

FUZZY TRACKING CONTROL OF TWO DEGREES OF FREEDOM ROBOTIC ARM

Mahmoud M. Othman, Abdel Badie Sharkawy, and Abouel Makarem A. Khalil

Mechanical Engineering Department, Faculty of Engineering Assiut University, Assiut, Egypt – Postal Code: 71516

E-mail: mmo_83@aun.edu.eg

(Received November 5, 2009 Accepted December 22, 2009.)

In this paper, we derive a fuzzy logic controller (FLC) for robotic systems. The approach implements fuzzy partition to the state variables based on Lyapunov synthesis. The resulting control law is stable and able to exploit the dynamic variables of the system in a linguistic manner. The presented methodology enables the designer to systematically derive the rule base of the control. We further simplify the procedure leading to a computationally efficient FLC. The methodology is model free approach and does not require any information about the system nonlinearities, uncertainties, time varying parameters, etc. Here, we present experimental results of the following controllers: the conventional PD controller and the proposed FLC. The two controllers are tested and compared with respect to ease of design, implementation, and performance of the closed loop system.

KEYWORDS: *Trajectory Tracking, Fuzzy Logic Controller (FLC), PD Control, Two-Link Robot*

1. INTRODUCTION

Robots are familiar examples of trajectory-following mechanical systems. Their nonlinearities and strong coupling of the robot dynamics present a challenging control problem, [1]. Conventional methods of controlling a nonlinear system are based on models, especially in the field of robot control. Many robotic control schemes can be considered as special cases of model-based control called computed torque, [2]. The basic concept of computed torque is to linearize a nonlinear system, and then to apply linear control theory. Practical implementation of the computed torque and other model based approaches can be found in [3] where the experimental results revealed that the simple PD controller has outperformed the other model based controllers. This is mainly due to the fact that in many dynamic systems the parameters may slowly change or cannot be exactly predicted in advance due to different operating conditions. This is particularly true if a manipulator changes its payload mass.

Adaptive control has been studied for many decades to deal with constant or slowly changing unknown parameters. Applications include manipulators, ship steering, aircraft control and process control, [4]. Although the perfect knowledge of the inertia parameters can be relaxed via adaptive technique, its real practical usefulness is not really clear and the obtained controllers may be too complicated to be easily implemented, [5]. Nevertheless, some experiments have been presented in [6,7].

Control engineers are now facing more and more complex systems, and mathematical models of these systems are difficult to obtain. Thus, model free approaches become important in control engineering. Conventional control has some model-free approaches, such as PD and PID control. Alternatively, fuzzy control is also a model free approach; that is, it does not require a mathematical model of the system under control. When compared with other nonlinear modeling and control techniques such as artificial neural networks [8-9], fuzzy systems have the important advantage of providing an insight on the linguistic relationship between the variables of the system, [10].

Fuzzy controllers have demonstrated excellent robustness in both simulations and real-life applications, [11-13]. They are able to function well even when the controlled system differs from the system model used by the designer. A customary for this phenomenon is that fuzzy sets, with their gradual membership property, are less sensitive to errors than crisp sets. Another explanation is that a design based on the "computing with words" paradigm is inherently robust; the designer forsakes some mathematical rigor but gains a very general model which remains valid even when the system's parameters and structure vary [10]. This is well-demonstrated by the method of fuzzy Lyapunov synthesis [11], which is followed in this paper.

Otherwise, FLCs consist of a number of parameters that are needed to be selected and configured in prior, i.e. input membership functions, fuzzification method, output membership functions, rule base, premises connective, inference method and defuzzification. Optimal tuning of FLCs using genetic algorithms has attracted many authors, [14-17]. In these papers, however, there are too many parameters involved in the development of FLCs. Furthermore, genetic algorithms cannot be used in real time control applications. In another study similar to the present work, i.e. real-time trajectory tracking control of two link robot using fuzzy systems [18], the controller needs 26 parameters to be experimentally selected. Also, the FLC in [14] needs 45 parameters to be tuned. This beside the huge number of calculations involved in the computation of the control signals.

In this research paper, a simple and computationally efficient FLC is introduced. The algorithm has been presented in [19] by the second author. Here we represent it along with experimental verification. The controller is stable in the sense of Lyapunov theory of stability and few parameters are needed to be tuned. The approach can be implemented to both tracking and stabilizing control problems. However, in this paper the emphasis is on the tracking control problem of robotic systems. The performance of the proposed controller is experimentally verified and compared with the conventional PD controller. As will be seen, the two controllers are model free approaches, so that a fair comparison is made. Robustness is examined in the presence of payload mass.

The rest of this paper is organized as follows. Section 2 presents the dynamic model, the parameters of the experimental manipulator utilized in this work and the control statement. The proposed control scheme is introduced in Section 3 and in Section 4 we derive the fuzzy controller for the tracking control problem. Section 5 describes the experimental setup, the examined trajectories and the criteria used in the control performance evaluation. The experimental results are demonstrated and discussed in Section 6. Section 7 offers our concluding remarks.

2. ROBOT DYNAMICS AND THE CONTROL PROBLEM FORMULATION

Consider the two-link planner robotic shown in Fig. 1. The vector of generalized coordinates is denoted by $\theta = [\theta_1 \ \theta_2]^T$ where θ_1 and θ_2 are the joint angles, and the corresponding generalized forces (torques) vector is given by $\tau = [\tau_1 \ \tau_2]^T$. The standard Euler-Lagrange dynamic equations for an n-link rigid robot in the absence of gravity may be written as [20]:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} = \tau \quad \text{with} \quad \theta \in R^n, \quad \tau \in R^n \quad (1)$$

where $M(\theta) \in R^{n \times n}$ is the inertia matrix, and θ , $\dot{\theta}$, and $\ddot{\theta}$ are the joint angles, velocities, and accelerations, respectively. The vector $C(\theta, \dot{\theta})\dot{\theta}$ represents centrifugal and Coriolis terms, and τ is the vector of applied torques.

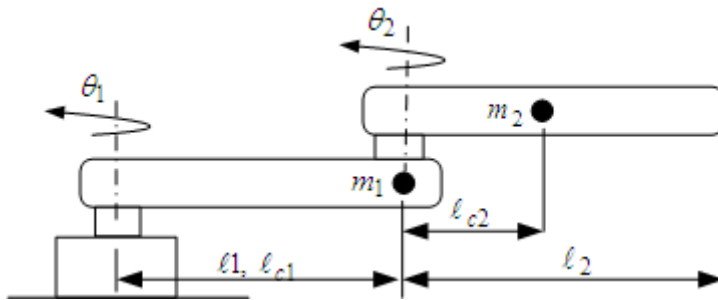


Figure 1 Schematic diagram of a two-link manipulator.

The equations of motion of the arm are given by

$$\tau_1 = [m_1 l_1^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} c_2) + (I_1 + I_2)] \ddot{\theta}_1 + [m_2 (l_{c2}^2 + l_1 l_{c2} c_2) + I_2] \ddot{\theta}_2 - m_2 l_1 l_{c2} s_2 (\dot{\theta}_2^2 + 2\dot{\theta}_1 \dot{\theta}_2) \quad (2a)$$

$$\tau_2 = [m_2 (l_{c2}^2 + l_1 l_{c2} c_2) + I_2] \ddot{\theta}_1 + (m_2 l_{c2}^2 + I_2) \ddot{\theta}_2 - m_2 l_1 l_{c2} s_2 \dot{\theta}_1 \dot{\theta}_2 \quad (2b)$$

where l_1 and l_2 are the links lengths; m_1 and m_2 are the masses of the links; l_{c1} and l_{c2} are the location of the center of masses; I_1 and I_2 are the moment of inertia about the center of masses of the two links. The short hand notation $s_2 = \sin(\theta_2)$ and $c_2 = \cos(\theta_2)$ has been used.

