

A Robust Fuzzy Tracking Control Scheme for Robotic Manipulators with Experimental Verification

Abdel Badie Sharkawy*, Mahmoud M. Othman, Abouel Makarem A. Khalil
Mechanical Engineering Department, Faculty of Engineering, Assiut University, Assiut, Egypt
E-mail: ab.shark@aun.edu.eg

Received March 19, 2011; revised April 2, 2011; accepted April 7, 2011

Abstract

The performance of any fuzzy logic controller (FLC) is greatly dependent on its inference rules. In most cases, the closed-loop control performance and stability are enhanced if more rules are added to the rule base of the FLC. However, a large set of rules requires more on-line computational time and more parameters need to be adjusted. In this paper, a robust PD-type FLC is driven for a class of MIMO second order nonlinear systems with application to robotic manipulators. The rule base consists of only four rules per each degree of freedom (DOF). 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. Furthermore, the controller is decoupled and the procedure is simplified 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 for the following controllers: the conventional PD controller, computed torque controller (CTC), sliding mode controller (SMC) and the proposed FLC. The four controllers are tested and compared with respect to ease of design, implementation, and performance of the closed-loop system. Results show that the proposed FLC has outperformed the other controllers.

Keywords: Fuzzy Logic Control (FLC), PD Control, Computed-Torque Control (CTC), Sliding Mode Control (SMC), Lyapunov Synthesis, Test Rig

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, [1-3]. 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 approach, [4]. 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 [5], 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 pre-

dicted in advance due to different operating conditions.

Sliding mode controllers (SMCs) were first proposed in early 1950s. Due to their good robustness to uncertainties, SMC has been accepted as an efficient method for robust control of uncertain systems. Being limited only by practical constraints on the magnitude of control signals, the sliding mode controller, in principle, can treat a variety of uncertainties as well as bounded external disturbances, [6]. A key step in the design of controllers is to introduce a proper transformation of tracking errors to generalized errors so that an n-order tracking problem can be transformed into an equivalent first-order stabilization problem. Since the equivalent first-order problem is likely to be simpler to handle, a control law may thus be easily developed to achieve the so-called reaching condition. Unfortunately, an ideal sliding mode controller inevitably has a discontinuous switching function. Due to imperfect switching in practice it will raise

the issue of chattering, which is highly undesirable. To suppress chattering, a continuous approximation of the discontinuous sliding control is usually employed in the literature. Although chattering can be made negligible if the width of the boundary layer is chosen large enough, the guaranteed tracking precision will deteriorate if the available control bandwidth is limited, [7]. A number of works related to sliding mode control of robotic manipulators have been published in [8-12].

Generally speaking, multiple-input multiple-output (MIMO) systems usually have characteristics of nonlinear dynamics coupling. Therefore, the difficulty in controlling MIMO systems is how to overcome the coupling effects between the degrees of freedom. The computational burden and dynamic uncertainty associated with MIMO systems make model-based decoupling impractical for real-time control. 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. 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, [13]. Because many design parameters (like learning rates and initialization of the parameters to be adapted) have to be considered in controller construction, most existing methodologies have limitations. Moreover, owing to the different characteristics among design parameters, attaining a complete learning, while considering an overall performance goal, is an extremely difficult task. Nevertheless, some experiments have been presented in [14,15].

Fuzzy controllers have demonstrated excellent robustness in both simulations and real-life applications, [16]. 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.

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, [17-19]. 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 [20], the controller needs 26 parameters to be experimentally selected. Also, the FLC in [21] needs 45 parameters to be tuned. This is beside the huge number of calculations involved in the online computation of the control signals.

In this research paper, we introduce a simple and computationally efficient FLC for MIMO second order systems with application to robotic manipulators. Earlier theoretical investigation of this controller, by the first author, can be found in [22]. 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, computed-torque controller (CTC), and sliding mode controller (SMC).

The rest of this paper is organized as follows. Section 2 presents the model based controllers (CTC and SMC) that are used for comparison purposes. The proposed control scheme is introduced in Section 3 and Section 4 describes the experimental setup, the examined trajectories and the performance measures used in the control performance evaluation. The experimental results are demonstrated and discussed in Section 5. Section 6 offers our concluding remarks.

2. Model-Based Control Schemes

2.1. Preliminaries

The dynamic model of an n -joint manipulator can be written as follows:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + f(\dot{q}) = u(t) \quad (1)$$

where q is the $n \times 1$ joint angle vector, $u(t)$ is the $n \times 1$ input torque vector, $M(q)$ is the $n \times n$ positive definite inertia matrix, $C(q, \dot{q})$ is the $n \times n$ matrix representing the centrifugal and Coriolis terms, $f(\dot{q})$ is the $n \times 1$ vector of the frictional terms and $g(q)$ is the $n \times 1$ vector of the gravity terms.

Decoupled or decentralized (also independent) control means that the torque u_i to be generated by the i th actuator is based only on the value of the position of the i th joint and its time derivative

$$u_i = u_i(q_i, \dot{q}_i, q_{di}, \dot{q}_{di}), \quad i = 1, 2, \dots, n, \quad (2)$$

where q_i is the actual value of the i th joint coordinate,

and q_{di} is its desired value. The later (q_{di}) is usually available signal from the robot operating system and is planned in advance. Generally, defining the position error as $e_i = q_{di} - q_i$, (2) can be written as

$$u_i = u_i(e_i, \dot{e}_i), \quad i = 1, 2, \dots, n \quad (3)$$

This approach is widely adopted in industrial settings because of its simplicity (no dynamic model is required, in general) and because of its fault-tolerant feature, since, in case a single joint is affected by a failure, the robot can be retrieved in a safe position by means of the other joints.

The motion control problem of manipulators in joint space can be stated in the following terms. Assume that the joint position q and the joint velocity \dot{q} are available for measurement. Let the desired joint position q_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:

$$\lim_{t \rightarrow \infty} q(t) = q_d(t) \quad (4)$$

2.2. Computed Torque Control

The computed torque (also called inverse dynamics) technique is a special application of feedback linearization of nonlinear systems. The computed torque controller is utilized to linearize the nonlinear equation of robot motion by cancellation of some, or all, nonlinear terms. For this purpose, the dynamic model of the manipulator is exploited. Taking into account (1) and defining

$$h(q, \dot{q}) = C(q, \dot{q})\dot{q} + g(q) + f(\dot{q}) \quad (5)$$

we can derive the control scheme shown in **Figure 1**, where K_p and K_v are user-chosen diagonal matrices. So that the system is decoupled, linearized and the error dynamics is governed by the following expression [23]:

$$\ddot{e} + K_v \dot{e} + K_p e = 0 \quad (6)$$

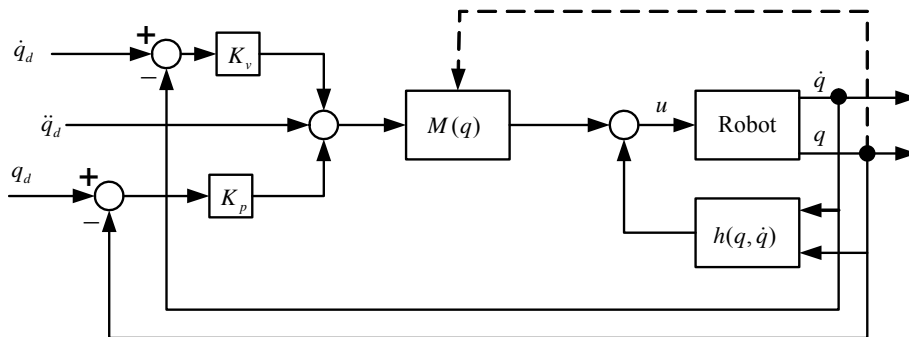


Figure 1. Computed torque control scheme.

