC.

In addition to this section, this paper is organized as follows. Section 2 presents the system description and defines parameters of the SCARA Robot. Section 3 contains short introduction about ANN and how it is used to identify the ideal torque required to derive the SCARA Robot. The control algorithm is presented in section 4. Section 5 shows the software architecture. Results and discussion are established in section 6. Finally, section 7 contains our concluding remarks.

2. System description

The system which has been used to test the proposed algorithm is the Selective Compliance Assembly Robot Arm (SCARA Robot). There are efforts which have been made to achieve the precise control of SCARA robot[14][15][16][17]. Figure 1 shows a photo of the SCARA robot. The robot exists in the Mechatronics Lab, Faculty of Engineering, Assiut University, Egypt. This robot is utilized in processes such as pick and place, painting, brushing, and pegging in hole, etc. It has four degrees of freedom (DOF), i.e. waist swivel, elbow swivel, roll and vertical arm. Axes are driven by dc electric servo motors. The dc motors are controlled by the robot computer through the servo power motor drive. The robot is also equipped with feedback position sensor for each axis. The feedback position sensor is optical encoder. The encoder sends signals back to a high speed computer. The computer can compare position with the programmed command signals to direct the movement of the robot. The encoder fixed on each joint is an incremental one and has three tracks, two of

them have one thousand holes for each direction, and the third track has one hole for the home position. Each encoder is supplied by 5 volts. The robot is also equipped with velocity sensors (tachometer). All axes of the robot have two limit switches. The robot has also a home limit sensor on each axis which is used for the starting reference point. Table 1 contains working range for the four driven axes. Experiments have been performed on only two links, the waist and elbow, which are nominated as axis1 θ_1 and axis2 θ_2 . In all experiments, the servo motors have been converted to direct drive dc motors. The feedback signals from the incremental optical encoder and control signals to the motor drive are sent to/from the computer via Programming Control Interface card (PCI). Resolution of the card is 16 bits or 0.000305V per LSB when it works at ± 10 volts. The motor driver is MID-7654/7652 Servo Power Motor Drive. It is manufactured by the National Instrument Company. It is a complete power amplifier and system interface for use with four axes of simultaneous motion control. The card is ideal for interfacing industrial applications. It can be used for interfacing hardware such as encoders, limit switches, inputs/outputs, and other motion devices. The control program is written in Lab View language. Section 5 introduces the flowchart of the control program.



Fig.1. A photo of the SCARA robot.

Table1.

Working Range for the four driven axes.

Axis	Symbols	Angle	Movement	Function
1	Θ_1	270	Waist Swivel	Arm Rotate
2	Θ_2	270	Elbow Swivel	Articulate
3	Θ_3	360	Roll	Wrist Twist
4	Z	20 cm	Vertical	Up/Down

3. Off-line identification using artificial neural networks (ANNs)

System identification is an important way of understanding and investigating the world around. Identification is a process of deriving a mathematical model of a predefined part of the world, using observations. There are several different approaches of system identification, and these approaches utilize different forms of knowledge about the system. When only input-output observations are used, behavioral or black box model can be constructed. In black box modeling, neural networks play an important role. In recent years

artificial neural networks (ANNs) have gained a wide attention in control applications and system identification. The ability of the artificial neural networks to control nonlinear systems can be the most readily exploited in the synthesis of nonlinear controllers. A survey of using ANNs in control systems can be found in [18]. ANNs have layers of nodes represented as circles, joined by connections through which signals may be transmitted. In Fig. 2, the left layer contains the input nodes and the right layer contains the output nodes. The input and output layers are linked by one hidden layer of nodes. Every node in a layer is connected with every node in adjacent layers [19],[20],[21].

Consider multi-layer neural networks for approximating general relationships. So we need learning algorithms for these more complicated neural networks. The following popular learning algorithm, referred to as the back propagation algorithm, is a generalization of the delta rule. The supervised learning method (learning with a teacher) is used because the correct action for every input in training sets (input /output pairs) is known.

Suppose we have a training set T where

$$\begin{aligned} T &= \{(x^q, y^q), q = 1, 2, \dots, N\} \\ x^q &= (x_1^q, x_2^q, \dots, x_n^q) \in R^n \\ y^q &= (y_1^q, y_2^q, \dots, y_m^q) \in R^m \\ O^q &= (O_1^q, O_2^q, \dots, O_m^q) \in R^m \end{aligned}$$

x and y are the pairs set and O is the output of the neural network. It is require using this data to adjust the weights and the biases of a neural network with n input nodes, and m output neurons with one hidden layer.

This paper proposes an artificial neural network (ANN) model for SCARA robot to accomplish path-planning of trajectory tracking. Its aim is to let the SCARA navigates through and follows its intended goal location. The movement of the robot is fully autonomous without any human assistance except when training the network in off-line mode. The proposed ANN is a three layers “4-10-2” network composed of an input, a hidden, and an output layer. The input layer is made out of 4 input source nodes $\theta_1, \dot{\theta}_1, \theta_2$ and $\dot{\theta}_2$. The hidden layer is made out of 10 neurons which receive input data from the input layer and multiply them by the values of the synaptic weights denoted by w_{ij} and then forward the resulted values to the output layer. The output layer is made out of 2 neurons that are directly linked to the motor driver of the robot which controls its movement and its mechanical operation. The employed activation function is hyperbolic tangent for the hidden neurons, whereas it is linear for the output neurons. The synaptic weights range from w_{11} to $w_{10,2}$ and they represent the interconnection between the different neurons of the network. Formally, the proposed neural network can be defined as follows:

$$ANN = \{I, T, w, v, A\}$$

where I denotes the set of input nodes, T denotes the topology of the network including the number of layers and the number of their neurons, w and denote the set of synaptic weights values, and A denotes the activation function.