The physical properties of the links used in this study are given in Table 1. These inertial parameters have been calculated by simply measuring and weighting the mechanical elements of the arms.

Table 1 Parameters of the robot arm.

Parameter	Link 1	Link 2
m mass (kg)	0.471	0.096
ℓ length (m)	0.154	0.205
ℓ_c position of c.g. (m)	0.154	0.1025
$I = \frac{1}{12} m\ell^2$ inertia around c.g. (kg.m ²)	0.00093	0.00033

The motion control problem of manipulators in joint space can be stated in the following terms. Assume that the joint position θ and the joint velocity $\dot{\theta}$ are available for measurement. Let the desired joint position θ_d be a differential vector function. We define a motion controller as a controller which determines the actuator torques u in such a way that the following control aim be achieved:

$$\lim_{t \rightarrow \infty} \theta(t) = \theta_d(t)$$

In this paper, we say that the control system is asymptotically stable if the control aim is guaranteed irrespective of the robot initial configuration, i.e. $\theta(0)$ and $\dot{\theta}(0)$. Throughout this work, we shall make the mild assumption that the desired trajectory $\theta_d(t)$ and its derivative (i.e. $\dot{\theta}_d(t)$) are continuous and bounded. We also assume that $\theta_d(t)$ and its derivative are available for online control computation. In robot tracking tasks, the desired position history is generally planned ahead of time and its derivative can be easily obtained.

3. THE PROPOSED FUZZY CONTROL SCHEME

In this section we describe an approach for the fuzzy control design. We apply the fuzzy synthesis to the design of stable controllers. To this end, consider a class of nonlinear systems whose dynamic equations can be expressed as:

$$\dot{x}(t) = f(x, u), \quad y = h(x) \quad (3)$$

where $f(x, u)$ is an unknown continuous function, u is the control input, y is the output, and $x(t) = [x_1, x_2, \dots, x_n]^T$ is the state vector, where $x_{i+1} = \frac{dx_i}{dt} = \dot{x}_i$, $i = 1, 2, \dots, n-1$. We now seek a smooth Lyapunov function $V : R^n \rightarrow R^n$ for the continuous feedback model (3) that is positive definite, i.e. $V(x) > 0$ when $x \neq 0$ and $V(x) = 0$ when $x = 0$, and grows to infinity: $V(x) \rightarrow \infty$ as $x^T x \rightarrow \infty$. Obviously, this holds for the following quadratic form:

$$V(x, t) = \frac{1}{2} x_1^2 + \frac{1}{2} x_2^2 \cdots \cdots + \frac{1}{2} x_n^2 \quad (4)$$

Differentiating (4) with respect to time gives

$$\dot{V}(x, t) = x_1 x_2 + x_2 x_3 \cdots \cdots + x_n \dot{x}_n \quad (5)$$

The standard results in Lyapunov stability theory imply that the dynamic system (3) has a stable equilibrium $x = x_e$ if $\dot{V} < 0$ along the system trajectories. To achieve this, we have chosen the control $u(x)$ to be proportional to \dot{x}_n .

Next, our controller design is achieved if we determine a fuzzy control $u(x)$ so that:

$$\dot{V}(x, t) = x_1 x_2 + x_2 x_3 \cdots \cdots + \alpha x_n u(x) \leq 0 \quad (6)$$

where α is a positive constant. The results of Wang [21-23] states that a fuzzy system that would approximate (6) exists. To this end, one would consider the state vector $x(t)$ to be an input to the fuzzy system. The output of the fuzzy system is the control u . A possible form of the control rules is:

IF x_1 is ($l v$) and/or x_2 is ($l v$) ... and/or x_n is ($l v$) THEN u is ($l v$)

where ($l v$) are linguistic values (e.g. *positive, negative*). These rules constitute the rule base for a Mamdani-type FLC.

In the above formulation, two basic assumptions have been made. They are:

- The knowledge of the state vector. It is assumed to be available from measurements.
- The control input u is proportional to \dot{x}_n . This assumption can be justified for a large class of second order nonlinear mechanical systems, (Margaloit *et al* [11] and Wang [23]). For instance, here in robotics, it means that the acceleration of links is proportional to the input torque.

These two assumptions represent the basic knowledge about the system which is needed to derive the FLC rules. Of course, the exact mathematical model is not needed. In the next section we derive a fuzzy logic controller for the tracking control problem of robotic systems.

4. ROBOTIC FUZZY TRACKING CONTROL

A robot manipulator is, in general, a highly nonlinear coupled dynamic system and, therefore, achieving high performance in trajectory tracking control is a very challenging task. In practice, the load may vary while performing different tasks, the friction coefficients may change in different configurations and some neglected nonlinearities as backlash may appear. Therefore, the control objective is to design a stable fuzzy controller so that the link movement follows the desired trajectory in spite of such effects.

Consider a class of robots whose vector of generalized coordinates is denoted by $\theta = [\theta_1 \theta_2 \cdots \theta_n]^T$ where θ_i , $i = 1, \cdots, n$ are the joint parameters. We consider the state variables of the robot as $\theta(t)$ and $\dot{\theta}(t)$, which are usually available as feedback signals. Define the tracking error vectors $e_p(t)$ and $e_v(t)$ as:

$$e_p(t) = \theta_d(t) - \theta(t), \text{ and } e_v(t) = \dot{\theta}_d(t) - \dot{\theta}(t) \quad (7)$$

where $\theta_d(t)$ and $\dot{\theta}_d(t)$ are vectors of the desired joint position and velocity respectively.

We now apply the approach presented in the previous section in order to find a fuzzy controller that would achieve tracking to the robotic system under consideration. To this end, let us choose the following Lyapunov function candidate

$$V = \frac{1}{2}(e_p^2 + e_v^2) \quad (8)$$

Differentiating with respect to time gives

$$\dot{V}_i(t) = e_{pi}e_{vi} + e_{vi}\dot{e}_{vi}$$

where $i \in [1, \dots, n]$ denotes the joint number. To enforce asymptotic stability, it is required to find u so that

$$\dot{V}_i(t) = e_{pi}e_{vi} + e_{vi}\dot{e}_{vi} < 0 \quad (9)$$

in some neighborhood of equilibrium of eqn (8). Taking the control u to be proportional to \dot{e}_v , eqn (9) can be rewritten as:

$$\dot{V}_i(t) = e_{pi}e_{vi} + \alpha e_{vi}u_i < 0 \quad (10)$$

where α is a positive constant, $i = 1, 2, \dots, n$. Sufficient conditions for (10) to hold can be stated as follows:

- (a) if, for each $i \in [1, \dots, n]$ e_{pi} and e_{vi} have opposite signs and u_i is zero, inequality (10) holds;
- (b) if e_{pi} and e_{vi} are both positive, then (10) will hold if u_i is negative; and
- (c) if e_{pi} and e_{vi} are both negative, then (10) will hold if u_i is positive.

Using these observations, one can easily obtain the rules listed below in Table 2.

