An Efficient Adaptive Fuzzy Control Scheme for Industrial Manipulators

Abdel Badie Sharkawy, Douglas A. Plaza, and Daniel E. Ochoa
Escuela Superior Politécnica del Litoral, ESPOL, Guayaquil, Ecuador.
Email: {asharkaw, dochoa, douplaza}@fiec.espol.edu.ec

***Abstract*—This paper develops a generalized adaptive fuzzy control scheme for MIMO nonlinear second order systems. Here, the example robotic manipulators is used to illustrate the control algorithm. The controller for each degree of freedom (DOF) consists of a feedback fuzzy PD systems used to keep the closed-loop stable. The rule base consists of only four rules per each DOF. Furthermore, the fuzzy feedback system is decentralized and simplified leading to a computationally efficient control scheme. The proposed control scheme has the following advantages: 1) it needs no exact dynamics of the system and the computation is time-saving because of the simple structure of the fuzzy systems; and 2) the controller is robust against various uncertainties. The computational complexity of the proposed control scheme has been analyzed and compared with previous works. Computer simulations show that this controller is effective in achieving the control goals.**

***Index Terms*—robot manipulators, fuzzy pd feedback control, closed-loop stability, computational complexity**

I. INTRODUCTION

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, [1]. Also, because of many design parameters, like learning rates and initialization of the parameters to be adapted, etc., 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, [2]

Fuzzy controllers have demonstrated excellent robustness in both simulations and real-life applications. However, it has been proved that standard fuzzy logic controllers are not suitable for loop controllers, [3]. This fact is referred to that there are many tuning parameters in membership functions and control rules. Furthermore, standard fuzzy logic controller has a long computation time since it performs fuzzification, inference, and defuzzification processes in determining control inputs. Thus, it is difficult for control inputs of standard fuzzy logic control to be computed within the sampling time of a loop controller. For this reason, complexity reduction of fuzzy feedback controllers was the topic of many researchers; for instance see [4].

In this paper, we focus on the design of an adaptive fuzzy feedback controller based on the Lyapunov synthesis. Only four rules constitute the rule base for each DOF. Furthermore, the fuzzy feedback controller is decentralized and simplified leading to a computationally efficient adaptive fuzzy control scheme. To demonstrate the proposed approach, we use the example of robotics because it is a well-known example of nonlinear MIMO second order systems.

The paper is outlined as follows: in Section 2, the robot model and the nominal value of its parameter are introduced. This model is used to generate simulation data instead of experimental data from real robot platform. In Section 3, the fuzzy feedback controller is derived based on the Lyapunov direct method. Furthermore, the controller is simplified, i.e. it has a closed form mathematical relation with only three parameters need to be tuned and the controller gain is adaptively determined on-line so as to minimize a performance index. Section 4 discusses the computational complexity of the proposed control scheme in comparison with previous works. Simulation results are demonstrated in Section 5. Finally, some concluding remarks are given in Section 6.

II. ROBOT MODELING AND THE CONTROL STATEMENT

Without the loss of generality, we take the two-link rigid robot shown in Fig. 1, as an example to demonstrate the proposed control scheme. The inverse dynamic model is expressed as:

$$u = M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) \qquad (1)$$

Manuscript received January 1, 2015; revised June 21, 2015.

doi: 10.18178/joace.4.3.213-219

where $\theta \in R^n$ is the joint angular position vector of the robot; $u \in R^n$ is the vector of applied joint torques (or forces); $M(\theta) \in R^{n\times n}$ is the inertia matrix, positive definite; $C(\theta,\dot{\theta})\dot{\theta} \in R^n$ is the effect of Coriolis and centrifugal torques; and $G(\theta) \in R^n$ is the gravitational torques. The physical properties of the above model (1) can be found in [5]; however, they are not needed here.

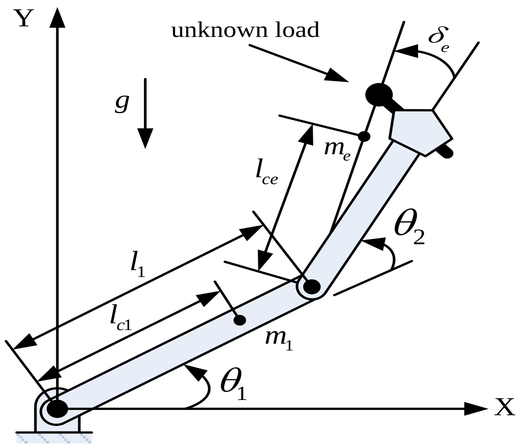

Figure 1. An articulated two-link manipulator.