where, again, $e = q_d - q$ is the difference between the desired joint position vector and the actual one.

Obviously, the assumption of exact knowledge of the robot dynamic model cannot be satisfied in practical cases. Hence, the achievement of the desired tracking performance cannot be guaranteed. For this purpose, it would be desirable to add a term in the controller that compensates for the modeling errors. Several related works can be found in literature which suggests the use of neural networks [23] and neuro-fuzzy systems [24] in-order to compensate for the modeling errors. However, a complete review in this area is out of the scope of this work. In the experimental verification (Section 5), the CTC algorithm has been implemented as it is shown in **Figure 1**.

2.3. Sliding Mode Control

In this subsection, the well-developed literature is used to demonstrate the main features and assumptions needed to synthesis a SMC for robotic systems. SMC employs a discontinuous control effort to derive the system trajectories toward a sliding surface, and then switching on that surface. Accordingly, it will gradually approach the control objectives, *i.e.* keep these trajectories at the origin of the phase plane. The following assumptions are needed to synthesis a SMC.

Assumption 1: The matrix $M(q)$ is positive definite and is bounded by a known positive definite matrix $\hat{M}(q)$.

Assumption 2: There exists a known estimate $\hat{h}(q, \dot{q})$ for the vector function $h(q, \dot{q})$ in (5).

Now, let us define the linear time-varying surface $s(t)$, [9]:

$$\begin{aligned} s(t) &= \dot{e}(t) + \beta(t), \\ s(t) &= [s_1(t), s_2(t), \dots, s_n(t)]^T \end{aligned} \quad (7)$$

where $\beta(t)$ is a time-varying linear function.

From (1), (5) and (7), we can get the equivalent controller (also called ideal controller)

$$u_{eq}(t) = h(q, \dot{q}) + M(q) [\ddot{q}_d + \dot{\beta}] \quad (8)$$

where $u_{eq}(t)$ is equivalently the average value of $u(t)$ which maintains the system's trajectories (*i.e.* tracking errors) on the sliding surface $s(t) = 0$. To ensure that they attain the sliding surface in a finite time and thereafter maintain there, the control torque $u(t)$ consists of a low frequency (average) component $u_{eq}(t)$ and a hitting (high frequency) component $u_{ht}(t)$ so that:

$$u(t) = u_{eq}(t) + u_{ht}(t) \quad (9)$$

The role of $u_{ht}(t)$ acts to overcome the effects of the uncertainties and bends the entire system trajectories towards the sliding surface until sliding mode occurs. The hitting controller $u_{ht}(t)$ is taken as:

$$u_{ht}(t) = M(q) K \operatorname{sgn}(s) \quad (10)$$

where $K = \operatorname{diag}(k_1, k_2, \dots, k_n)$, $k_i > 0$, $i = 1, 2, \dots, n$ and $\operatorname{sgn}(s) = [\operatorname{sgn}(s_1), \operatorname{sgn}(s_2), \dots, \operatorname{sgn}(s_n)]^T$.

To verify the control stability, let us first get an expression for $\dot{s}(t)$. Using (1), (5), (9) and (10), the first derivative of (7) is:

$$\begin{aligned} \dot{s}(q, t) &= \ddot{e}(t) + \dot{\beta}(t) \\ &= \ddot{q}_d(t) - \ddot{q}(t) + \dot{\beta}(t) \\ &= \ddot{q}_d(t) - M^{-1}(q) [u(t) - h(q, \dot{q})] + \dot{\beta}(t) \\ &= -M^{-1}(q) u_{ht} \\ &= -K \operatorname{sgn}(s) \end{aligned} \quad (11)$$

Choosing a Lyapunov function

$$V_s = \sum_{i=1}^n \frac{1}{2} s_i^2(t) \quad (12)$$

and differentiating (12) using (11)

$$\begin{aligned} \dot{V}_s &= \sum_{i=1}^n s_i(t) \dot{s}_i(t) = -\sum_{i=1}^n k_i s_i(t) \operatorname{sgn}(s_i(t)) \\ &= -\sum_{i=1}^n k_i |s_i(t)| \leq 0. \end{aligned} \quad (13)$$

which provides an asymptotically stable system.

Since the parameters of (1) and (5) depend on the manipulator structure, it is difficult to obtain completely accurate values for $M(q)$ and $h(q, \dot{q})$. In SMC theory, estimated values are usually used in the control context instead of the exact parameters. So that, (8)-(10) can be written as:

$$\begin{aligned} u_{eq}(t) &= \hat{h}(q, \dot{q}) + \hat{M}(q) [\ddot{q}_d + \dot{\beta}], \\ u_{ht}(t) &= \hat{M}(q) K \operatorname{sgn}(s), \\ u(t) &= u_{eq}(t) + u_{ht}(t) \end{aligned} \quad (14)$$

where $\hat{M}(q)$ and $\hat{h}(q, \dot{q})$ are bounded estimates for

$M(q)$ and $h(q, \dot{q})$ respectively. As mentioned earlier in assumption 1 and 2, they are assumed to be known in advance.

In sliding mode, the system trajectories are governed by:

$$s_i(t) = 0 \quad \text{and} \quad \dot{s}_i(t) = 0, \quad i = 1, 2, \dots, n \quad (15)$$

So that, the error dynamics are determined by the function $\beta(t)$. If the coefficients of $\beta(t)$ were chosen to correspond to the coefficients of Hurwitz polynomial, it is thus implying that $\lim_{t \rightarrow \infty} e(t) = 0$. This suggests $\beta(t)$ taking the following form:

$$\beta(t) = c_1 e(t) + c_2 \int e(t) dt \quad \text{with} \quad c_1, c_2 > 0 \quad (16)$$

So that in the sliding mode, the error dynamics is:

$$\ddot{e}(t) + c_1 \dot{e}(t) + c_2 e(t) = 0, \quad (17)$$

and the desired performance is governed by the coefficients c_1 and c_2 .