Table 2 Fuzzy rules for the tracking controller.

		e_{vi}	
		P	N
e_{pi}	P	u_N	u_Z
	N	u_Z	u_P

In this table, P, N, denote respectively positive, negative errors; u_P , u_N , and u_Z are respectively positive, negative and zero control inputs. These rules are simply the fuzzy partitions of e_p , e_v and u which follow directly from the stabilizing conditions of Lyapunov function (8).

In concluding words, the presented approach transforms classical Lyapunov synthesis from the world of exact mathematical quantities to the world of words [10].

This combination provides us with a solid analytical basis from which the rules are obtained and justified.

To complete the design, we must specify the membership functions defining the linguistic terms in the rule base. Here, we use the Gaussian membership functions

$$\mu_{positive}(x) = G(x; a_z) = e^{-(x-a_z)^2}$$

$$\mu_{negative}(x) = G(x; -a_z)$$

$$\mu_{zero}(x) = G(x; 0)$$

where $a_z > 0$ and z stands for control variable, the product for "and" and center of gravity inferencing [21]. The above four rules can be represented by the following equation:

$$u_i = \frac{G(e_{pi}, a_{epi})(-a_{ui}) + G(e_{pi}, -a_{epi})(a_{ui})}{G(e_{pi}, a_{epi}) + G(e_{pi}, -a_{epi})} + \frac{G(e_{vi}, a_{evi})(-a_{ui}) + G(e_{vi}, -a_{evi})(a_{ui})}{G(e_{vi}, a_{evi}) + G(e_{vi}, -a_{evi})}$$

This yields the FLC controller

$$u_i = -\frac{a_{ui}}{2} [\tanh(2a_{epi} e_{pi}) + \tanh(2a_{evi} e_{vi})], \quad i = 1, \dots, n \quad (11)$$

In (11), the inputs are the error in position e_{pi} and the error in velocity e_{vi} and the output is the control input. So that it may be called fuzzy PD controller. Simplicity is evident, since equation (11) contains the Gaussian membership functions, the fuzzy rules, fuzzification and defuzzification which implies easy implementation, i.e. the FLC is computationally effective. The following remarks are in order:

- Only three parameters per each degree of freedom (DOF) need to be tuned, namely, they are a_{ui} , a_{epi} and a_{evi} . This greatly simplifies the implementation.
- This controller is inherently bounded since $|\tanh(x)| \leq 1$.
- Despite the nonlinearity and coupling effects, only four rules constitute the rule base for each joint.
- It should be noted that this controller is decoupled, i.e. each joint has independent control input.
- Finally, the FLC (11) does not depend on the equations of motion.

A schematic diagram for the closed-loop control is shown in Fig. 2.

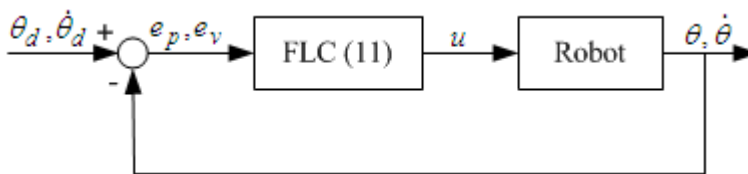


Figure 2 Configuration of the robotic fuzzy control structure.

5. THE EXPERIMENTAL SETUP

5.1 System Description

The experimental set-up consists of a geared-drive horizontal robot arm with 2 DOF whose rigid links are joined with revolute joints; Fig. 3. The robot has been built at the Mechatronics lab, Faculty of Engineering, Assiut University. It is equipped with joint position sensors, motor drivers, ADDA interface card, and a host computer. Both links, made of aluminum, are actuated by brushed dc motors with gear reduction controlled via simple H-bridge drive circuit. The motors operate at rated 24 volt, 2rpm, and 1.5 Nm. Position information is obtained from analogue angular potentiometers for both angles. The potentiometers are one turn (300 degrees) and 1 k Ω . Each potentiometer is coupled to the joint motor. Both potentiometers are supplied by ± 5 V, so that each one has a resolution of 0.033 volt/degree. The velocity of each link is obtained by using the position signal and utilizing first order backward differencing technique.

The feedback signals from the potentiometers and the control signals to the motor drives are sent to/from the computer via PCI-DAS6014 ADDA interface card. The card has a minimum 200 kS/s conversion rate and has an absolute accuracy of 8.984 mV when operates at the range ± 10 V. The control program is written in C++ and executed at 1 ms sampling rate. Figure 4 shows the closed loop control system.

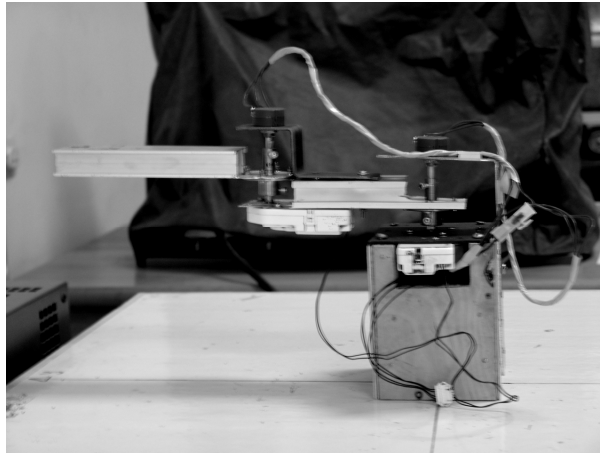


Figure 3 Experimental two-link planar arm

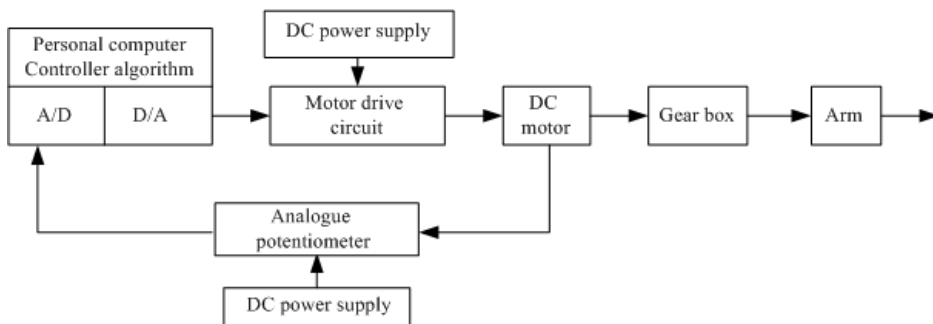


Figure 4 Block diagram of the experimental setup.

5.2 The Reference Trajectories

In order to compare the performance of the two controllers (i.e. PD and FLC) on the two-link manipulator we picked three joint space trajectories. The joint space reference trajectories for each joint were computed as follows:

1) Linear trajectory with parabolic bends

The parabolic blend segments are given by [24]:

$$\theta_d^b(t) = c_0 + c_1 t + c_2 t^2 \quad (12)$$

The linear segment is given by

$$\theta_d(t) = c_3 + c_4 t \quad (13)$$

where $\theta_d^b(t)$ and $\theta_d(t)$ are the desired angular position at any time t of the parabolic segments and linear segment respectively, c_0, c_1, c_2, c_3 and c_4 are constants determined according to the trajectory constraints.

2) Cubic polynomial trajectory

$$\theta_d(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 \quad (14)$$

where c_0, c_1, c_2 and c_3 are constants determined upon the trajectory constraints.

3) Sinusoidal trajectory

$$\theta_d(t) = A \sin(\omega t) \quad (15)$$

where A and ω are the amplitude and frequency of motion, respectively.