For the robot shown in Fig. 1, (1) can be rewritten as:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} -h\dot{\theta}_2 & -h(\dot{\theta}_1+\dot{\theta}_2) \\ h\dot{\theta}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}$$

where

$$M_{11} = a_1 + 2a_3\cos(\theta_2) + 2a_4\sin(\theta_2),\ M_{22} = a_2,$$

$$M_{21} = M_{12} = a_2 + a_3\cos(\theta_2) + a_4\sin(\theta_2),$$

$$h = a_3\sin(\theta_2) - a_4\cos(\theta_2)$$

$$G_1 = b_1\cos(\theta_1) + b_2\cos(\theta_1+\theta_2),$$

$$G_2 = b_2\cos(\theta_1+\theta_2)$$

With

$$a_1 = I_1 + m_1 l_{c1}^2 + I_e + m_e l_{ce}^2 + m_e l_1^2,\ a_2 = I_e + m_e l_{ce}^2,$$

$$a_3 = m_e l_1 l_{ce}\cos(\delta_e),\ a_4 = m_e l_1 l_{ce}\sin(\delta_e)$$

$$b_1 = m_1 g l_{c1} + m_e g l_1,\ b_2 = m_e g l_{ce}.$$

The nominal parameters of the two-link manipulator are chosen as follows:

$$m_1 = 5\,kg,\ m_e = 2.5\,kg,\ l_1 = 1.0\,m,\ l_{c1} = 0.5\,m$$

$$l_{ce} = 0.5\,m,\ \delta_e = 30^0,\ I_1 = 0.36\,kgm^2,\ I_e = 0.24\,kgm^2$$

Position control, or also the so-called regulation problem is one of the most relevant issues in the operation of robot manipulators. This is a particular case of the motion control or trajectory control. The primary goal of motion control in joint space is to make the robot joints track a given time-varying desired joint position, $\theta^d = [\theta_1^d, \theta_2^d]^T$.

Several control architectures related to robot control can be found in literature ranging from the simple PD, learning based, adaptive, and adaptive/learning hybrid controllers. The reader is referred to [6], [7] and the references included. The main advantage of the PD controller is that it can easily be implemented on simple microcontroller architectures. On the other hand, the performance obtained from PD controllers is not satisfying for most of the sensitive applications [7].

III. DECENTERALIZED ADAPTIVE FUZZY CONTROL

The performance of any fuzzy logic controller 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 fuzzy controller. However, a large set of rules requires more on-line computational time and more parameters need to be adjusted. Adjustment of the fuzzy system may be achieved using GAs [8]. However, GAs cannot be used on-line and perfect mathematical model or experimental data should be available. In this Section, a robust adaptive PD-type fuzzy feedback controller is driven for a class of MIMO second order nonlinear systems.

A. Construction of Fuzzy Feedback Controllers

In this Sub-section we apply the fuzzy synthesis 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:

$$\ddot{x}(t) = f(x,\dot{x},u) \qquad (2)$$

where $f(x,\dot{x},u)$ is an unknown continuous function, u is the feedback control input and $x(t) = [x_1, x_2, \cdots, x_n]^T$ is the state vector and $\dot{x} = \frac{dx}{dt} = [\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_n]^T$. We now seek a smooth Lyapunov function $V: R^n \to R^n$ for the continuous feedback model (1) that is positive definite, i.e. $V(x) > 0$ when $x \neq 0$ and $V(x) = 0$ when $x = 0$, and grows to infinity: $V(x) \to \infty$ as $x^T x \to \infty$. Obviously, this holds for a generalized Lyapunov candidate function of the following quadratic form:

$$V(x,t) = \tfrac{1}{2}x^T x + \tfrac{1}{2}\dot{x}^T\dot{x} \qquad (3)$$

Differentiating (3) with respect to time gives

$$\dot{V}(x,t) = x_1\dot{x}_1 + x_2\dot{x}_2 + \cdots + x_n\dot{x}_n + \dot{x}_1\ddot{x}_1 + \dot{x}_2\ddot{x}_2 + \cdots + \dot{x}_n\ddot{x}_n$$

from which

$$\dot{V}(x,t) = (x_1\dot{x}_1 + \dot{x}_1\ddot{x}_1) + (x_2\dot{x}_2 + \dot{x}_2\ddot{x}_2) + \cdots + (x_n\dot{x}_n + \dot{x}_n\ddot{x}_n)$$