$$\begin{aligned} I &= \{\theta, \dot{\theta}\} \\ T &= \{L_{in-2}, L_{h0-10}, L_{out-1}\} \\ W_{L\ in} &= \begin{Bmatrix} w_{11} & w_{21} & w_{31} & w_{41} & w_{51} & w_{61} & w_{71} & w_{81} & w_{91} & w_{101} \\ w_{12} & w_{22} & w_{32} & w_{42} & w_{52} & w_{62} & w_{72} & w_{82} & w_{92} & w_{102} \end{Bmatrix} \end{aligned}$$

$$v = \{v_{11} \ v_{12} \ v_{13} \ v_{14} \ v_{15} \ v_{16} \ v_{17} \ v_{18} \ v_{19} \ v_{110}\}$$

$$A = \{\exp(x) - \exp(-x)/\exp(x) + \exp(-x), \ 1\}$$

The identification using neural network is essentially a one step ahead prediction structure in which past inputs and outputs are used to predict the current output. In this paper, ANNs are applied to identify two degrees of freedom. In Fig.2, the input layer consists of four nodes: $\theta_1, \dot{\theta}_1, \theta_2$ and $\dot{\theta}_2$ which are the position and velocity of the waist and elbow link respectively. The hidden layer consists of 10 nodes. The input weights to each neuron in the hidden layer are w_{i1}, w_{i2}, w_{i3} , and w_{i4} , where ($i = 1, 2, \dots, 10$) is number of neuron of the hidden layer. The weights of the output layer are v_{1i}, v_{2i} . τ_{FF1}, τ_{FF2} are the feed forward term for joint1 and joint2, respectively. The network has been trained off-line using experimental data which was previously obtained from open loop control tests for the two links under investigation. The following trajectories have been examined:

$$\theta_{d1} = \theta_{d2} = 50 \sin(0.056 * 2\pi * t) \quad \text{for } 0 < t \leq 30 \quad (1)$$

The identification error of the system is shown in Fig.3. In this Figure, the identification error of both joint 1 and joint 2 are shown in (a) and(b), respectively. The aim of the feedforward component is to deliver on-line the apocoptate torque component to the controller which is suitable to carry the desired trajectory equation (1). A generalized model for the robot which can be implemented for any trajectory is out of scope of this paper.

The best training performance is shown in Fig.4.

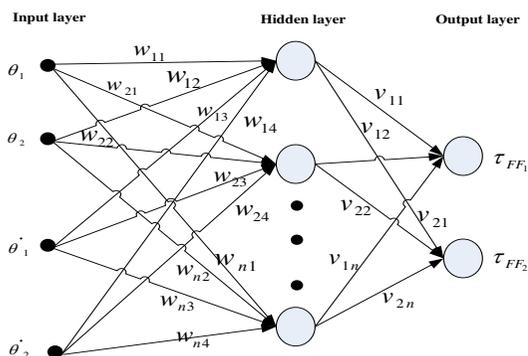


Fig. 2. Feed forward Neural Network of the system.

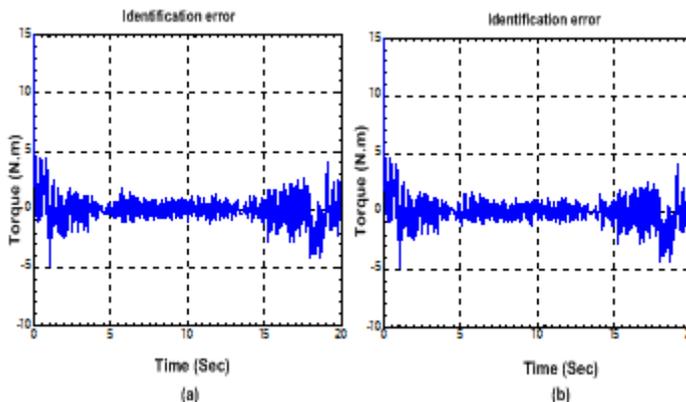


Fig. 3. Identification error: (a) identification error of joint 1. (b) error of joint 2.

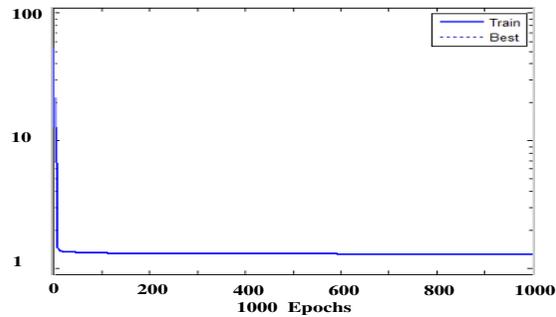


Fig. 4. The best training performance

4. The control algorithms

It's known that, the trajectory tracking of the robotic systems is the best method for testing any control algorithm. The nonlinearities, changed friction coefficients, and strong coupling of the robot dynamics present a challenging control problem.

In this paper, five of model free control algorithms had been applied. These algorithms are PD, PDT, FLC, AFLC, and AFLCF. The parameters of joints are considered $\theta = [\theta_1 \ \theta_2]$ where θ_1 is the angle of joint1, and θ_2 stands for joint2. We consider the state variables as $\theta(t)$ and $\dot{\theta}(t)$ which are usually available as feedback signals.

Define the tracking error vector $e(t)$ and $\dot{e}(t)$ as:

$$e(t) = \theta(t) - \theta_d(t) \quad (2)$$

$$\dot{e}(t) = \dot{\theta}(t) - \dot{\theta}_d(t) \quad (3)$$

where θ_d and $\dot{\theta}_d$ are vectors of the desired joint position and velocity, respectively. The desired trajectory for each joint is mentioned in (1).

4.1. PD control

The control torque for the proportional-plus-derivative (PD) controller is defined by:

$$U(t) = K_p e(t) + K_d \dot{e}(t) \quad (4)$$

where, K_p and K_d are 2×2 positive definite diagonal matrices called the proportional and the derivative gain matrices of the controller, respectively. Despite its simplicity, a traditional problem associated with PD control is that the constant control gains could not be increased infinitely, to improve the controller's performance. When the values of the gains exceed their critical values, the system becomes unstable. Thus the performance of the PD controller is restricted with these values of gains which are usually experimentally determined. The block diagram of the PD controller is shown in Fig.5.

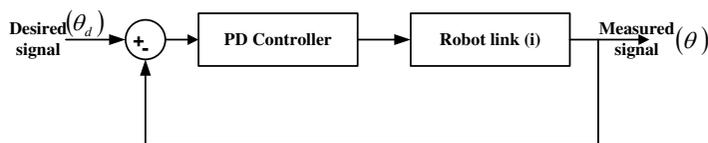


Fig. 5. Block diagram of PD Controller.

4.2. PD controller tuning by fuzzy system (PDT)

The performance of classical PD controllers depends on values of proportional and derivative. Thus the performance of the closed-loop can be improved by tuning these gains. In [22] the author used PID control and examined on-line tuning for gain by using fuzzy system. In this paper, the tuning was made only on PD control. Table 2 shows the

rules of used fuzzy system (FS) for joint one and joint two. The block diagram of the PDT controller is shown in Fig.6.

Table 2.
The rules of fuzzy system.