5.3 Performance Evaluation (Error Criteria)

In order to compare the performance of the various control algorithms quantitatively we used two error measures. We computed the L_2 norm of the tracking error of each joint in the joint space. These errors are presented in table form in the coming section along with plots of the errors themselves.

Recall that the L_2 norm of the tracking error of joint i is

$$|e_i| := \sqrt{\int_{t_0}^{t_f} (\theta_i(t) - \theta_{di}(t))^2 dt} \quad (16)$$

where $i \in \{1, 2\}$ is the joint number. Since data are only sent back at discrete time intervals, $t_1 \cdots t_k$ with constant sampling period $\Delta T = t_{n-1} - t_n$ for all n ; we discretize (16) as

$$|e_i| := \sqrt{\sum_{j=1}^k (\theta_i(j) - \theta_{di}(j))^2 \Delta T} = \sqrt{\Delta T} \sqrt{\sum_{j=1}^k (\theta_i(j) - \theta_{di}(j))^2}$$

where $\theta_i(j)$ denotes $\theta_i(t_j) = \theta_i(j\Delta T)$. Because ΔT is constant we can include it on the left side which yields our definition of the joint space error criteria

$$|error_i| := \frac{|e_i|}{\sqrt{\Delta T}} = \sqrt{\sum_{j=1}^k (\theta_i(j) - \theta_{di}(j))^2} \quad (17)$$

The maximal error in joint space is given by

$$\max error_i = \max_{1 \leq j \leq k} |\theta_i(j) - \theta_{di}(j)| \quad (18)$$

The above two measures have been also adopted in [25].

6. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, the experiments conducted in this study will be presented. The experimental results include all the three trajectories with and without payload mass. For the sake of comparison, we ran each controller with the same initial conditions to analyze the strength and weakness of each design. To show robustness, the two controllers have been initiated with initial position error equal to 10° , i.e. $\theta(0) = [10^\circ - 10^\circ]^T$ and the robot is at rest, i.e. $\dot{\theta}(0) = [0 \ 0]^T$.

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

$$u(t) = K_p e_p(t) + K_D e_v(t) \quad (19)$$

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. A traditional problem associated with PD control is that we can not increase the controller gains, as much as we want, 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 the values of these gains.

In the experiments, the proportional feedback gains of the PD controller were set to $K_{p1} = 40$, $K_{p2} = 30$ and the derivative gains were chosen to be $K_{D1} = 0.01$, $K_{D2} = 0.005$ for the base and elbow links, respectively. They have been selected as high as possible without violating the stability of the overall system.

With respect to the proposed FLC, the control gains were set to $a_{u1} = a_{u2} = 5$ thus ensuring that the control signals computed according to (11) remain in the range of ± 10 V which is a hardware requirement. The other control parameters were picked as $a_{ep1} = 10$, $a_{ep2} = 9$ and $a_{ev1} = 0.05$, $a_{ev2} = 0.045$. We chose these parameters experimentally after few trials.

With the above mentioned gains, the performance of the two controllers has been also examined in the presence of payload mass whose weight is 205 grams. For

reasons of space, figures related to experiments done in the presence of payload have been omitted. The performance criteria i.e. L_2 norm (17) of the tracking error and the maximal error (18) for all experiments are tabulated in order. These two criteria are accounted for after 2 seconds in order to avoid the transient period and to give more insight on the performance at the steady state. In all of the tables to follow, the lower value in each column is identified in bold to facilitate comparison.

A. Linear trajectory with parabolic bends

In this section we present results for a linear trajectory with parabolic blends. The desired motion is from zero to 45° with the following constraints:

Lower parabolic blend segment:

$\theta_i = 0^\circ$, $\theta^b = 15^\circ$ and $\dot{\theta}_i = 0 \text{ rad/sec}$, where θ_i is the initial angular position and θ^b is end-angle of the blend. The desired trajectory according to (12) becomes:

$$\theta_d^b(t) = 1.28t^2, \quad 0 \leq t \leq 3.42 \text{ sec} \quad (20)$$

Higher parabolic blend segment:

$$\theta^h = 30^\circ, \theta_f = 45^\circ \text{ and } \dot{\theta}_f = 0 \text{ rad/sec}$$

where θ^h is the initial angular position of the higher parabolic blend segment, θ_f is the final angular position. The desired trajectory according to (12) becomes:

$$\theta_d^h(t) = -83 + 25.6t - 1.28t^2, \quad 6.58 \leq t \leq 10 \text{ sec} \quad (21)$$

Linear segment:

$$\theta^b(t = 3.42) = 15^\circ$$

The desired trajectory according to (13) will be:

$$\theta_d(t) = -1.245 + 4.75t, \quad 3.42 \leq t \leq 6.58 \quad (22)$$

The first implemented control algorithm was the PD. The tracking performance for the base (joint 1) and elbow link (joint 2) is shown in Fig. 5, 6 and 7. Results of the FLC are shown in Fig. 8, 9 and 10. As it can be noticed, faster convergence (lower transient period) has been achieved by the FLC. Table 3 summarizes the error criteria mentioned in Section 5. Better tracking has been realized by the FLC as it can be noticed.

B. Cubic polynomial trajectory

In this section we present experimental results for a reference cubic polynomial trajectory. The desired motion is from zero to 45° with the following constraints:

$$\theta_i(t=0) = 0, \theta_f(t=10) = 45^\circ, \dot{\theta}_i = 0 \text{ and } \dot{\theta}_f = 0.$$

The desired trajectory according to (14) will be:

$$\theta_d(t) = 1.35t^2 - 0.09t^3, \quad 0 \leq t \leq 10 \quad (23)$$

The tracking performance of the PD controller is shown in Fig. 11, 12 and 13. Results of the FLC are depicted in Fig. 14, 15 and 16. Faster convergence has been realized by the FLC. Table 4 summarizes the tracking performance of the two controllers. The FLC exhibits low tracking errors as it can be noticed from Table 4.

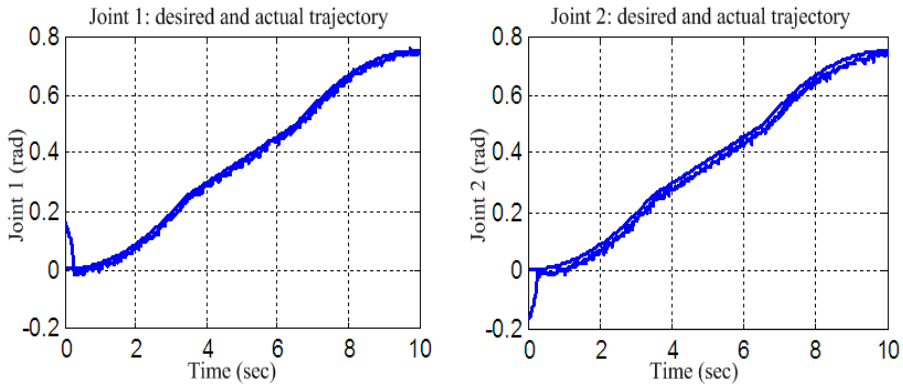


Figure 5. The actual and desired trajectories (PD controller).