This is equal to

$$\dot{V}(x,t) = \dot{V}_1 + \dot{V}_2 + \cdots + \dot{V}_n \qquad (4)$$

where

$$\dot{V}_i(x,t) = x_i\dot{x}_i + \dot{x}_i\ddot{x}_i, \qquad i = 1,2,\cdots n$$

Then the standard results in Lyapunov stability theory imply that the dynamic system (2) has a stable equilibrium $x = x_e$ if each $\dot{V}_i$ in (4) is ≤ 0 along the system trajectories. To achieve this, we have chosen the control $u_i(x)$ to be proportional to $\ddot{x}_i$.

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

$$\dot{V}_i(x,t) = x_i\dot{x}_i + \alpha_i\dot{x}_i u_i(x) \leq 0, \qquad i = 1,2,\cdots n \quad (5)$$

where α_i is a positive constant. The results of Wang [9] state that, a fuzzy system that would approximate (5) 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 feedback control u_i. A possible form of the control rules is:

IF x_i is (*lv*) and/or $\dot{x}_i$ is (*lv*)

THEN u_i is (*lv*), $\qquad i = 1,2,\cdots n$

where the (*lv*) are linguistic values (e.g. *positive*, *negative*). These rules constitute the rule base for a Mamdani-type fuzzy controller.

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 $\ddot{x}$. This assumption can be justified for a large class of second order nonlinear mechanical systems, [10]. 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 control rules. Clearly, the exact mathematical model is not needed.

B. Adaptive Fuzzy Feedback Control Design

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.

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 $e(t)$ and $\dot{e}(t)$ as:

$$e(t) = \theta(t) - \theta^d(t), \text{ and } \dot{e}(t) = \dot{\theta}(t) - \dot{\theta}^d(t) \qquad (6)$$

where θ^d and $\dot{\theta}^d$ are vectors of the desired joint position and velocity, respectively. We now apply the approach presented in the previous Sub-section in order to find a fuzzy controller that achieves tracking to the robotic system under consideration. To this end, let us choose the following Lyapunov function candidate

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

Differentiating with respect to time and using (4) 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

$$\dot{V}_i = e_i\dot{e}_i + \dot{e}_i\ddot{e}_i \leq 0 \qquad (8)$$

In some neighborhood of the equilibrium of (7). Taking the control u to be proportional to $\ddot{e}$, Eqn (8) can be rewritten as:

$$\dot{V}_i = e_i\dot{e}_i + \alpha_i\,\dot{e}_i u_{FB_i} \leq 0 \qquad (9)$$

where α_i is positive constant, $i = 1,\cdots,n$. Sufficient conditions for (9) to hold can be stated as follows.

- if, for each $i \in [1,\cdots,n]$, e_i and $\dot{e}_i$ have opposite signs and $\alpha_i u_i$ is zero, inequality (9) holds;
- if e_i and $\dot{e}_i$ are both positive, then (9) will hold if $\alpha_i u_i$ is negative; and
- if e_i and $\dot{e}_i$ are both negative, then (9) will hold if $\alpha_i u_i$ is positive. $i \in [1,\cdots,n]$ denotes the joint number.

Using these observations and assuming that α_i is positive small number, one can easily obtain the four rules listed below in Table I.

TABLE I. FUZZY RULES FOR THE FUZZY FEEDBACK CONTROLLER

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

$i = 1,\cdots n$

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_i, $\dot{e}_i$ and u_i which follow directly from the stabilizing conditions of the Lyapunov function, (7).

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. Relative to other works, this number of rules is quite small. For example, in [11], the rule base of a two-link robot consists of 625 rules. After introducing a rule base reduction approach, the authors in [11] reach to a rule base consists of 160 rules, which is hard to be implemented.