\dot{e}	N	Z	P
e			
N	N	N	Z
Z	N	Z	P
P	Z	P	P

The control law of the PDT is defined by

$$U(t) = K_p[b(t)\theta_d - \theta] + K_d\dot{e}(t) \tag{5}$$

$$b(t) = \gamma + f(t) \tag{6}$$

where, K_p and K_d are the proportional and derivative gains, θ_d and θ are the desired and measured angle respectively, γ is a positive constant less than or equal 1, and $f(t)$ is the output of the fuzzy inference system [22]. When $b(t)$ is equal to one, the controller behaves as conventional PD controller. The block diagram of the PDT is shown in Fig.6. The fuzzy system consists of three Gaussian membership functions for each of the two inputs $e(t)$, $\dot{e}(t)$ and the output. The linguistic variables are negative N, zero Z, and positive P. The used fuzzy system has membership functions which are shown in Fig.7. The similar membership has been used for joint 2 except the range of output signal.

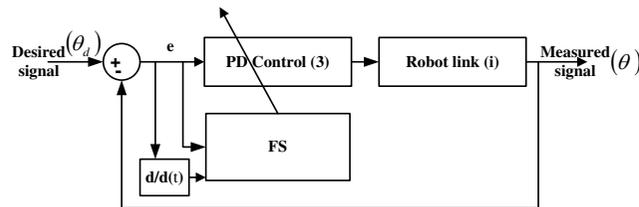


Fig. 6. The closed-loop diagram of the PDT Controller.

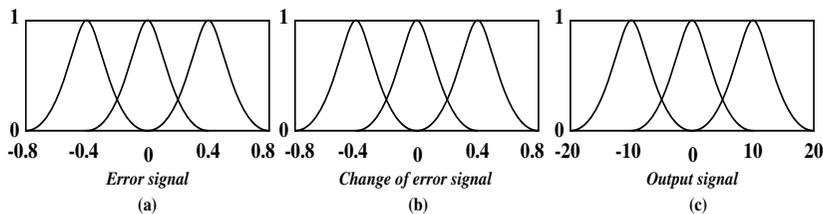


Fig.7. membership function of the fuzzy system

4.3. Fuzzy logic control (FLC)

The performance of any fuzzy logic control is dependent on its inference rules. In this paper number of rules which are used is four rules only. They can be derived as follows [1]:

Consider e_i and \dot{e}_i are the error and the change in error of the system, and i is the DOF. The Lyapunov direct method can be applied to achieve stability of the system. To this end, consider the following Lyapunov candidate function:

$$V = \frac{1}{2}(e^T e + \dot{e}^T \dot{e}) \tag{7}$$

By differentiating with respect to time gives

$$\dot{V}_i = e_i \dot{e}_i + \dot{e}_i \ddot{e}_i \tag{8}$$

To enforce asymptotic stability, it is required to find u_{FB_i} so that

$$\dot{V}_i = e_i \dot{e}_i + \dot{e}_i \ddot{e}_i \leq 0 \tag{9}$$

In some neighborhood of the equilibrium of (6). Taking the control signal u_{FB} to be proportional to \ddot{e} so the previous equation can be rewritten as:

$$\dot{V}_i = e_i \dot{e}_i + \alpha_i \dot{e}_i u_{FB_i} \tag{10}$$

where, α_i is positive constant and u is the control signal.

Sufficient conditions for the previous equation can be stated as follows.

- If for $i \in [1, \dots, n]$ e_i and \dot{e}_i have opposite signs then $\alpha_i u_{FB_i}$ must to be zero;
- If for $i \in [1, \dots, n]$ e_i and \dot{e}_i are both positive then $\alpha_i u_{FB_i}$ must to be negative;
- If for $i \in [1, \dots, n]$ e_i and \dot{e}_i are both negative then $\alpha_i u_{FB_i}$ must to be positive;

One can easily obtain the four rules listed below in Table 3. In this table P, N, denote positive and negative error, respectively. $u_P, u_N,$ and u_Z are positive, negative, and zero control signal respectively. These rules are simple and satisfy the conditions of stability.

Table 3.
Fuzzy rules for the fuzzy feedback controller

$\begin{matrix} \dot{e}_i \\ e_i \end{matrix}$	P	N
P	u_N	u_Z
N	u_Z	u_P

To complete the design, we must specify the fuzzy system with which the fuzzy feedback computes the control signal. The Gaussian membership defining the linguistic terms in the rule base is chosen as follows:

$$\begin{aligned} \mu_{positive}(x) &= G(x, a) = e^{-(x-a)^2} \\ \mu_{negative}(x) &= G(x, -a) \\ \mu_{zero}(x) &= G(x, 0) \end{aligned}$$

where, $a > 0$.

The above four rules can be represented by the following mathematical expression

$$u_{FB_i} = \frac{G(e_i, a_{1i})(-k_i) + G(e_i, -a_{1i})(k_i)}{G(e_i, a_{1i}) + G(e_i, -a_{1i})} + \frac{G(\dot{e}_i, a_{2i})(-k_i) + G(\dot{e}_i, -a_{2i})(k_i)}{G(\dot{e}_i, a_{2i}) + G(\dot{e}_i, -a_{2i})} \tag{11}$$

In more details

$$u_{FB_i} = -k_i \left[\frac{\exp(-(e_i - a_{1i})^2) - \exp(-(e_i + a_{1i})^2)}{\exp(-(e_i - a_{1i})^2) + \exp(-(e_i + a_{1i})^2)} + \frac{\exp(-(\dot{e}_i - a_{2i})^2) - \exp(-(\dot{e}_i + a_{2i})^2)}{\exp(-(\dot{e}_i - a_{2i})^2) + \exp(-(\dot{e}_i + a_{2i})^2)} \right]$$

From which

$$u_{FB_i} = -k_i \left[\frac{\exp(2a_{1i}e_i) - \exp(-2a_{1i}e_i)}{\exp(2a_{1i}e_i) + \exp(-2a_{1i}e_i)} + \frac{\exp(2a_{2i}\dot{e}_i) - \exp(-2a_{2i}\dot{e}_i)}{\exp(2a_{2i}\dot{e}_i) + \exp(-2a_{2i}\dot{e}_i)} \right] \tag{12}$$

where a_{1i} and a_{2i} are positive constants. The control law becomes

$$u_{FB_i} = -k_i [\tanh(2a_{1i}e_i) + \tanh(2a_{2i}\dot{e}_i)], \quad i = 1, \dots, n \tag{13}$$

This equation gives the feedback control signal needed to stabilize the system. The inputs of this equation are only e_i and \dot{e}_i , and u_{FB_i} is the control input for each joint i . The block diagram of the FLC is shown in Fig.8. The benefits of this control law are

- This control law is a special case of the fuzzy systems.
- Only three parameters per each DOF need to be tuned, namely, they are k_i , a_{1i} , and a_{2i}
- This controller is inherently bounded since $|\tanh(x)| \leq 1$.
- Each joint has independent control input u_{FB_i} , $i = 1, 2, \dots, n$.