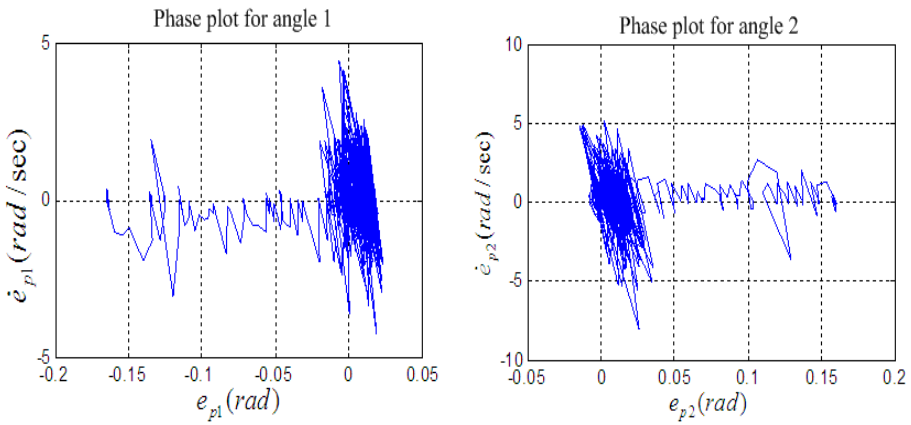


Figure 6. The phase plots (PD controller).

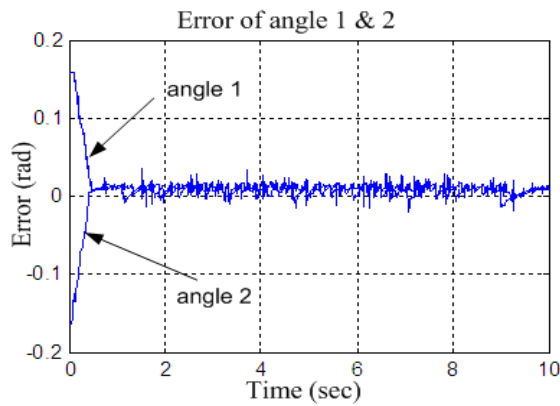


Figure 7. The tracking errors (PD controller).

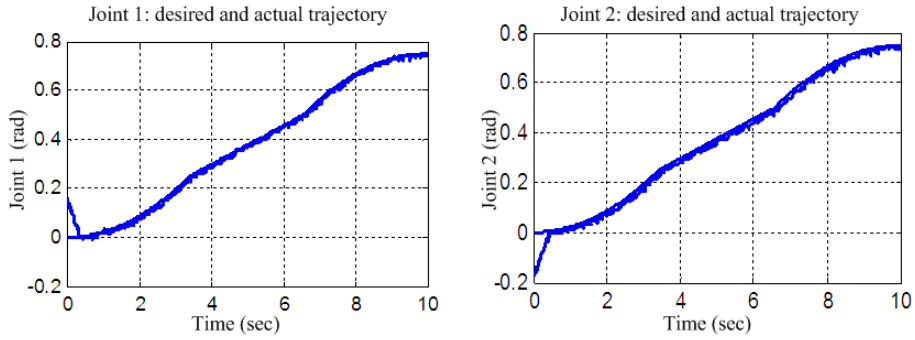


Figure 8. The actual and desired trajectories (FLC).

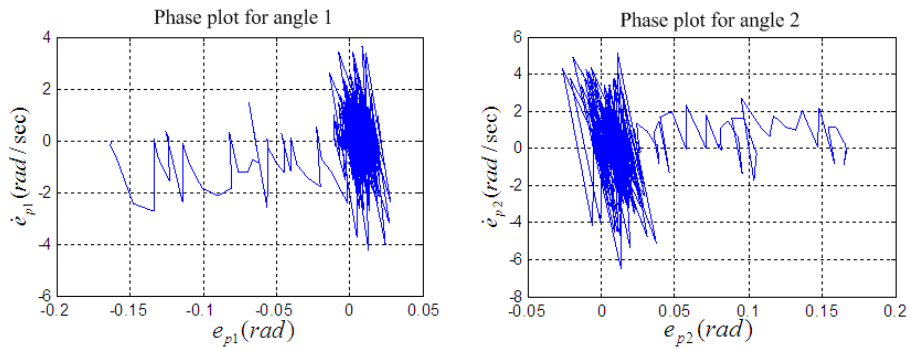


Figure 9. The phase plots (FLC).

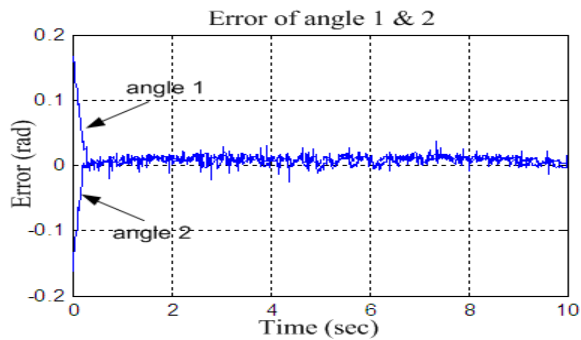


Figure 10. The tracking errors (FLC).

Table 3. Linear trajectory with parabolic blends (after 2 seconds).

Algorithm	$ error1 $	maxerror ₁ (rad)	$ error2 $	maxerror ₂ (rad)
PD	0.0170	0.0281	0.0269	0.0435
Fuzzy	0.0124	0.0220	0.0166	0.0368

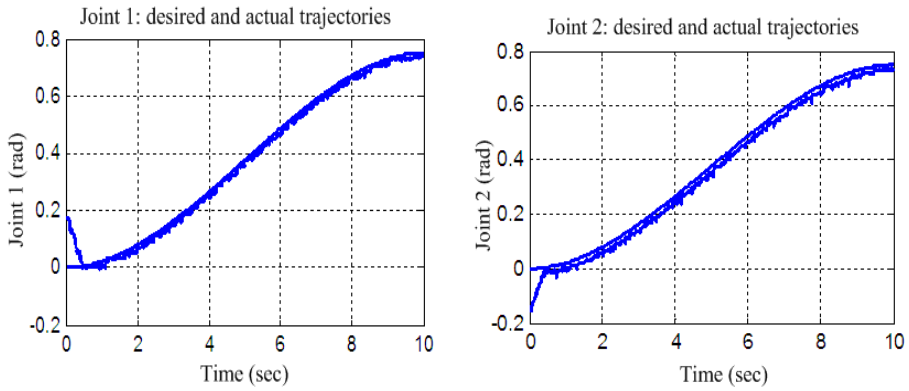


Figure 11. The actual and desired trajectories (PD controller).

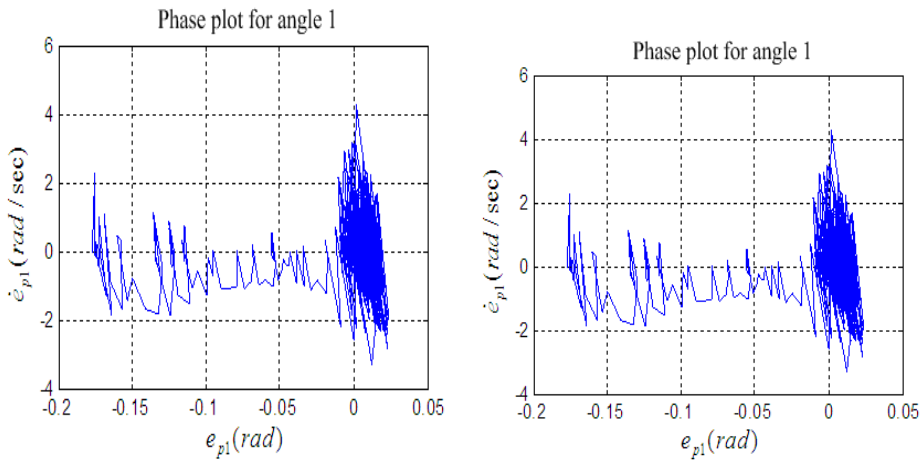


Figure 12. The phase plots (PD controller).