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:

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

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

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

where $a_z > 0$ and z stands for control variable, the product for "and" and center of gravity inferencing. For some positive constant k_i, $i \in [1, \cdots, n]$ denotes the joint number, the above four rules can be represented by the following mathematical expression:

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

This yields the fuzzy feedback controller

$$u_i = -k_i[\tanh(2a_{1i}e_i) + \tanh(2a_{2i}\dot{e}_i)], \quad i = 1, \cdots, n \quad (10)$$

In (10), 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 fuzzy feedback controller. The following remarks are in order:

- The fuzzy controller in (10) 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 [10] for other alternatives. For example, using different membership functions (e.g. triangular, trapezoidal etc.) will result in a different fuzzy controller. However, the controller in (10) 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 k_i , a_{1i} and a_{2i} . This greatly simplifies the tuning procedure; since the search space is quite small relative to other works. For instance, the fuzzy controller in [12] needs 45 parameters to be tuned for a one DOF system.
- This controller is inherently bounded since $|\tanh(x)| \le 1$.

Finally, the fuzzy PD gain, i.e. k_i , $i \in [1, \cdots, n]$ is chosen so as to minimize the following quadratic performance index:

$$J_i = \frac{1}{2}\left\{r_i[u_i(k)]^2\right\} \quad (11)$$

where input r_i is a constant. According to the gradient method, the learning algorithm of the parameter k_i in the feedback fuzzy controller (10) can be derived as follows:

$$\begin{aligned}\Delta k_i &= -\frac{\partial J_i}{\partial k_i} = -\frac{\partial J_i \partial u_i}{\partial u_i \partial k_i} \\ &= -r_i u_i[\tanh(2c_{1i}e_i) + \tanh(2c_{2i}\dot{e}_i)]\end{aligned} \quad (12)$$

Thus, the fuzzy feedback controller uses the e_i , $\dot{e}_i$ and u_i to compute (12) and update the control gain k_i given that $k_i(0) \neq 0$. The overall closed-loop control system is shown in Fig. 2.

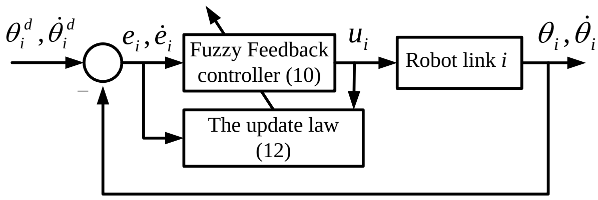

Figure 2. Configuration of the proposed decentralized fuzzy control scheme of joint i.

IV. Computational Aspects

In general, control algorithms for closed loop control should require short computation time due to limited memory of low-cost microprocessors. This Section discusses the computational complexity of the feedback controller and compares it with that of a self-tuning fuzzy controller proposed in [13]. It is shown that the proposed control scheme is computationally very efficient.

Naturally, the computational burden can be evaluated in terms of required mathematical multiplication and addition operations. The computation of the fuzzy feedback controller can be divided into two parts: computation of (10) and computation of the adaptive gain; k_i , (12). For the sake of comparison, Table II demonstrates the computational complexity of our scheme with the self-tuning fuzzy controller proposed in [13]. The comparison is fair since the feedback controller in [13] is essentially a PD fuzzy controller with self-tuning mechanism. In [13], the rule base has been transformed to a decision table and is used by a back-propagation algorithm to adjust the scaling factors of the fuzzy system. The difference resides in the fact that the rule base in [13] consists of 49 rules for one DOF system and the mapped elements (e and $\dot{e}$) are obtained by interpolation. Furthermore, the tuning procedure is composed of two stages and some learning steps are needed by the second stage, while the tuning system using (12) is much simpler. Simulation results, in the coming Section, show that it is also efficient.

TABLE II. COMPUTATIONAL COMPLEXITY OF THE FUZZY FEEDBACK CONTROLLER

	Self-tuning fuzzy controller [13]	The proposed fuzzy controller
Addition	97*n*	6*n*
Multiplication	113*n*	15*n*