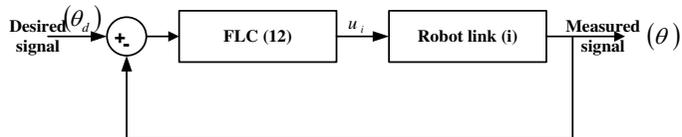


Fig. 8. The block diagram of the FLC Controller.

4.4. Adaptive fuzzy logic control (AFLC)

The problem in the previous FLC is that, there are three parameters needed to be empirically determined, so that a solution of this problem is to achieve some degree of adaptability. This provides attractive features such as fast response and improves the control system. [In this research paper, AFLC which was used here can be produced by addition of adaptive term to the previous FLC. The PD fuzzy gain K_i is chosen so as to optimize the following quadratic performance index:

$$J_i = \frac{1}{2} \{ r_i (u_{FB_i})^2 \} \quad (14)$$

where, input r_i is a constant. According to the gradient method, the learning algorithm of the parameter k_i in the fuzzy feedback controller can be derived as follow

$$\Delta k_i = -\frac{\partial J_i}{\partial k_i} = -\frac{\partial J_i \partial u_{FB_i}}{\partial u_{FB_i} \partial k_i} \quad (15)$$

$$\Delta k_i = -r_i u_{FB_i} [\tanh(2c_{1i} e_i) + \tanh(2c_{1i} \dot{e}_i)]. \quad (16)$$

The block diagram of the AFLC is shown in Fig.9.

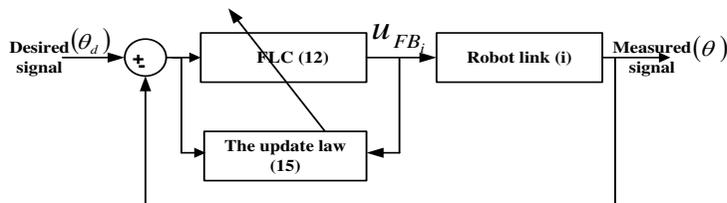


Fig. 9. Block diagram of the AFLC Control.

4.5. The adaptive fuzzy logic controller with feed forward component (AFLCF)

The proposed control consists of feedback and feed forward torque components. The feedback torque component is produced from AFLC which keeps the system stable. The feed forward torque component is computed on-line using an artificial neural network (ANN) which has been trained off-line as mentioned in Section 3. This feedforward component is designed to deliver the ideal torque component to the robot drivers. The addition of this

feedforward component is further improves the performance[23]. The inputs of feedforward component are θ_i , and $\dot{\theta}_i$ for each joint and the output is the torque τ_{FF_i} , $i = 1,2$. The overall closed-loop control system is shown in Fig.10. In this Figure, u_{FF_i} is the value of the feed forward component torque, u_{FB_i} is the value of torque which is produced by the AFLC, and u_i is the value of the total input torque to the actuator.

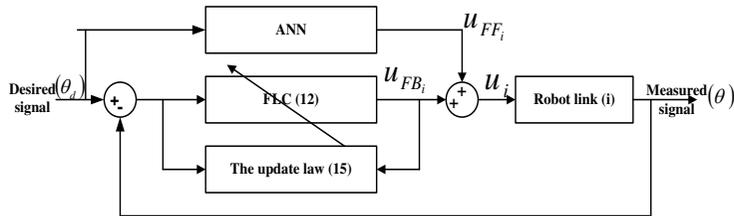


Fig.10. The overall close-loop of the system.

5. Software architecture

This section presents the software architecture which is used to implement the AFLCF. It consists of three steps: FLC, AFLC, and the feedforward component. The adaptation term is made by the performance index to minimize the quadratic performance index (13). The three steps are joined to form the AFLCF algorithm. The program is written in Lab View language (National Instrument). This language depends on engineering graphics and connections. The program consists of two windows: front panel, and the block diagrams. The front panel has the same function of the graphical user interface (GUI).The objective of this window is to permit the change in the design and specification of the program. The block diagram is the place which the program is written. The flow chart of the program is shown in Fig.11. It shows the basic stages of the control algorithm.

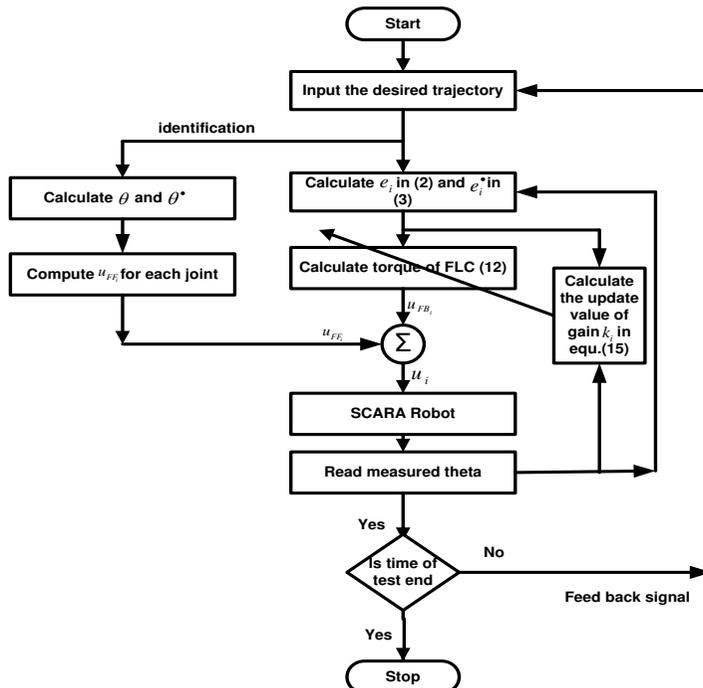


Fig. 11. Flow chart of the software architecture for the proposed algorithm.

6. The Experimental results

In this section, we show the experimental results of the AFLCF algorithm and compare it with other four controllers: PD conventional controller, PDT, FLC, and AFLC. The whole experimental system contains the host computer, data acquisition, and the SCARA robot. In all the experiments, the five controllers are initiated with initial position error equal 10° , i.e. $q(0) = [10^\circ - 10^\circ]^T$ to analyze the strength and weakness of each one. This condition yields an initial position error; $e = [0.175 - 0.175]^T$ radians. As mentioned earlier, experiments have been performed on only two links, the waist and elbow, which are nominated as axis1 θ_1 , and axis2 θ_2 . The sinusoidal trajectory with amplitude 50° or 0.87 rad. and frequency of 0.056 cycle/sec was applied for all control algorithms. The desired trajectory becomes $[\theta_d = 50 \sin(0.056 * 2\pi t)]$ for each joint.