In summary, the sliding mode controller in (14) can guarantee the stability in the Lyapunov sense even under parameter variations. As a result, the system trajectories are confining to the sliding surfaces (7). The control law (14) however, shows that the coupling effects have not been eliminated since the control signal for each degree of freedom is dependent on the dynamics of the other degrees of freedom. Independency is usually preferred in practice. Furthermore, to satisfy the existence condition of the sliding modes, a large uncertainty bound should be used. In this case, the controller results in large implementation cost and may lead to chattering efforts which should be avoided in practical implementation.

3. The Proposed FLC Scheme

3.1. Lyapunov Based Fuzzy Logic Controllers

In this section, we apply the fuzzy synthesis [25], to the design of stable controllers. To this end, consider a class of MIMO nonlinear second order systems whose dynamic equation can be expressed as:

$$\ddot{x}(t) = f(x, \dot{x}, u), \quad (18)$$

where $f(x, \dot{x}, u)$ is an unknown continuous function, u is the control input and $x(t) = [x_1^1, x_1^2, \dots, x_1^n]^T$ is the state

vector and $\dot{x} = \frac{dx}{dt} = [x_2^1, x_2^2, \dots, x_2^n]^T$, where $\dot{x}_1^i = x_2^i$, $i = 1, 2, \dots, n$. We now seek a smooth Lyapunov function $V: R^n \rightarrow R^n$ for the continuous feedback model (18) 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 a generalized

Lyapunov candidate function of the following quadratic form:

$$V(x, t) = \frac{1}{2} x^T x + \frac{1}{2} \dot{x}^T \dot{x} \quad (19)$$

Differentiating (19) with respect to time gives

$$\begin{aligned} \dot{V}(x, t) = & x_1^1 \dot{x}_1^1 + x_1^2 \dot{x}_1^2 + \dots + x_1^n \dot{x}_1^n \\ & + x_2^1 \dot{x}_2^1 + x_2^2 \dot{x}_2^2 + \dots + x_2^n \dot{x}_2^n \end{aligned}$$

So that

$$\begin{aligned} \dot{V}(x, t) = & x_1^1 x_2^1 + x_1^2 x_2^2 + \dots + x_1^n x_2^n \\ & + x_2^1 \dot{x}_2^1 + x_2^2 \dot{x}_2^2 + \dots + x_2^n \dot{x}_2^n \end{aligned}$$

From which

$$\begin{aligned} \dot{V}(x, t) = & (x_1^1 x_2^1 + x_2^1 \dot{x}_2^1) + (x_1^2 x_2^2 + x_2^2 \dot{x}_2^2) \\ & + \dots + (x_1^n x_2^n + x_2^n \dot{x}_2^n) \end{aligned}$$

This is equal to

$$\dot{V}(x, t) = \dot{V}_1 + \dot{V}_2 + \dots + \dot{V}_n \quad (20)$$

where

$$\dot{V}_i(x, t) = x_1^i x_2^i + x_2^i \dot{x}_2^i, \quad i = 1, 2, \dots, n$$

Then the standard results in Lyapunov stability theory imply that the dynamic system (18) has a stable equilibrium $x = x_e$ if each \dot{V}_i in (20) is ≤ 0 along the system trajectories. To achieve this, we have chosen the control $u_i(x)$ to be proportional to \dot{x}_2^i . Next, our controller design is achieved if we determine a fuzzy control $u(x)$ so that:

$$\dot{V}_i(x, t) = x_1^i x_2^i + \alpha_i x_2^i u_i(x) \leq 0, \quad i = 1, 2, \dots, n \quad (21)$$

where α_i is a positive constant. The results of Wang [26] state that, a fuzzy system that would approximate (21) exists. To this end, one would consider the state vector $x(t)$ and $\dot{x}(t)$ to be the inputs 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^i is (*lv*) and/or x_2^i is (*lv*) THEN u_i is (*lv*)

where the (*lv*) 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}_2 . This assumption can be justified for a large class of second order nonlinear mechanical systems, [27,28]. 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 coming sub-section, we use this approach to design a PD-type FLC for the tracking control problem of robotic systems.

3.2. Robotic Fuzzy Tracking Control

Robots are familiar examples of trajectory-following mechanical systems. Their nonlinearities and strong coupling of the robot dynamics present a challenging control problem. 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.

We now apply the approach presented in the previous subsection 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

$$V = \frac{1}{2} (e^T e + \dot{e}^T \dot{e}) \quad (22)$$

where again, $e(t) = q_d(t) - q(t)$, $\dot{e}(t) = \dot{q}_d(t) - \dot{q}(t)$ and $q_d(t)$ and $\dot{q}_d(t)$ are vectors of the desired joint position and velocity respectively. Differentiating with respect to time and using (20) gives

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

To enforce asymptotic stability, it is required to find u so that

$$\dot{V}_i = e_i \dot{e}_i + \dot{e}_i \ddot{e}_i \leq 0 \quad (23)$$

in some neighborhood of the equilibrium of (22). Taking the control u to be proportional to \ddot{e} , (23) can be rewritten as:

$$\dot{V}_i = e_i \dot{e}_i + \alpha_i \dot{e}_i u_i \leq 0 \quad (24)$$

where α_i is positive constant, $i = 1, \dots, n$. Sufficient conditions for (24) to hold can be stated as follows.

- 1) if, for each $i \in [1, \dots, n]$, e_i and \dot{e}_i have opposite signs and u_i is zero, inequality (24) holds;
- 2) if e_i and \dot{e}_i are both positive, then (24) will hold if u_i is negative; and
- 3) if e_i and \dot{e}_i are both negative, then (24) will hold if u_i is positive. $i \in [1, \dots, n]$ denotes the joint number.

Using these observations, one can easily obtain the four rules listed below in **Table 1**.

Table 1. Fuzzy rules for the tracking controller.

		\dot{e}_i		
		P	N	
e_i	P	u_N	u_Z	$i=1, \dots, n$
	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 , \dot{e} and u which follow directly from the stabilizing conditions of the Lyapunov function, (22).

In concluding words, the presented approach transforms classical Lyapunov synthesis from the world of exact mathematical quantities to the world of words [16]. 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

$$\begin{aligned}\mu_{positive}(x) &= G(x, a_z) = \exp\left[-(x - a_z)^2\right] \\ \mu_{negative}(x) &= G(x, -a_z) \\ \mu_{zero}(x) &= G(x, 0)\end{aligned}$$

where $a_z > 0$ and z stands for control variable, the product for “and” and center of gravity inferencing.