V. SIMULATION RESULTS

The purpose of the simulation is to investigate the robustness of the proposed control scheme. The robot system considered in the simulation is the two-link robot presented in Section 2. Through the simulations, the physical insight of the behavior is revealed. In the coming results, it is assumed that $\theta_1^d = 0.5\pi(1-e^t)$, $\theta_2^d = \pi(1-e^t)$ and initial positions of joints $\theta_1(0) = \theta_2(0) = \pi/15\,rad$, which are equivalent to the initial position errors, since the desired positions are $\theta_1^d(0) = \theta_2^d(0) = 0$. Also, the robot is initialized at rest, i.e. the initial velocities of joints $\dot{\theta}_1(0) = \dot{\theta}_2(0) = 0^o\ rad/\sec$. This initialization imposes a large initial velocity error since $\dot{e}_1(0) = -\pi/2,\quad \dot{e}_2(0) = -\pi\ rad/\sec$. The input torque has been saturated to $|u_1|,|u_2| \le 300\ N.m$. With these initialization conditions, one can expect uneasy transient stage.

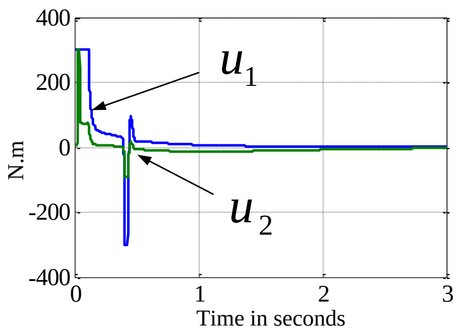

Figure 3. The control effort.

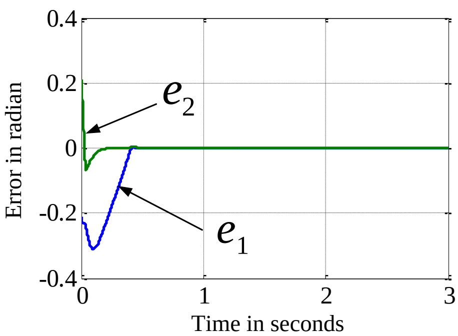

Figure 4. The tracking errors

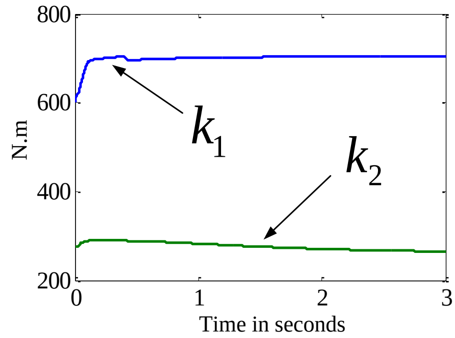

Figure 5. Record of the adaptive control gains during motion.

The input torques are shown in Fig. 3 and Fig. 4 shows the evolution of the tracking errors. They show that the errors have converged to zero. Note that the transient period is less than 0.5 seconds. Otherwise, it is interesting to notice how the control gains evolve with time. Fig. 5 depicts the evolution of these parameters with time. They have been initialized as $k_1(0) = 600\ N.m$ and $k_2(0) = 275\ N.m$.

In order to observe how the controller behaves in the presence of various uncertainties, two types of uncertainties are considered, namely, unmodeled nonlinear friction and unknown payloads.

A. Unmodeled Friction

At the off-line training stage of our simulation, we obtain the training samples from the robot model in (1), which does not consider the nonlinear friction. In order to examine the performance of the controller in the presence of unmodeled nonlinear friction, the following unmodeled nonlinear friction is added at the control stage:

$$F = F_d + F_s$$

where F_d and F_s are the dynamic and static friction torques, respectively. They can be expressed by:

$$F_d = \begin{bmatrix} d_1\cos(x_1) & 0 \\ 0 & d_2\cos(x_2) \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} \text{ and } F_s = \begin{bmatrix} c_1\,\mathrm{sgn}(\dot{x}_1) \\ c_2\,\mathrm{sgn}(\dot{x}_2) \end{bmatrix}$$

We use $d_1 = 50, d_2 = 30$ and $c_1 = 18, c_2 = 12$. Results are shown in Fig. 6 and Fig. 7. It can be noticed that the transient period has increased relative to the cases when the friction was not considered. Also the input torques is relatively higher during this period. Nevertheless, convergence of the tracking errors has been achieved.

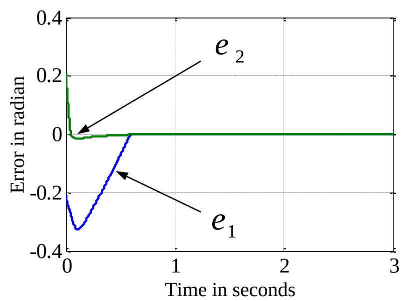

Figure 6. The tracking errors in the presence of unmodeled friction.