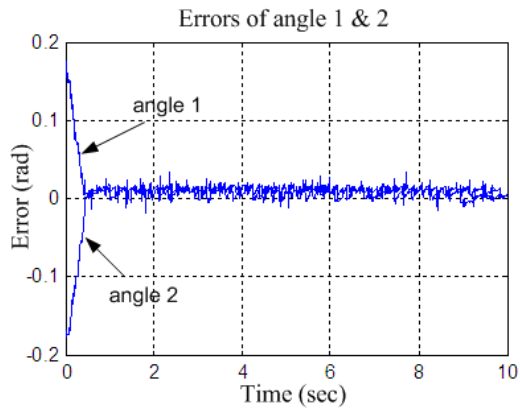


Figure 13. The tracking errors (PD controller).

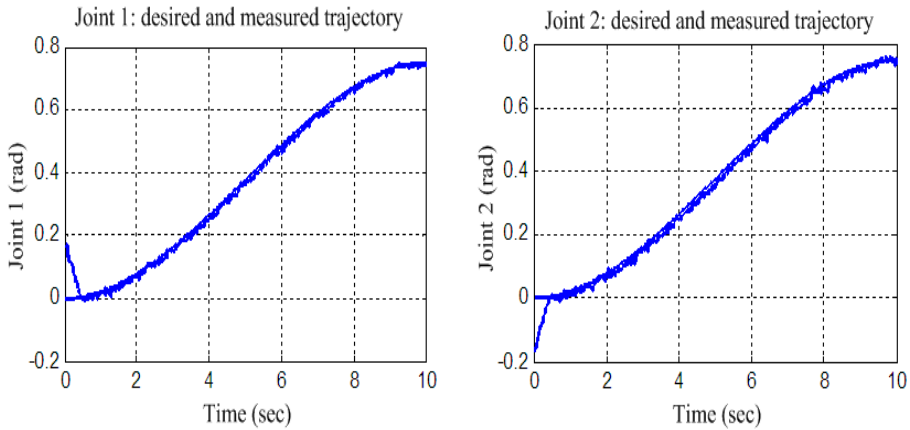


Figure 14. The actual and desired trajectories (FLC).

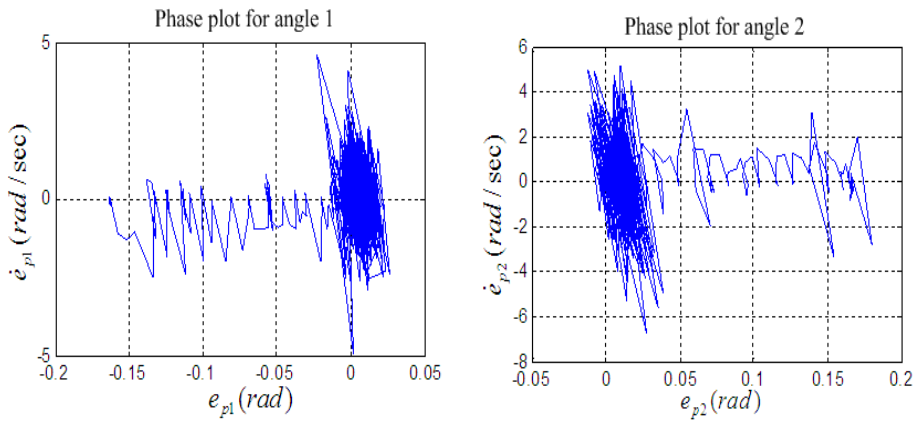


Figure 15. The phase plots (FLC).

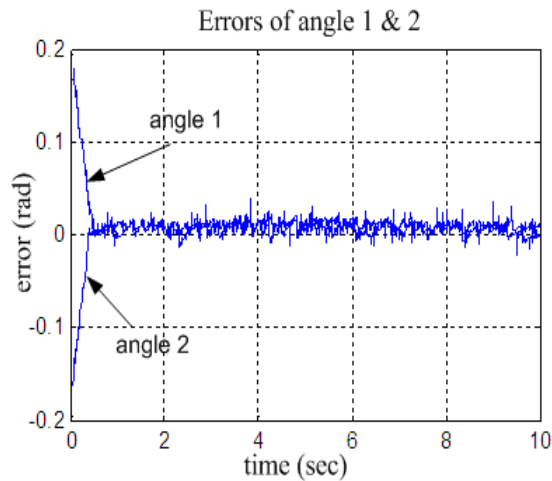


Figure 16. The tracking errors (FLC).

Table 4. Cubic polynomial trajectory (after 2 seconds)

Algorithm	$ error1 $	maxerror ₁ (rad)	$ error2 $	maxerror ₂ (rad)
PD	0.0170	0.0279	0.0272	0.0500
Fuzzy	0.0123	0.0266	0.0156	0.0373

C. Sinusoidal trajectory

In this section we present experimental results of the sinusoidal trajectory. The desired motion for angle 1 and angle 2 has the following constraints:

Amplitude1 = 20°, frequency1 = 0.05 rad/sec, and trajectory period = 20 sec

Amplitude2 = 30°, frequency2 = 0.1 rad/sec, and trajectory period = 20 sec

Thus, the desired trajectory according to (15) is:

$$\theta_{d1}(t) = \pi/9 \sin(0.05 \times 2\pi \times t), \quad (24)$$

$$\theta_{d2}(t) = \pi/6 \sin(0.1 \times 2\pi \times t) \quad (25)$$

The tracking performance is shown in Fig. 17, 18 and 19 for the PD controller. The figures show very good tracking performance. Figures 20, 21 and 22 show the results of the proposed FLC. As it can be noticed, lower transient period has been achieved by the FLC. The error criteria tabulated in Table 5 also show that lower tracking errors have been realized by the FLC.

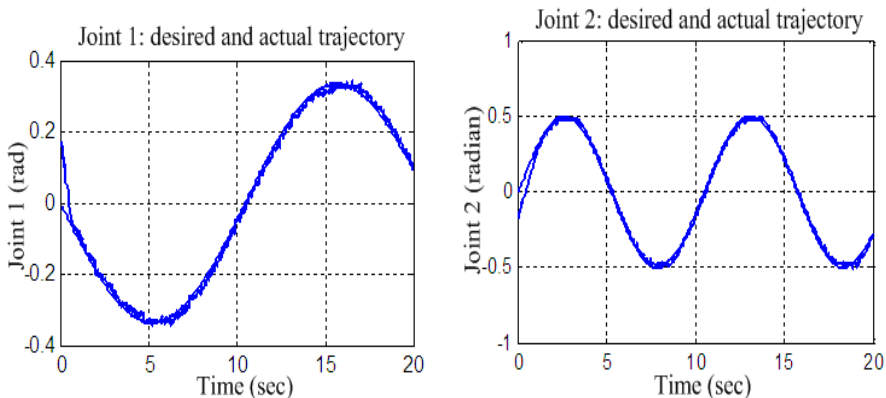


Figure 17. The actual and desired trajectories (PD controller).