The trajectory tracking of PD conventional is shown in Fig.12. In this Figure, the desired and actual trajectories of both joint 1 and joint 2 are shown in (a) and (b), respectively. In both Figures, the solid line describes the desired trajectory and the dash line describes the actual trajectory. The proportional and derivative gains used in the experiments for joint 1 and joint 2 are $K_p = 600$, $K_d = 8$ and $K_p = 700$, $K_d = 8$ respectively. It is shown that the system is stable and tracking has been achieved with reasonable error. The torque signal of PD controller is shown in Fig.13.

Figure.14 shows the trajectory of PDT algorithm. In this Figure, The desired and actual trajectories of both joint 1 and joint 2 are shown in (a) and (b), respectively. In PDT algorithm, the fuzzy system was used to tune the proportional gain. The output of fuzzy system is shown in Fig.15. Fig.16 shows the value of updated gain k_i of the PDT controller.

The error signals of PD and PDT are shown in Fig.17. In This Figure, the error signal of PD is shown in (a) and (b) shows the error of PDT.

The trajectory tracking of FLC is shown in Fig.18. In this Figure, the desired and actual trajectories of both joint 1 and joint 2 are shown in (a) and (b), respectively. In both Figures, the solid line describes the desired trajectory and the dash line describes the actual trajectory.

Figure 19 shows the trajectory tracking of AFLC algorithm. This Figure shows the desired and actual trajectories of both joint 1 and joint 2 in (a) and (b), respectively. In both Figures, the solid line describes the desired trajectory and the dash line describes the actual trajectory. The value of updated gain of AFLC algorithm which is mentioned in (15) is shown in Fig.20.

The error signals of FLC and AFLC are shown in Fig.21. In This Figure, the error signal of FLC is shown in (a) and (b) shows the error of AFLC.

The trajectory tracking of AFLCF is shown in Fig.22. In this Figure, the desired and actual trajectories of both joint 1 and joint 2 are shown in (a) and (b), respectively. In both Figures, the solid line describes the desired trajectory and the dash line describes the actual trajectory. The torque signal of AFLCF algorithm is shown in Fig.23. This torque consists of fuzzy torque component which shown in Fig.24 and feed forward torque component. The last second component is resulted from robot identification. Figure 25 shows value of this component. The value of updated gain of AFLC algorithm which is mentioned in (15) is shown in Fig.26. The error signal of AFLCF is shown in Fig.27.

For each algorithm, the error performance of joint 1 is shown in Fig.28, and joint 2 is shown in Fig.29.

In this research, two meaningful methods of the tracking error are applied to compare the performance results of the five algorithms. The first method is the scalar value Root Mean Square error (RMS) which is defined as

$$RMS = \sqrt{\frac{1}{T_f} \sum_{j=1}^N [\theta_d(j) - \theta(j)]^2 \Delta T} \tag{17}$$

Where, θ_d and θ are the desired and measured trajectory, respectively. T_f is the final time which is equal $N * \Delta T$. The performance of these criteria for five controllers is shown in Fig.30. The second method is the maximum absolute value of the tracking error after two second from the starting time. It is defined as

$$e_{max} = \max_{0 \leq j \leq N} |\theta_d(j) - \theta_m(j)|. \tag{18}$$

The performance of these criteria is shown in Fig.31.

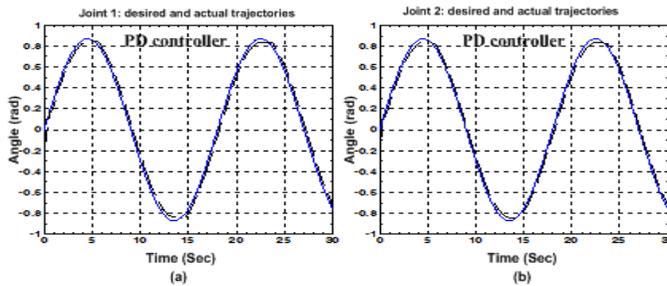


Fig. 12. The performance of the conventional PD Controller.

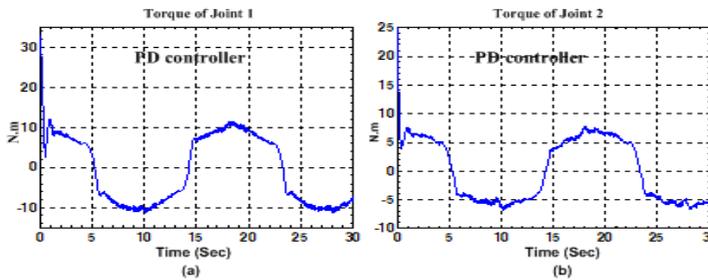


Fig. 13. the torque signal of the conventional PD Controller.

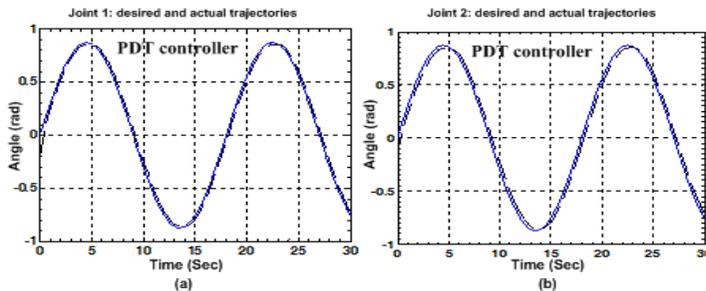


Fig. 14. The performance of the PDT Controller.

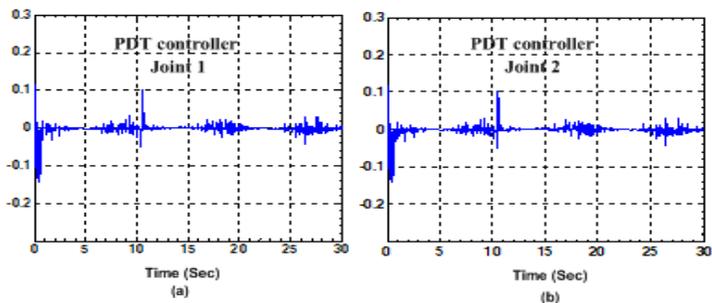


Fig. 15. The output of the fuzzy system in (5).

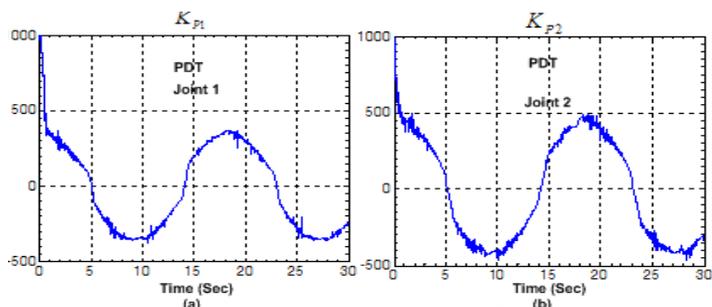


Fig. 16. The value of updated gain k_i of the adaptive PD controller.