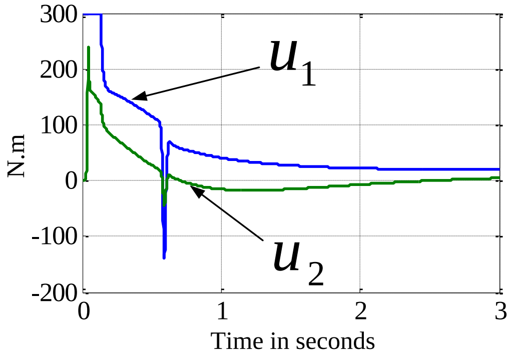

Figure 7. The control input in the presence of unmodeled friction.

B. *Unknown Payload*

In robot systems, the unknown payload is one of the major dynamic uncertainties. Compared with the parameter uncertainties and unmodeled friction, the influence of unknown payload is much greater. The coming results are obtained when the mass and inertia of the base and elbow links (carrying the payload) have been increased to 150%. This increase in the mass and inertia of the two links is supposed to be unknown. Fig. 8 shows that input torque is relatively high. Also, the tracking errors exhibit larger overshoot during the transient period, Fig. 9. However, convergence of errors to a narrow region close to zero has taken place.

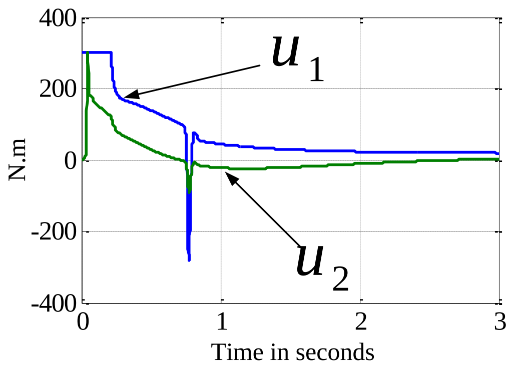

Figure 8. The input torques when the payload increases to 150%.

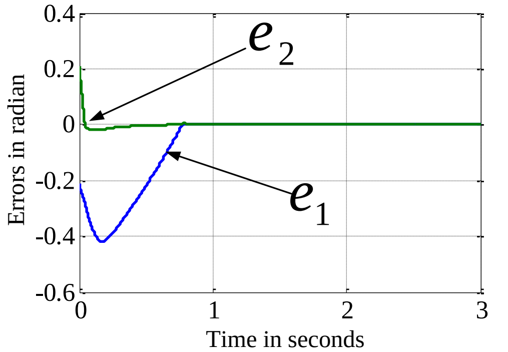

Figure 9. The tracking errors in the presence of 150% increase in the payload.

VI. CONCLUSIONS

In this paper, a decentralized adaptive fuzzy control scheme for second order systems has developed. Simulation results for robot manipulator show that the proposed control scheme works well, even if the ideal model is not in concordance with the real inverse dynamics. An important feature of this study is that it has transferred the proposed fuzzy feedback controller to a closed-form relation between the inputs and the output, leading to a computationally efficient adaptive fuzzy logic controller. The rule base consists of only four rules and has a PD-like structure. The gains are tuned on-line based on the gradient method. This feedback controller is inherently bounded; the upper and lower bounds can be arbitrary selected by suitably adjust its parameters. Finally, it can be concluded that using the proposed control approach presents a convenient option for controlling a large class of nonlinear MIMO second order systems.

ACKNOWLEDGMENT

This work was supported by a grant from the Ecuadorian Government, SENESCYT.

REFERENCES

[1] S. Tong, S. Sui, and Y. Li, “Adaptive fuzzy decentralized tracking fault-tolerant control for stochastic nonlinear large-scale systems with unmodeled dynamics,” *Information Sciences*, vol. 289, pp. 225–240, 2014.

[2] S. Tong, Y. Li, Y. Li, and Y. Liu, “Observer-based adaptive fuzzy backstepping control for a class of stochastic nonlinear strict-feedback systems,” *IEEE Trans. on Systems, Man, and Cybernetics—Part b: Cybernetics*, vol. 41, no. 6, pp. 1693-1704, December, 2011.