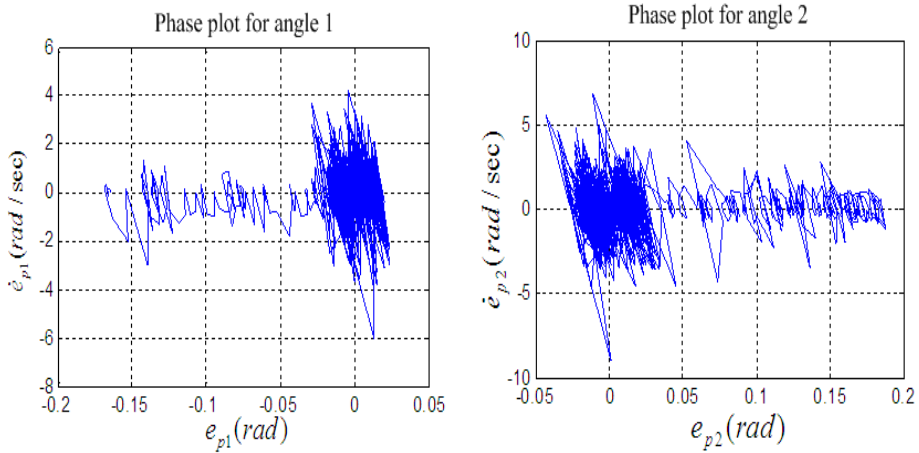


Figure 18. The phase plots (PD controller).

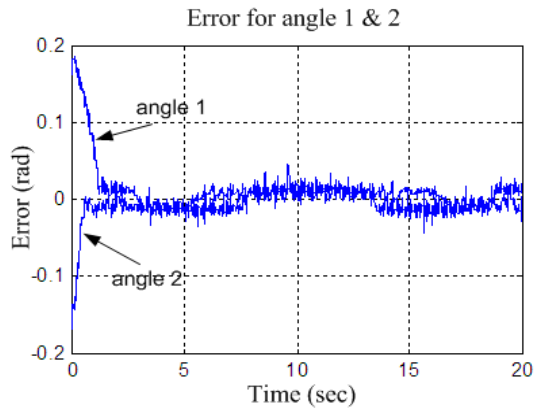


Figure 19. The tracking errors (PD controller).

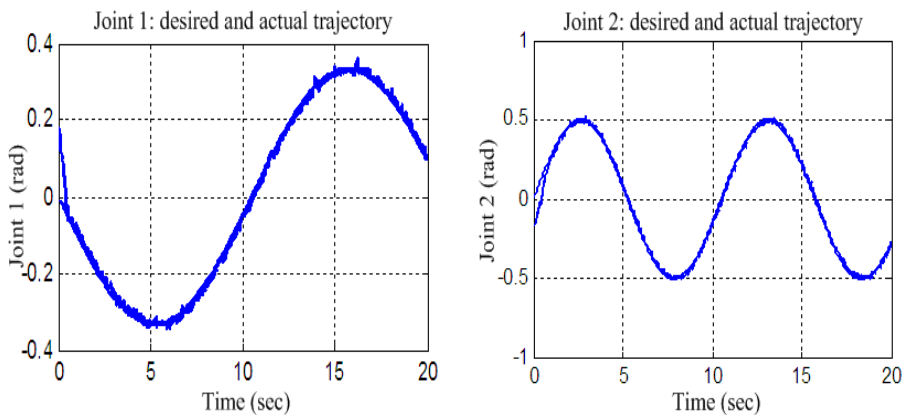


Figure 20. The actual and desired trajectories (FLC).

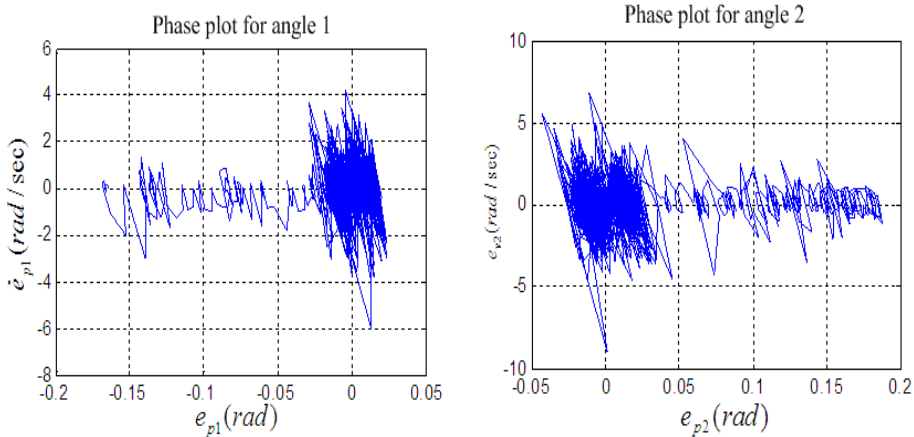


Figure 21. The phase plots (FLC).

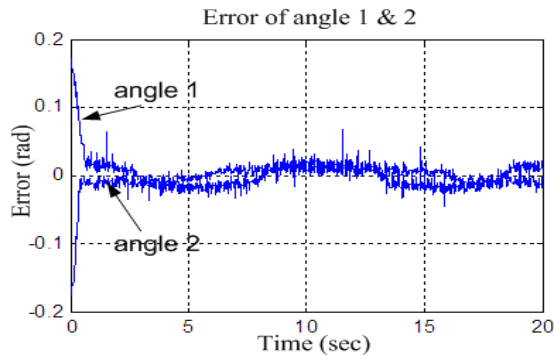


Figure 22. The tracking errors (FLC).

Table 5. Sinusoidal Trajectory (after 2 seconds).

Algorithm	$ error1 $	maxerror ₁ (rad)	$ error2 $	maxerror ₂ (rad)
PD	0.0234	0.0301	0.0402	0.0508
Fuzzy	0.0182	0.0293	0.0338	0.0470

D. Experiments with payload mass

This section demonstrates the experimental results of adding a payload mass of 205 gram at the tip of the elbow link. The two controllers have been examined by implementing the three trajectories in the presence of the payload mass. Results are tabulated in Table 6, 7 and 8. They show that trajectory tracking has been achieved by the two controllers. Better tracking performance has been achieved by the FLC. Also, the transient period in case of the FLC is lower than that of the PD controller in all the three trajectories (the related figures are not presented here to save space).

Table 6. Linear trajectory with parabolic blends (after 2 seconds).

Algorithm	$ error1 $	maxerror ₁ (rad)	$ error2 $	maxerror ₂ (rad)
PD	0.0166	0.0327	0.0265	0.0496
Fuzzy	0.0121	0.0246	0.0152	0.0385

Table 7. Cubic trajectory (after 2 seconds).

Algorithm	$ error1 $	maxerror ₁ (rad)	$ error2 $	maxerror ₂ (rad)
PD	0.0158	0.0276	0.0254	0.0396
Fuzzy	0.0121	0.0277	0.0161	0.0373

Table 8. Sinusoidal trajectory (after 2 seconds)

Algorithm	$ error1 $	maxerror ₁ (rad)	$ error2 $	maxerror ₂ (rad)
PD	0.0226	0.0311	0.0416	0.0447
Fuzzy	0.0171	0.0303	0.0308	0.0460

From the previous figures and tables, the following remarks are in order:

- It is shown from tables 3, 4, 5, 6, 7, 8 that the overall tracking performance with the fuzzy controller is better than that of the PD.
- When applying initial position error, it has been noticed that the fuzzy controller converge faster than the PD controller.

Summarizing, it can be said that the PD controller provides high performance in trajectory tracking and therefore proposed new solutions have necessarily to be compared with this form of PID controllers in order to show their effectiveness. The proposed FLC surely represents a very valuable approach in this context, taking into account the easiness of the implementation.