For some positive constants a_u , a_{ep} and a_{ev} , the above four rules can be represented by the following equation:

$$\begin{aligned}u_i &= \frac{G(e_i, a_{epi})(-a_{ui}) + G(e_i, -a_{epi})(a_{ui})}{G(e_i, a_{epi}) + G(e_i, -a_{epi})} \\ &+ \frac{G(\dot{e}_i, a_{evi})(-a_{ui}) + G(\dot{e}_i, -a_{evi})(a_{ui})}{G(\dot{e}_i, a_{evi}) + G(\dot{e}_i, -a_{evi})}\end{aligned}$$

in more details

$$\begin{aligned}u_i &= -a_{ui} \left[\frac{\exp\left(-\left(e_i - a_{epi}\right)^2\right) - \exp\left(-\left(e_i + a_{epi}\right)^2\right)}{\exp\left(-\left(e_i - a_{epi}\right)^2\right) + \exp\left(-\left(e_i + a_{epi}\right)^2\right)} \right] \\ &- a_{ui} \left[\frac{\exp\left(-\left(\dot{e}_i - a_{evi}\right)^2\right) - \exp\left(-\left(\dot{e}_i + a_{evi}\right)^2\right)}{\exp\left(-\left(\dot{e}_i - a_{evi}\right)^2\right) + \exp\left(-\left(\dot{e}_i + a_{evi}\right)^2\right)} \right]\end{aligned}$$

from which

$$\begin{aligned}u_i &= -a_{ui} \left[\frac{\exp(2a_{epi}e_i) - \exp(-2a_{epi}e_i)}{\exp(2a_{epi}e_i) + \exp(-2a_{epi}e_i)} \right] \\ &- a_{ui} \left[\frac{\exp(2a_{evi}\dot{e}_i) - \exp(-2a_{evi}\dot{e}_i)}{\exp(2a_{evi}\dot{e}_i) + \exp(-2a_{evi}\dot{e}_i)} \right]\end{aligned}$$

This yields the FLC

$$u_i = -a_{ui} \left[\tanh(2a_{epi}e_i) + \tanh(2a_{evi}\dot{e}_i) \right], i = 1, \dots, n \quad (25)$$

In (25), the inputs are the error in position e_i and the error in velocity \dot{e}_i and the output is the control input of joint i ; *i.e.* it is a PD-type FLC. The following remarks are in order:

- The FLC in (25) is a special case of fuzzy systems, where Gaussian membership functions are used to introduce the input variables (e_i and \dot{e}_i) to the fuzzy network. Also, the fuzzification and defuzzification methods used in this study are not unique; see [28] for other alternatives. For example, using different membership functions (e.g. triangular, trapezoidal ... etc.) will result in a different FLC. However, the FLC in (25) is a simple one and the closed form relation between the inputs and the output makes it computationally inexpensive.
- Only three parameters per each DOF need to be tuned, namely, they are a_{ui} , a_{epi} and a_{evi} . This greatly simplifies the tuning procedure, since the search space is quite small relative to other works. For instance, the FLC in [21] needs 45 parameters to be tuned for a one DOF system.
- This controller is inherently bounded since $|\tanh(x)| \leq 1$.
- Each joint has independent control input u_i , $i = 1, 2, \dots, n$.
- In the case of robotic control, this controller can be regarded as output feedback controller since the joint's position and velocity are usually the outputs. If the joint velocity is not measured, it can be easily obtained using a differentiator as shown in **Figure 2**.

4. Experimental Setup: Test Rig, Reference Trajectory and Performance Measures

In this study, we have considered a two link planar robot whose diagrammatic sketch is shown in **Figure 3**. In this Figure, 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 parameter values of the links are given in **Table 2**. These inertia parameters have been calculated by simply measuring and weighting the mechanical elements of the arms.

4.1. The Test Rig

The test rig consists of a geared-drive horizontal robot arm with 2 DOF whose rigid links are joined with revolute joints; **Figure 4**. The robot has been built at the

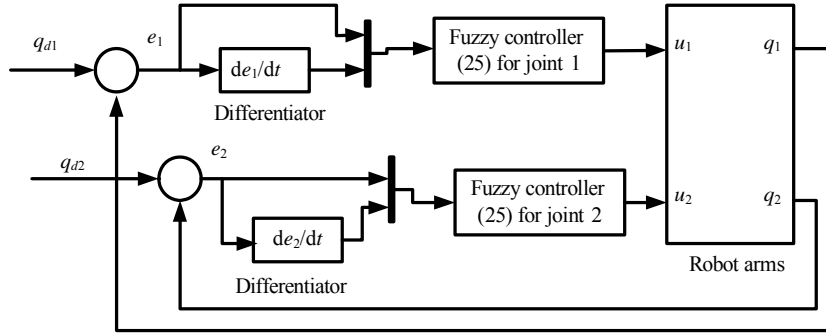


Figure 2. Configuration of the robotic fuzzy control structure (the case of two-link robot).

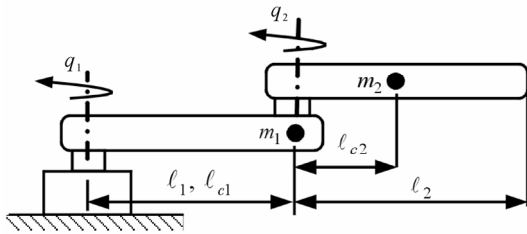


Figure 3. Schematic diagram of the two-link robot.

Table 2. 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
$I = m\ell^2/12$ inertia (kg·m ²)	0.00093	0.00033

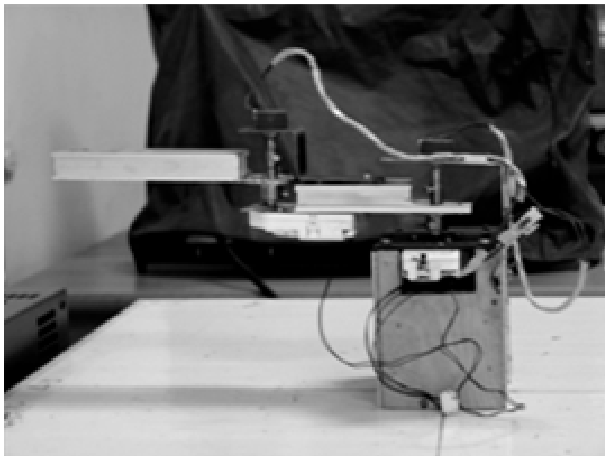


Figure 4. Experimental two-link planar arm.

Mechatronics lab, Faculty of Engineering, Assiut University. It is equipped with joint position sensors, motor drivers, AD/DA 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, 2 rpm, 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 AD/DA 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 5 shows the closed loop control system.

