Robot Manipulators Tracking Using Hybrid Fuzzy Logic Controller

Eddie Kwan and Ming Liu
Department of Electrical and Computer Systems Engineering
Monash University, Clayton Campus
Clayton, VIC 3168
Australia
eddie.kwan@eng.monash.edu.au

Abstract

This paper proposes a hybrid fuzzy controller which combines a conventional PD and a self-tuning fuzzy logic controller (STFLC) for robot manipulators trajectory tracking control. With the PD control as the preliminary component in maintaining the local stability of the entire system, the added STFLC provides fast responses to system variations and uncertainty in the robot dynamics. The STFLC considered belongs to a rule-based type, which uses linguistic IF...THEN rules obtained by human expert and mean of maxima defuzzification method. The scaling factors of the input variables are self-tuned online to cover the entire universe of discourse of its membership functions based on some performance measurements. The proposed controller has been verified using simulation of a two-link robot arm to track sinusoidal reference input signals with load variation in the elbow.

1 Introduction

The trajectory tracking control of robot manipulators has been well studied in the last two decades. Numerous approaches based on the methodology of feedback linearization, adaptive control and robust control, have shown excellent tracking performance with global asymptotically convergence of tracking errors. However most of these approaches require an exact model of the system under control, which was difficult to obtain for most robotic systems due to their complexity, nonlinear couplings and structured and unstructured uncertainties such as load variation, friction, external disturbances etc.

Since the introduction of *fuzzy sets theory* [Zadeh, 1965], and successfully implementation to a steam engine control system [Mamdani and Assilian, 1975], many research publications have shown that fuzzy logic control (FLC) yields better performance and robustness compared with the traditional counterpart [Kim and Bien, 1995; Yi and Chung, 1997]. In particular, FLC appears to be very effective in dealing with complex systems to which the traditional control techniques fail to provide satisfactory results. FLC uses human expert opinion as the knowledge base to reason for an appropriate control action, it treats the system to be controlled as a "black box" that neither mathematical knowledge nor explicit model of the plant are required. Another important issue of using FLC is the fact that it can cope the nonlinearity of most systems controlled.

In this paper, we propose a hybrid fuzzy controller which consists of a decentralized PD controller and a self-tuning fuzzy logic controller (STFLC) for robot manipulators tracking control. The PD controller provides quantitative control to ensure local stability of the whole system, and the STFLC uses qualitative information to enhance the system performance.

This paper is organized as follows: Section 2 outlines the equations of motion for rigid robot, the tracking error dynamics and its properties. In section 3, we review the traditional linear PD control scheme and its limitations in robot tracking control. In section 4, we describe the basic construction of the hybrid fuzzy logic controller proposed. We present the self-tuning algorithm in section 5. Section 6 shows some simulation results for verification and finally in section 7 are conclusions.

2 Motion Equations

We consider the equations of motion of rigid robot arms with n DOFs, where $n = \{1, 2, \cdots, n\}$ is an integer set:

$$D(q)\ddot{q} + h(q, \dot{q}) + g(q) = \tau - f(\dot{q}). \qquad (1)$$

where $q = q(t) \in \Re^n$ is the robot arm joint angle vector; $\tau \in \Re^n$, the input torque vector; $D(q) = D^T(q) \in \Re^{n \times n}$, the positive definite inertia matrix.

An important property of system (1) is the passive mapping from input torque $(\tau - f)$ to joint velocity $\dot{q}$ [Arimoto and Miyazaki, 1984; Slotine and Li, 1987], which is characterized by that

$$S = \dot{D}/2 - H_m = -S^T \qquad (2)$$

i.e., S is a skew-symmetrical matrix.