7. CONCLUSIONS

Since extracting knowledge from experts in many cases is a tedious task, one would assume basic physical information about the system. We have implemented the Lyapunov second method to get such basic information and designed a fuzzy control law so that the system is stable in the sense of Lyapunov. Thus, greatly simplifies the extraction of the fuzzy rules. The important feature of the study is that it has transferred the proposed fuzzy PD controller into a closed form relation (hyperbolic tangential

function) between the inputs and the output. Relative to other works in this area, the number of parameters needs to be tuned is quite small which has greatly facilitated the implementation. Unlike the PD controller, the proposed FLC is inherently bounded and the upper bound can be selected by suitably adjust its parameters.

The presented approach provides a systematic step by step procedure for the design of fuzzy based feedback controllers for a wide class of second order nonlinear systems. The methodology has been applied to the control of a two-link robot. It can also be extended to n number of link robots. Experimental results show that the design procedure has been successful in representing the nonlinear dynamics in the control context and resulted in a stable closed-loop system. Higher tracking performance and faster convergence of errors have been achieved by the proposed FLC relative to the conventional PD controller.

REFERENCES

- [1] L. Tian and C. Collins, "Optimal placement of a two-link planar manipulator using a genetic algorithm," Cambridge University Press, *Robotica* Vol. 23 (2005), pp. 169–176.
- [2] J. J. Craig, "Introduction to Robotics: Mechanics and Control," Addison-Wesley Inc., 2nd ed., 1989.
- [3] F. Reyes, R. Kelly, "Experimental evaluation of model-based controllers on a direct-drive robot arm," *PERGAMON, Mechatronics* 11 (2001) 267-282.
- [4] J. Slotine and Li, "Applied Nonlinear Control," Prentice-Hall, Inc., 1991.
- [5] B. Brogliato, D. Rey, A. Pastore and J. Barnier, "Experimental comparison of nonlinear controllers for flexible joint manipulators," *The International Journal of Robotics Research*, Vol. 17, No. 3 (1998), pp. 260-281.
- [6] F. Alonge, F.D'Ippolito, F.M. Raimondo, "Globally convergent adaptive and robust control of robotic manipulators for trajectory tracking," *Control Engineering Practice*, 12 (2004), pp. 1091-1100.
- [7] M. K. Ciliz and M. O. Tuncay, "Comparative Experiments with a Multiple Model Based Adaptive Controller for a SCARA Type Direct Drive Manipulator", *Robotica* Vol. 23 (2005), pp. 721-729.
- [8] L.X. Wang, *Adaptive fuzzy systems and control*, Printice-Hall, Englewood Cliffs, NJ, 1994.
- [9] B. Kosko, *Neural networks and fuzzy systems: A dynamical system approach to machine intelligence*, Prentice Hall, 1991.
- [10] L.A Zadeh, "Fuzzy Logic = Computing with words", *IEEE Trans. Fuzzy Systems* Vol. 4(2) (1996), pp. 103-111.
- [11] M. Margaloit, and G. Langholz, "Fuzzy Lyapunov-based Approach to the Design of Fuzzy Controllers", *Fuzzy Sets and Systems*, 106 (1999), pp. 49-59.
- [12] M. Margaloit, and G. Langholz, "Hyperbolic Optimal Control and Fuzzy Control", *IEEE Trans. on Systems, Man and Cybernetics-Part A: Systems and Humans*, Vol. 29, No. 1 (1999), pp. 1-10.
- [13] M. Margaloit and G. Langholz, "Fuzzy Control of a Benchmark Problem: Computing with Words Approach", *IEEE Trans. on Fuzzy Systems*, Vol. 12, No. 2 (2004), pp. 230-235.

-
- [14] T.L. Seng, M. Khalid and R. Yusof, "Tuning of a Neuro-Fuzzy Controller by Genetic Algorithm", *IEEE Transaction on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 29, No. 2 (1999), pp. 226-239.
 - [15] V. Giordano, D. Naso, and B. Turchiano, "Combining genetic algorithms and Lyapunov-based adaptation for online design of fuzzy controllers", *IEEE Trans. On Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 36, No. 5 (2006), pp. 1118-1127.
 - [16] O. Cordon, F. Herrera, F. Hoffmann and L. Magdalena, "Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases", World Scientific, Singapore, July 2001.
 - [17] P.T. Chan, X.F. Xie and A.B. Rad, "Tuning of fuzzy controller for an open-loop unstable system: a genetic approach", *Fuzzy Sets and Systems* 111 (2000), pp. 137-152.
 - [18] M. A. Llama, R. Kelly and V. Santibanez, "A stable motion control system for manipulators via fuzzy self-tuning," *Fuzzy Sets and Systems*, 124 (2001), pp. 133-154.
 - [19] A. B. Sharkawy, "A computationally efficient fuzzy logic controller for robotic systems," 9th International Conference on Production Engineering, Design and Control, PEDAC'2009, Alexandria, Egypt, Feb. 10-12, 2009.
 - [20] M. W. Spong and M. Vidyasagar, "Robot Dynamics and Control." Wiley, New York, 1989.
 - [21] L.X. Wang, "Fuzzy systems are universal approximations", in: *Proc. IEEE International Conference on Fuzzy Systems*, pp. 1163-1170, 1992.
 - [22] L.X. Wang, *Adaptive fuzzy systems and control*, Printice-Hall, Englewood Cliffs, NJ, 1994.
 - [23] L.X. Wang, *A Course in Fuzzy Systems and Control*. Upper Saddle River, NJ: Prentice-Hall, 1997.
 - [24] J. J. Craig, "Introduction to Robotics: Mechanics and Control," Addison-Wesley Inc., 2nd ed., 1989.
 - [25] A Jaritz and M. W. Spong, " An Experimental Comparison of Robust Control Algorithms on a Direct Drive Manipulator," *IEEE Transactions on Control Systems Technology*, Vol. 4, no. 6, November 1996.

تتبع المسار باستخدام التحكم الفازي لذراع آلي ذو درجتين حرية م. محمود محمد عثمان - د. عبد البديع شرقاوي - أ.د. أبو المكارم أحمد خليل

هذا البحث يقوم بدراسة تتبع المسار لذراع آلي ذو درجتين حرية، وقد تم عمل دراسة مقارنة عملية على نوعين من التحكم هما التحكم التناسبي التفاضلي (PD) والتحكم الفازي (FLC)، وقد تم تطبيق طريقة للتحكم الفازي تعتمد على تقليل الحسابات بشكل فعال، وقد تم اختبار هذين النوعين لتتبع مسار الذراع الآلي على ثلاثة مسارات مختلفة وهي: مسار الخط المستقيم الذي يبدأ وينتهي بمعادلة من الدرجة الثانية ومسار الدالة التكميلية كثيرة الحدود ومسار دالة الجيب.

وبمقارنة النتائج وجد أن المتحكم الفازي يتميز بنسبة خطأ أقل من المتحكم التناسبي التفاضلي على طول المسار بينما يتميز المتحكم التناسبي التفاضلي بأن أكبر خطأ في تتبع المسار لديه أقل من المتحكم الفازي.

ولزيادة التأكد من مدى فاعلية نظام التحكم الفازي المقترح في هذه الدراسة قمنا بعمل مجموعة من التجارب العملية تم خلالها وضع حمل إضافي على النهاية الطرفية للذراع الآلي. وقد أظهرت النتائج تقاربا بين المتحكمين (PD & FLC) الذين شملتهما الدراسة.