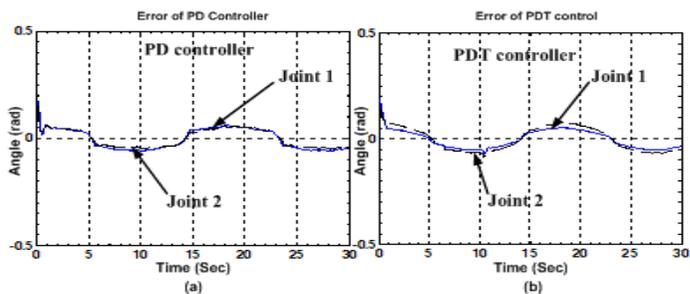


Fig. 17. The error performance of the PD, and PDT controller.

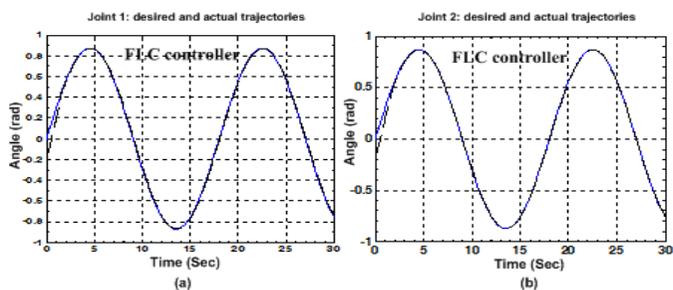


Fig. 18. The performance of the FLC Controller.

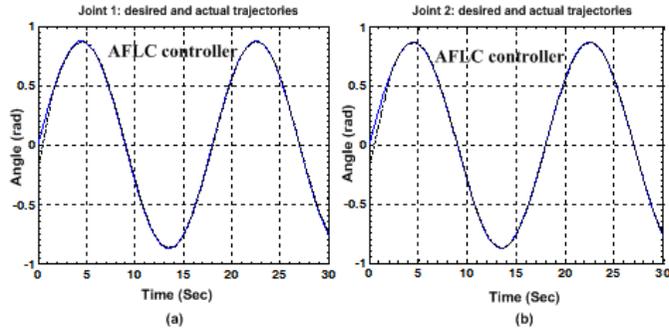


Fig.19. The performance of the AFLC Controller.

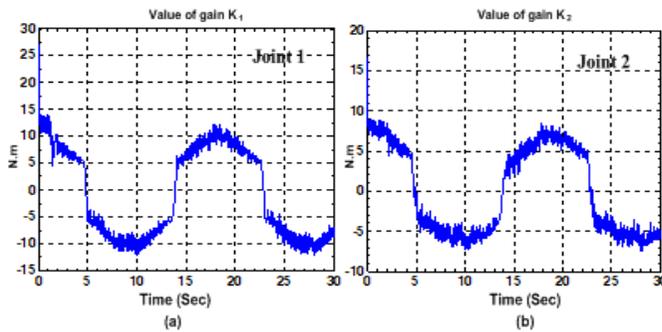


Fig. 20. The value of updated gain k_i of the adaptive controller.

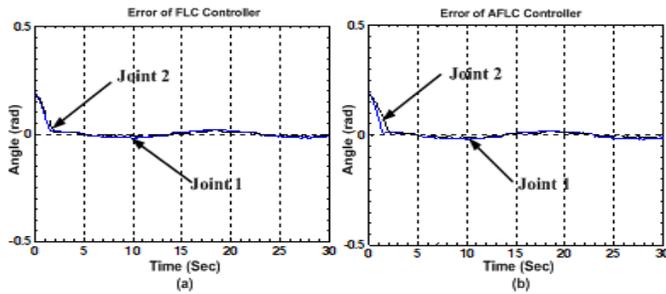


Fig. 21. The error performance of the FLC, and AFLC controller.

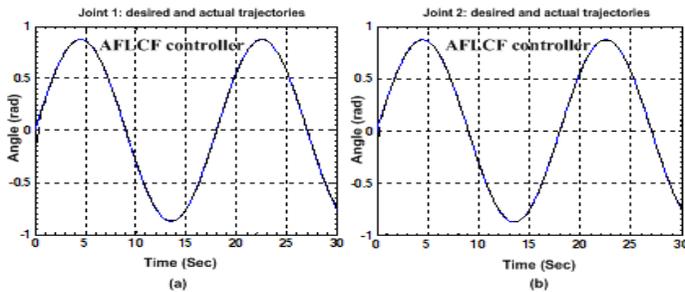


Fig. 22. The performance of the proposed algorithm.

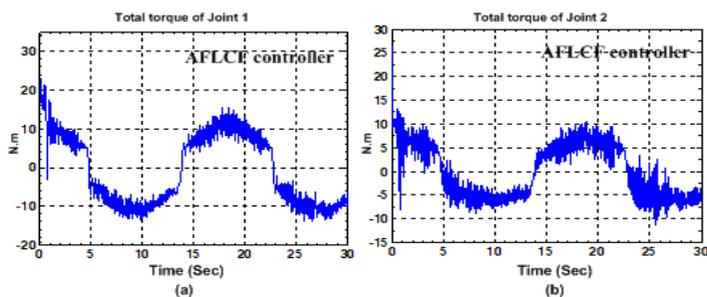


Fig. 23. The total torque signal of the proposed algorithm.

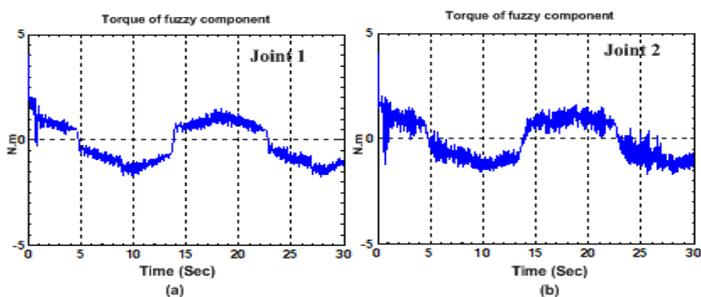


Fig. 24. The value of fuzzy component of the system.