4.2. The Reference Trajectory

During the preliminary evaluation of the proposed FLC, we have examined three trajectories. They are sinusoidal trajectory, linear trajectory with parabolic blends and the cubic polynomial trajectory. In this paper, we present results of the latest trajectory. Results of other two trajectories; sinusoidal trajectory and linear trajectory with parabolic blends, can be found in the master thesis of the second author, [29]. A cubic polynomial trajectory in the joint space is defined by:

$$q_d(t) = a_0 + a_1t + a_2t^2 + a_3t^3 \tag{26}$$

where a_0, a_1, a_2 and a_3 are constants determined upon the trajectory constraints. The desired motion of the two joints is identical and starts from zero to 45° in 10 seconds. The motion constraints (boundary conditions) are: $q_{di}(t=0) = 0$, $q_{di}(t=10) = 45^\circ$, $\dot{q}_{di}(t=0) = 0$ and $\dot{q}_{di}(t=10) = 0$, where $i=1,2$ is the joint number. The desired trajectory according to (26) will be:

$$q_d(t) = 1.35t^2 - 0.09t^3, 0 \leq t \leq 10, \tag{27}$$

where $q_d(t)$ is in degrees.

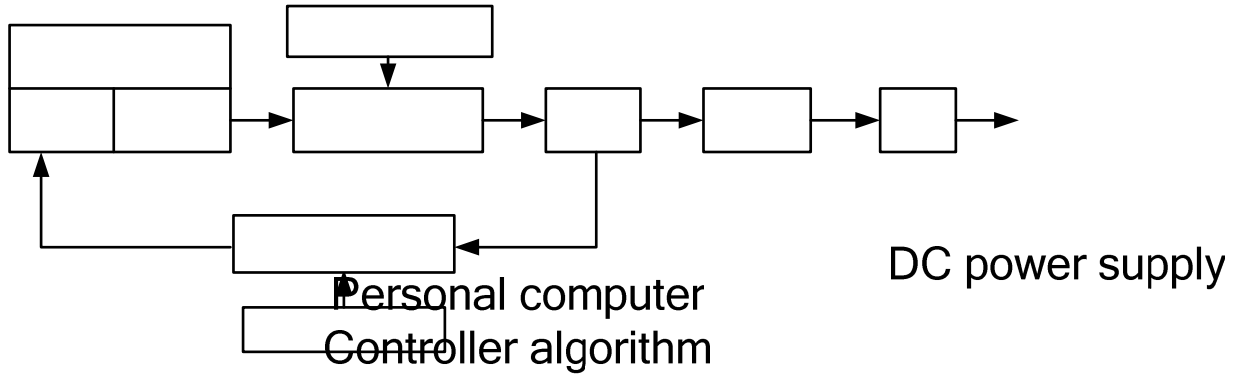


Figure 5. Block diagram of the test rig.

4.3. The Performance Measures

While comparing the efficiencies of the controllers that were experimentally tested on the robot arm, we will use some meaningful measures of the tracking error to quantitatively compare the performance results. One measure that will be used is the scalar valued Root Mean Square (RMS) error defined as

$$RMS = \left((1/T_f) \int_0^{T_f} \|e(t)\|^2 dt \right)^{1/2} \quad (28)$$

where $e(t)$ is the tracking error. Since data are only sent back at discrete time intervals, $t_1 \dots t_N$ with constant sampling period $\Delta T = t_{j-1} - t_j$ for all j ; we discretize (28) as

$$\begin{aligned} RMS &= \sqrt{\frac{1}{T_f} \sum_{j=1}^N [q_d(j) - q(j)]^2 \Delta T} \\ &= \sqrt{\frac{1}{N} \sum_{j=1}^N [q_d(j) - q(j)]^2} \end{aligned} \quad (29)$$

where $q(j)$ denotes $q(t_j) = q(j\Delta T)$ and $T_f = N \times \Delta T$. To get more insight on the tracking performance, we also use the maximum absolute value of the tracking error after two second from the starting time. We name it as e_{\max} which is defined as

$$e_{\max} = \max_{1 \leq j \leq N} |q_d(j) - q(j)| \quad (30)$$

The above two measures have been also adopted in [15].

5. Results and Discussion

In this section, the experiments conducted using four controllers are presented. These controller are the conventional PD controller, the proposed FLC, the CTC and the SMC. 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 four controllers have been initiated with initial position error equal to 10° , i.e. $q(0) = [10^\circ - 10^\circ]^T$ and the robot is at rest, i.e. $\dot{q}(0) = [00]^T$. This condition yields an initial position error $e = [-0.175 \ 0.175]^T$ radian.

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

$$u(t) = K_p e(t) + K_D \dot{e}(t) \quad (31)$$

where K_p and K_D are 2×2 positive definite diagonal matrices called the proportional and the derivative gain matrices, respectively. A traditional problem associated with PD control is that we cannot choose the proportional 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 which are computed according to (25) 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. The criteria upon which these values have been chosen is simply the fastest possible convergence of the initial errors.

For the CTC, the control gains according to (6) were selected after trials as $K_{p1} = 8 \times 10^6$, $K_{p2} = 7 \times 10^6$ and the derivative gains were chosen to be $K_{D1} = 3$, $K_{D2} = 2.5$ for the base and elbow links, respectively. These control gains have achieved the best possible tracking performance. The matrix $M(q)$ and the vector

$h(q, \dot{q})$ were computed on-line using the parameter values presented in **Table 2** and the equation of motion of two link planar robot which can be found in [4].

For the SMC, the estimated values for $\hat{M}(q)$, $\hat{h}(q, \dot{q})$ and the hitting control gain K in (14) were set as:

$$\hat{M} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.3 \end{bmatrix}, \hat{h} = \begin{bmatrix} 0.01 \\ 0.003 \end{bmatrix} \text{ and } K = \begin{bmatrix} 40 & 0 \\ 0 & 30 \end{bmatrix}.$$

The coefficients of the function $\beta(t)$ in (16) were selected as $c_1 = \text{diag}(0.3, 0.2)$ and $c_2 = \text{diag}(50, 40)$. Again, these control parameter values have been picked after a trial and error procedure so as to keep stability of the robot motion and achieve the best possible tracking performance, *i.e.* the fastest possible rate of convergence of the errors. In the coming experiments, the sign function in (10) has been replaced by saturation function to avoid chattering.

The performance criteria *RMS* as defined in (29) and the maximum error (30) for all experiments are visualized 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.

The tracking performance of the four controllers is

demonstrated in **Figures 6-10**. They show that, after suitable selection of the tuning parameters, the tracking errors of the four controllers have converged to a close zone around zero in the steady state phase. **Figure 6** shows that the transient period of the PD controller is slightly higher than that of the FLC, **Figure 7**. The transient phase of the CTC was the longest one as it can be noticed from **Figure 8**. **Figure 10** shows that the SMC was successful in bending the system trajectories toward sliding surfaces and consequently the errors have converged as depicted in **Figure 9**.