[3] M. Margaloit and G. Langholz, “Fuzzy control of a benchmark problem: Computing with words approach,” *IEEE Trans. on Fuzzy Systems*, vol. 12, no. 2, pp. 230-235, 2004,.

[4] Y. H. Kim, S. C. Ahn, and W. H. Kwon, “Computational complexity of general fuzzy logic control and its simplification for a loop controller,” *Fuzzy Sets and Systems*, vol. 111, pp. 215-224, 2000.

[5] T. Hseng, S. Li, and Y. C. Huang, “MIMO adaptive fuzzy terminal sliding-mode controller for robotic manipulators,” *Information Sciences*, vol. 180, pp. 4641-4660, 2010.

[6] S. Liuzzo and P. Tomei “A global adaptive learning control for robotic manipulators,” *Automatica*, vol. 44, pp. 1379-1384, 2008.

[7] S. Yamacli and H. Canbolat, “Simulation of a SCARA robot with PD and learning controllers,” *Simulation Modelling Practice and Theory*, vol. 16, pp. 1477-1487, 2008.

[8] A. B. Sharkawy, “Genetic fuzzy self-tuning PID controllers for antilock braking systems,” *Engineering Applications of Artificial Intelligence*, vol. 23, no. 7, pp. 1041-1052, 2010.

[9] L. X. Wang, *A Course in Fuzzy Systems and Control*, Upper Saddle River, NJ: Prentice-Hall, ch. 4, 1997, pp. 76-81.

[10] L. A. Zadeh, “Fuzzy logic = computing with words,” *IEEE Trans. Fuzzy Systems*, vol. 4, no. 2, pp. 103-111, 1996.

[11] H. Bezine, N. Derbel, and A. M. Alimi, “Fuzzy control of robot manipulators: Some issues on design and rule base size reduction,” *Engineering Applications of Artificial Intelligence*, vol. 15, pp. 401-416, 2002.

[12] 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, pp. 226-239, April, 1999.

[13] C. T. Chao and C. C.Teng, “A PD-like self-tuning fuzzy controller without steady-state error,” *Fuzzy Sets and Systems*, vol. 87, pp. 141-154, 1997.

Abdel Badie Sharkawy was born in Assiut Egypt on December 4th, 1958 and got PhD in Robotic Control in 1999 from the Slovak Technical University in Bratislava, Slovakia. The major area of interest are modeling and control using intelligent systems. He was a senior lecturer within the department of mechatronics engineering, the Hashemite University, Jordan for three years (2001-2004) and within the electrical engineering department, Al-Tahady University, Sirte, Libya during the fall semester, 2005. He was a professor at mechanical engineering department, Assiut University, Egypt (April 2012- July 2014). Now, he is a visiting professor (Prometeo Program) in the Faculty of Electrical and Computer Engineering, ESPOL, Guayaquil, Ecuador. His research interests include adaptive fuzzy identification and control, automotive control systems, robotics (modeling and control), and the use of neural networks in the control of mechanical systems.

Douglas A. Plaza born in Guayaquil, Ecuador on 10th June 1977. He obtained his Ph.D. degree in Electromechanical Engineering from Ghent University, Ghent, Belgium in 2013 and a master degree in Industrial Control Engineering from Universidad de Ibague, Ibague, Colombia in 2008. Currently, he is a lecturer of control theory associated to the Faculty of Electrical and Computing Engineering at Escuela Superior Politecnica del Litoral ESPOL. His research interests include: Kalman filtering, sequential Monte Carlo methods, model predictive control and non-linear control.

Daniel E. Ochoa was born in Guayaquil-Ecuador on October 17th, 1975. He was awarded an Computer Engineering degree at Escuela Superior Politécnica del Litoral (ESPOL), Guayaquil-Ecuador in 2000 and a PhD in Computer Science in 2011 at Gent University in Gent-Belgium. His interest are computer vision and robotics. He worked as research assistant at IPI group in Gent University and currently he is the head of the Computer Vision and Robotic center at ESPOL where he also works as a professor since 2013. He has published several indexed publications and has been reviewer of national and international scientific journals. Most of his research work has been done in biological image analysis. He is a founding member and head of the robotic and intelligent systems network of Ecuador.