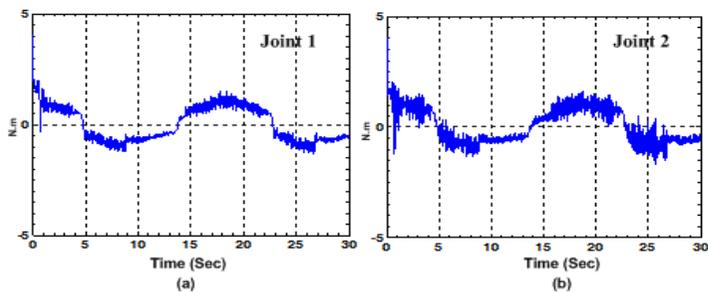


Fig. 25. The value of feed forward component of the system.

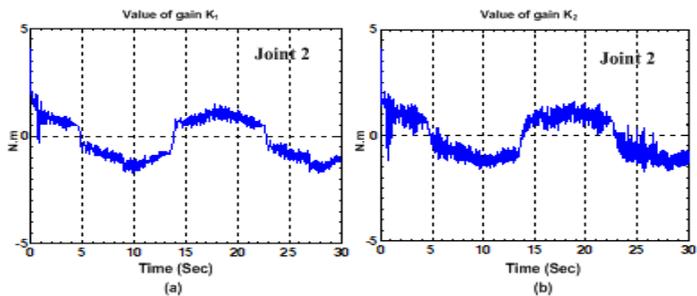


Fig. 26. The value of update gain k_i of the adaptive controller.

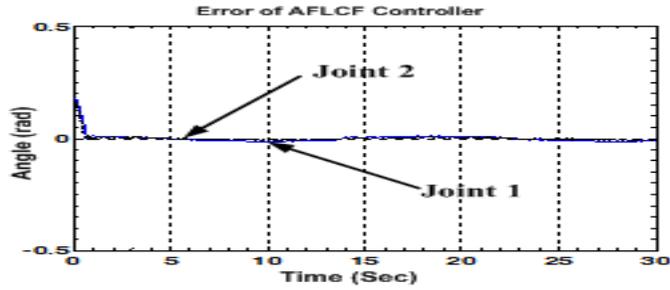


Fig. 27 .The error performance of proposed controller.

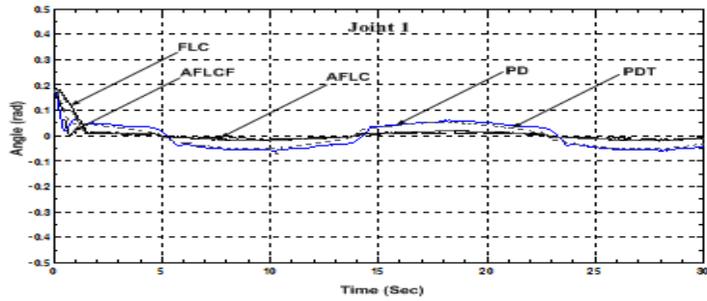


Fig. 28. The error performance of joint 1.

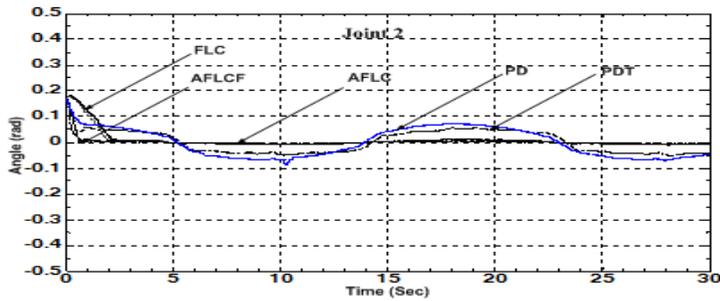


Fig. 29. The error performance of joint 2.

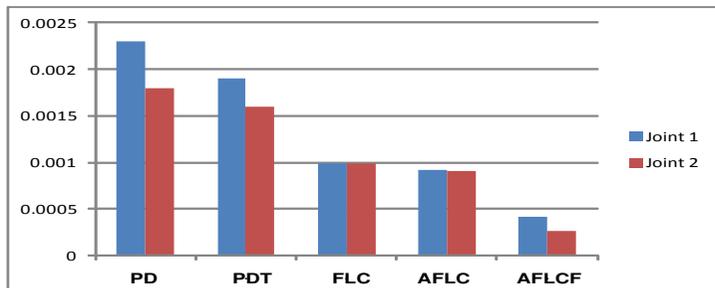


Fig. 30. The RMS error for the five controllers in radian.

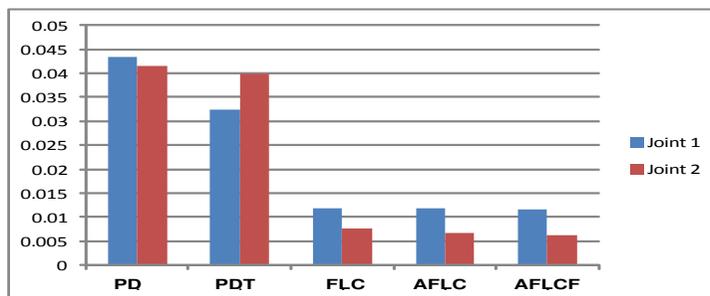


Fig.31. The maximum error for the five controllers in radian.

7. Conclusion

An adaptive neuro-fuzzy control scheme which is suitable for industrial applications has been successfully implemented in this paper. In addition, ANNs were used to identify the SCARA robot which has been used to deliver an ideal torque component in the feedforward path. The experiments are implemented on only two joints, but the procedure followed in this paper can be extended to number of link robots. The initial error is used to test the robustness of the system. The performance of the robot has been quantified using two performance measures, the root mean square of error and the maximum error. Relative to conventional PD controller, PDT, FLC, and AFLC, the adaptive neuro-fuzzy controller has achieved the best performance.