The performance measures are given in **Figure 11** and **Figure 12**. Referring to these two Figures the following remarks are in order.

- When applying initial position and velocity errors, it can be noticed that the FLC performs better than the PD controller in terms of the two performance criteria; *i.e.* RMS of errors and the maximum error.
- The CTC requires the accurate knowledge of the system dynamic model and the complete equations of motion are computed in real time. These conditions are difficult to verify in practice. As a result the CTC has the worst tracking performance.
- There are too many parameters which are needed to be tuned (experimentally selected) in the case of the

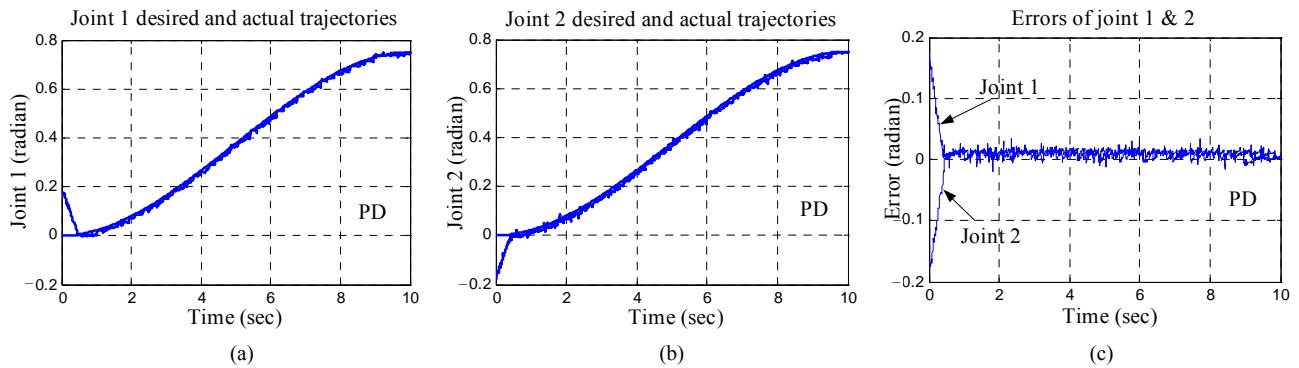


Figure 6. The desired and actual trajectories of (a) joint one, (b) joint two (b) and (c) the tracking errors of the PD controller.

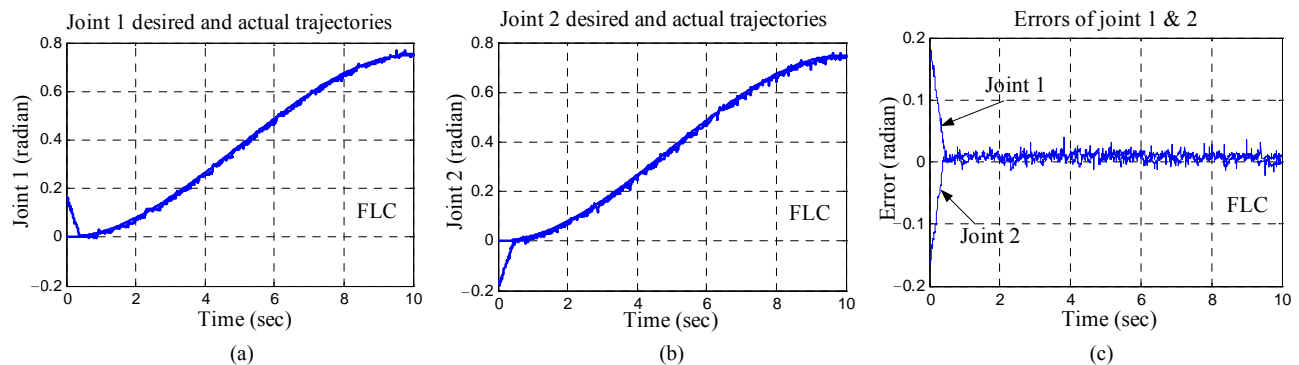


Figure 7. The desired and actual trajectories of (a) joint one, (b) joint two (b) and (c) the tracking errors of the proposed FLC.

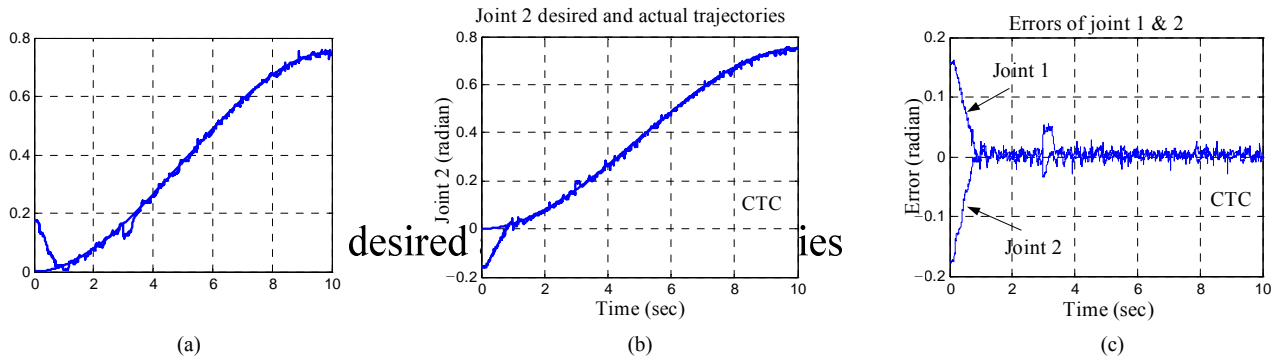


Figure 8. The desired and actual trajectories of (a) joint one, (b) joint two (b) and (c) the tracking errors of the CTC.

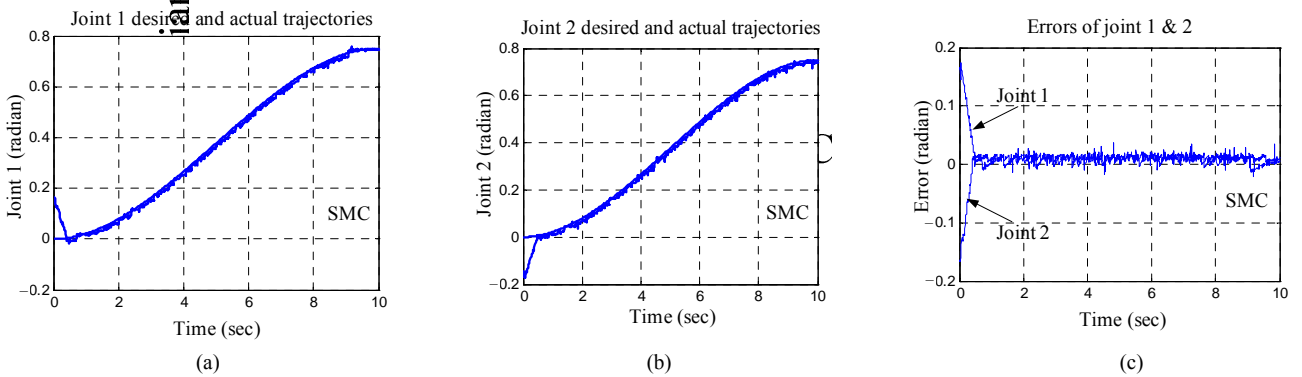


Figure 9. The desired and actual trajectories of (a) joint one, (b) joint two (b) and (c) the tracking errors of the SMC.