REFERENCES

- [1] A. B. Sharkawy, "A computationally efficient fuzzy control scheme for a class of MIMO systems," *Alexandria Eng. J.*, vol. 52, no. 4, pp. 583–594, Dec. 2013.
- [2] J. Alvarez-Ramirez, I. Cervantes, and R. Kelly, "PID regulation of robot manipulators: stability and performance," *Syst. Control Lett.*, vol. 41, no. 2, pp. 73–83, Oct. 2000.
- [3] I. Cervantes and J. Alvarez-Ramirez, "On the PID tracking control of robot manipulators," *Syst. Control Lett.*, vol. 42, no. 1, pp. 37–46, Jan. 2001.
- [4] C. W. De Silva, "application of fuzzy logic in the control of robotic manipulators," *Fuzzy Sets Syst.*, vol. 70, pp. 223–234, 1995.
- [5] Y. Cai, Q. Zhan, and X. Xi, "Path tracking control of a spherical mobile robot," *Mech. Mach. Theory*, vol. 51, pp. 58–73, May 2012.
- [6] S. Jagannathan, M. W. Vandegrift, and F. L. Lewis, "Adaptive fuzzy logic control of discrete-time dynamical systems," *Automatica*, vol. 36, pp. 229–241, 2000.
- [7] K.-C. Chiou and S.-J. Huang, "An adaptive fuzzy controller for robot manipulators," *Mechatronics*, vol. 15, no. 2, pp. 151–177, Mar. 2005.
- [8] A. Marwan, F. Nagi, K. S. M. Sahari, S. Hanim, and I. Fadi, "On-Line Adaptive Fuzzy Switching Controller for SCARA Robot," *WSEAS Trans. Syst. Control*, vol. 6, no. 11, pp. 404–416, 2011.
- [9] J. Lin and R.-J. Lian, "Hybrid fuzzy-logic and neural-network controller for MIMO systems," *Mechatronics*, vol. 19, no. 6, pp. 972–986, Sep. 2009.
- [10] J. P. P., J. P. Perez, R. Soto, A. Flores, F. Rodriguez, and J. L. Meza, "Trajectory tracking error using PID control law for two-link robot manipulator via adaptive Neural Networks," *Procedia Technol.*, vol. 3, no. 81, pp. 139–146, Jan. 2012.
- [11] C. Cheng and W. Baoqiang, "A Trajectory Tracking Method for Welding Manipulator Based on Fuzzy Gauss Function Neural Network," *Procedia Eng.*, vol. 29, pp. 189–193, Jan. 2012.

- [12] M. S. Alam and M. O. Tokhi, "Hybrid fuzzy logic control with genetic optimisation for a single-link flexible manipulator," *Eng. Appl. Artif. Intell.*, vol. 21, no. 6, pp. 858–873, Sep. 2008.
- [13] Z. Bingül and O. Karahan, "A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 1017–1031, Jan. 2011.
- [14] M. J. Er, M. T. Lim, and H. S. Lim, "Real-time hybrid adaptive fuzzy control of a SCARA robot," *Microprocess. Microsyst.*, vol. 25, no. 8, pp. 369–378, Nov. 2001.
- [15] M. Popescu, I. B. Ş. I, O. Olaru, L. Ț. A. Popescu, and F. Grofu, "The simulation hybrid fuzzy control of scara Robot," *WSEAS Trans. Syst. Control*, vol. 3, no. 2, pp. 105–114, 2008.
- [16] S. Akamatsu, "Position Control of 2-Link SCARA Robot by using Internal Model Control," *Mem. Fac. Eng. Okayama Univ.*, vol. 43, no. January, pp. 49–54, 2009.
- [17] M. Isaksson, "Improving the Kinematic Performance of the SCARA-Tau PKM," *IEEE*, pp. 4683–4690, 2010.
- [18] S. Paper, "Neural Networks for Control Systems A Survey *," *automatia*, vol. 28, no. 6, pp. 1083–1112, 1992.
- [19] H. T. Nguyen, N. R. Prasad, C. L. Walker, and E. A. Walker, *A First Course in fuzzy and Neural control*. 2003, pp. 1–305.
- [20] F. Cus, U. Zuperl, and J. Balic, "Combined feedforward and feedback control of end milling system," *Achiev. Mater. Manuf. Eng.*, vol. 45, no. 1, pp. 79–88, 2011.
- [21] N. A. Shiltagh and D. A. A. Kadeer, "Modified training method for feedforward neural networks and its application in 4-link SCARA Robot identification," *Engineering*, vol. 17, no. 5, pp. 1335–1344, 2011.
- [22] A. B. Sharkawy, "Engineering Applications of Artificial Intelligence Genetic fuzzy self-tuning PID controllers for antilock braking systems," *Eng. Appl. Artif. Intell.*, vol. 23, no. 7, pp. 1041–1052, 2010.
- [23] S. Kolyubin, D. Efimov, V. Nikiforov, H. B. A, and P. Plaza, "Control of Nonlinear Systems Using Multiple Model Black-Box Identification," *IFAC Ncolcos*, vol. 61, no. 1, pp. 1–6, 2013.

اختبارات عملية لمحكم من نوع عصبي هلامي مهائى للمناولات الالية الصناعية

الملخص العربى

يهدف هذا البحث الى تطبيق التحكم الهلامي المهائى بالاضافه الى مركبة للتغذية الامامية (AFLCF) على المناول الالى المستخدم فى الصناعة (SCARA Robot). حيث يتم حساب مركبة التغذية الامامية اثناء تشغيل المناول الالى (on-line) باستخدام الشبكات العصبية الاصطناعية التى تم تمرينها مستقلة (off-line) باستخدام معلومات مسبقة تم الحصول عليها من خلال اختبارات تم اجرائها على المناول الالى بنظام الدائرة المفتوحة (open loop). اما مركبة المحكم الهلامي المستخدم فى التغذية الخلفية هو من نوع المهائى (adaptive) والهدف الاساسى منه هو الحفاظ على اتزان المناول الالى اثناء تشغيله. كما يعتمد التحكم الهلامي على قاعدة بيانات تم استنتاجها باستخدام نظرية ليابانوف (Lyapunov). وهذا المحكم صالح لتطبيقه على مجموعة كبيرة من الانظمة الميكانيكية من الدرجة الثانية.

يهدف هذا البحث الى الاختبار الاداء العملى للمحكم السابق ذكرة ومقارنة ادائه بمجموعة اخرى من المحكمات التى لا تعتمد على النموذج الرياضى للمناول الالى وهذه المجموعة بيانها كالاتى

- 1- المحكم التقليدى من نوع التناسبى التفاضلى (PD)
- 2- المحكم التفاضلى التناسبى ذاتى الضبط باستخدام نظام هلامي (PDT)
- 3- المحكم الهلامي (FLC)
- 4- المحكم الهلامي المهائى (AFLC)

ولضمان مقارنة عادلة بين هذه المحكمات الخمس تم اختيار الثوابت الخاصة بكل محكم عمليا بحيث يحقق افضل اداء ممكن. تم اختبارهم جميعا بنفس الظروف الابتدائية للحركة فى وجود خطأ ابتدائى للتأكد من مدى قدرتها على توجيه حركة المناول الالى (Robustness). وتم تقييم الاداء باستخدام معيار الجذر التربيعى للخطأ (RMS) وكذلك معيار اقصى خطأ (Max. error).

وقد اظهرت النتائج تفوق ملحوظ للمحكم الهلامي المهائى باستخدام مركبة للتغذية الامامية (AFLCF) على المحكمات الاربعة الاخرى.