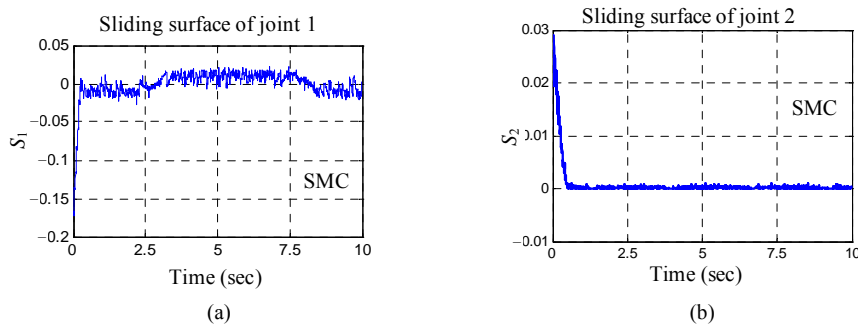


Figure 10. The sliding surfaces (SMC).

SMC in order to achieve the fastest possible convergence of the initial errors. This fact has affected the tracking performance of the controller.

- Finally, it can be concluded that the proposed FLC has achieved the ease of implementation and the best tracking performance.

6. 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. This procedure greatly simplifies the extraction of the fuzzy rules.

An important feature of this study is that it has transferred the proposed fuzzy PD controller to a closed-form relation between the inputs and the output, leading to a computationally efficient FLC. 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, CTC and SMC, the proposed FLC is inherently bounded; the upper and lower bounds can be arbitrary selected by suitably adjust its parameters.

The presented approach provides a systematic step by step procedure for the design of fuzzy-based decoupled

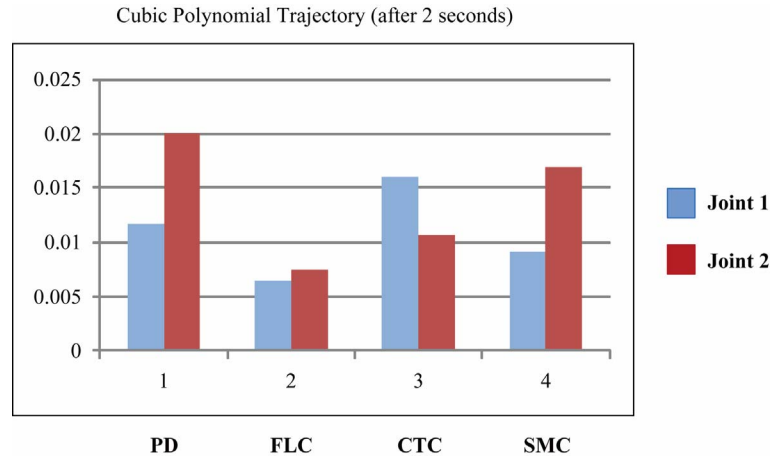


Figure 11. The RMS error of joint 1 and joint 2 for the four controllers in radian.

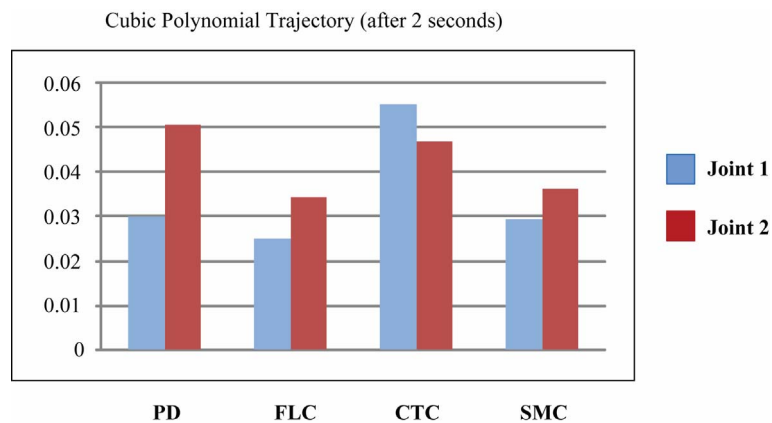


Figure 12. The maximum error of the four controllers in radian.

feedback controllers for a wide class of MIMO second order nonlinear systems. This control scheme 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 control. Robustness of the FLC has been examined via initial position errors. Relative to the conventional PD controller, CTC and SMC, the proposed FLC exhibits the best performance.

7. References

- [1] Z. Bingul and O. Karahan, "A Fuzzy Logic Controller Tuned with PSO for 2 DOF Robot Trajectory Control," *Expert Systems with Applications*, Vol. 38, No. 1, 2011, pp. 1017-1031. [doi:10.1016/j.eswa.2010.07.131](https://doi.org/10.1016/j.eswa.2010.07.131)
- [2] T.-H. S. Li and Y.-C. Huang, "MIMO Adaptive Fuzzy Terminal Sliding-Mode Controller for Robotic Manipulators," *Information Sciences*, Vol. 180, No. 23, 2010, pp. 4641-4660. [doi:10.1016/j.ins.2010.08.009](https://doi.org/10.1016/j.ins.2010.08.009)
- [3] L. Tian and C. Collins, "Optimal Placement of a Two-Link Planar Manipulator Using a Genetic Algorithm," *Robotica*, Cambridge University Press, Cambridge, Vol. 23, No. 2, 2005, pp. 169-176.
- [4] M. W. Spong, S. Hutchinson and M. Vidyasagar, "Robot Modeling and Control," John Wiley & Sons Inc., New York, 2006.
- [5] F. Reyes and R. Kelly, "Experimental Evaluation of Model-Based Controllers on a Direct-Drive Robot Arm," *Mechatronics*, Vol. 11, No. 3, 2001, pp. 267-282. [doi:10.1016/S0957-4158\(00\)00008-8](https://doi.org/10.1016/S0957-4158(00)00008-8)
- [6] J. Slotine and W. Li, "Applied Nonlinear Control," Prentice-Hall Inc., New Jersey, 1991.
- [7] H. B. Guo, Y. G. Liu, G. R. Liu and H. R. Li, "Cascade Control of a Hydraulically Driven 6-DOF Parallel Robot Manipulator Based on a Sliding Mode," *Control Engineering Practice*, Vol. 16, No. 9, 2008, pp. 1055-1068. [doi:10.1016/j.conengprac.2007.11.005](https://doi.org/10.1016/j.conengprac.2007.11.005)
- [8] W. Gueaieb, S. Al-Sharhan and M. Bolic, "Robust Computationally Efficient Control of Cooperative Closed-Chain Manipulators with Uncertain Dynamics," *Automatica*, Vol. 43, 2007, pp. 842-851. [doi:10.1016/j.automatica.2006.10.025](https://doi.org/10.1016/j.automatica.2006.10.025)

- [9] P. Herman, "Sliding Mode Control of Manipulators Using First-Order Equations of Motion with Diagonal Mass Matrix," *Journal of the Franklin Institute*, Vol. 342, No. 4, 2005, pp. 353-363. [doi:10.1016/j.jfranklin.2004.12.001](https://doi.org/10.1016/j.jfranklin.2004.12.001)
- [10] P. Herman, "Strict Lyapunov Function for Sliding Mode Control of Manipulator Using Quasi-Velocities," *Mechanics Research Communications*, Vol. 36, No. 2, 2009, pp. 169-174. [doi:10.1016/j.mechrescom.2008.09.010](https://doi.org/10.1016/j.mechrescom.2008.09.010)
- [11] D. Brambilla, L. M. Capisani, A. Ferrara and P. Pisu, "Fault Detection for Robot Manipulators via Second-Order Sliding Modes," *IEEE Transactions on Industrial Electronics*, Vol. 55, No. 11, 2008, pp. 3954-3963. [doi:10.1109/TIE.2008.2005932](https://doi.org/10.1109/TIE.2008.2005932)
- [12] M. Jin, J. Lee, P. H. Chang and C. Choi, "Practical Non-Singular Terminal Sliding-Mode Control of Robot Manipulators for High-Accuracy Tracking Control," *IEEE Transactions on Industrial Electronics*, Vol. 56, No. 9, 2009, pp. 3593-3601. [doi:10.1109/TIE.2009.2024097](https://doi.org/10.1109/TIE.2009.2024097)
- [13] 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. [doi:10.1177/027836499801700304](https://doi.org/10.1177/027836499801700304)
- [14] F. Alonge, F. D'Ippolito and F. M. Raimondo, "Globally Convergent Adaptive and Robust Control of Robotic Manipulators for Trajectory Tracking," *Control Engineering Practice*, Vol. 12, No. 9, 2004, pp. 1091-1100. [doi:10.1016/j.conengprac.2003.11.007](https://doi.org/10.1016/j.conengprac.2003.11.007)
- [15] 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, No. 6, 2005, pp. 721-729. [doi:10.1017/S026357470500158X](https://doi.org/10.1017/S026357470500158X)
- [16] M. Margaloit and G. Langholz, "Fuzzy Control of a Benchmark Problem: Computing with Words Approach," *IEEE Transactions on Fuzzy Systems*, Vol. 12, No. 2, 2004, pp. 230-235. [doi:10.1109/TFUZZ.2004.825083](https://doi.org/10.1109/TFUZZ.2004.825083)
- [17] V. Giordano, D. Naso, and B. Turchiano, "Combining Genetic Algorithms and Lyapunov-Based Adaptation for Online Design of Fuzzy Controllers," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 36, No. 5, 2006, pp. 1118-1127. [doi:10.1109/TSMCB.2006.873187](https://doi.org/10.1109/TSMCB.2006.873187)
- [18] 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.
- [19] A. B. Sharkawy, "Genetic Fuzzy Self-Tuning PID Controllers for Antilock Braking Systems," *Engineering Applications of Artificial Intelligence*, Vol. 23, No. 7, 2010, pp. 1041-1052. [doi:10.1016/j.engappai.2010.06.011](https://doi.org/10.1016/j.engappai.2010.06.011)
- [20] M. A. Llama, R. Kelly and V. Santibanez, "A Stable Motion Control System for Manipulators via Fuzzy Self-Tuning," *Fuzzy Sets and Systems*, Vol. 124, No. 2, 2001, pp. 133-154. [doi:10.1016/S0165-0114\(00\)00061-0](https://doi.org/10.1016/S0165-0114(00)00061-0)
- [21] T. L. Seng, M. Khalid and R. Yusof, "Tuning of a Neuro-Fuzzy Controller by Genetic Algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 29, No. 2, 1999, pp. 226-239. [doi:10.1109/3477.752795](https://doi.org/10.1109/3477.752795)
- [22] A. B. Sharkawy, "A Computationally Efficient Fuzzy Logic Controller for Robotic Systems," *Proceedings of the 9th International Conference on Production Engineering, Design and Control (PEDAC'9)*, Alexandria, 10-12 February 2009.
- [23] A. Visioli and G. Legnani, "On the Trajectory Tracking Control of Industrial SCARA Robot Manipulators," *IEEE Transactions on Industrial Electronics*, Vol. 49, No. 1, February 2002, pp. 224-232. [doi:10.1109/41.982266](https://doi.org/10.1109/41.982266)
- [24] F.-J. Lin and R.-J. Wai, "A Hybrid Computed Torque Controller Using Fuzzy Neural Network for Motor-Quick-Return Servo Mechanism," *IEEE/ASME Transactions on Mechatronics*, Vol. 6, No. 1, March 2001, pp. 75-89. [doi:10.1109/3516.914394](https://doi.org/10.1109/3516.914394)
- [25] L. A. Zadeh, "Fuzzy Logic = Computing with Words," *IEEE Transactions on Fuzzy Systems*, Vol. 4, No. 2, 1996, pp. 103-111. [doi:10.1109/91.493904](https://doi.org/10.1109/91.493904)
- [26] L. X. Wang, "Fuzzy Systems Are Universal Approximations," *Proceedings of IEEE International Conference on Fuzzy Systems*, San Diego, 8 March 1992. [doi:10.1109/FUZZY.1992.258721](https://doi.org/10.1109/FUZZY.1992.258721)
- [27] M. Margaloit and G. Langholz, "Fuzzy Lyapunov-Based Approach to the Design of Fuzzy Controllers," *Fuzzy Sets and Systems*, Vol. 106, No. 1, 1999, pp. 49-59. [doi:10.1016/S0165-0114\(98\)00356-X](https://doi.org/10.1016/S0165-0114(98)00356-X)
- [28] L. X. Wang, "A Course in Fuzzy Systems and Control," Prentice-Hall, Upper Saddle River, 1997.
- [29] M. M. Othman, "Fuzzy Control of Two Degrees of Freedom Robotic Arm," M.Sc Dissertation, Mechanical Engineering Department, Assiut University, Assiut, 2010.