Consider the trajectory tracking problem we denote the reference trajectories as $q_d \in C^2$, a class of twice differentiable functions, and define the position tracking error vector as $e = q_d - q$. Further, by introducing a generalized tracking error vector r [Slotine and Li, 1987]:

$$r = \dot{e} + \Phi e \tag{3}$$

where $\Phi = diag\{\phi_1, \cdots, \phi_n\} > 0$ is a constant matrix, (1) can be rewritten as

$$D(q)\dot{r} + H_m(q, \dot{q})r = \xi - \tau \tag{4}$$

$$\xi = D(q)(\ddot{q}_d + \Phi\dot{e}) + H_m(q, \dot{q})(\dot{q}_d + \Phi e) + g(q) + f(\dot{q}) \tag{5}$$

Thus, for a class of q_d and Φ given, equations (3), (4) and (5) define the tracking error dynamics of (1) in which input torque τ is to be designed.

The tracking error feedback system (3), (4) and (5) has the following properties:

Property 1: Using the passivity feature given by 2 it can be shown that the mapping $(\xi - \tau) \mapsto r$ defined by 4 is passive [Liu, 1997].

Property 2: In terms of the truncated L_∞ norms the cascade mapping $r \mapsto (e, \dot{e}) \mapsto \xi$ defined by (3) and (5) satisfies

$$\|\xi\|_{T\infty} \leq \beta_1 + \beta_2\|r\|_{T\infty} + \beta_3\|r\|^2_{T\infty} \tag{6}$$

where β_1, β_2 and β_3 are some positive constants.

Property 3: If we let

$$\tau = Kr + \tau_0 \tag{7}$$

where $K = diag\{k_1, k_2, \cdots, k_n\} > 0$ and $Kr = K\dot{e} + K\Phi e$ is a PD control law, then (4) becomes

$$D(q)\dot{r} + H_m(q, \dot{q})r + Kr = \xi - \tau_0. \tag{8}$$

which defines a finite gain mapping $\xi - \tau_0 \mapsto r$ satisfying

$$\|r\|^2_T \leq k^2_{min}\|\xi - \tau_0\|^2_T + \|e(0)\|_\infty \tag{9}$$

where $k_{min} = min_{k\in n}k_i$ is the minimum element of K, and $e(0)$ is the initial position error.

Based on Property 3, if we have $\tau_0 = \xi$ (full model based scheme), or, if we can design a control law such that mapping $r \mapsto (\xi - \tau_0)$ is finite gain (adaptive scheme [Slotine and Li, 1987]), then $r \to 0$ globally according to the small gain theorem [Hill, 1976].

3 Linear PD Control

The most popular decentralized control is the linear PD control that obtained by setting τ_0 in (7) to zero, its i-th component is given by

$$\tau_i = k_i r = k_i e_i + k_i\phi_i\dot{e}_i \tag{10}$$

Due to its computation simplicity and resultant independent-joint controller structure, the PD control has been broadly adopted by robot arms. For set-point control, it is well known that the PD law ensures the global asymptotically stability if the gravity force can be compensated. For tracking control, the local ultimate bound of the tracking errors has been studied in [Liu, 1997]. In our problem formulation of using (4) and (5), in terms of the infinite norms, the L_∞ stability can be described as follows: If $r_0 = r(0)$ and k_i satisfy

$$\|r_0\|_\infty < \delta_2 \tag{11}$$

$$k_{min} > n(\beta_2 + 2\sqrt{\beta_1\beta_3}) \tag{12}$$

then the tracking error r converges to a residual set

$$\Omega_{fp} = \{r : \|r\|_\infty \leq \delta_1\} \tag{13}$$

where

$$\delta_1 = \frac{1}{2n\beta_3}\{(k_{min} - n\beta_2) - d\} \tag{14}$$

$$\delta_2 = \frac{1}{2n\beta_3}\{(k_{min} - n\beta_2) + d\} \tag{15}$$

in which $d = \sqrt{(k_{min} - n\beta_2)^2 - 4n^2\beta_1\beta_3}$. The stability condition on k_{min} given by (12) can be regarded as a nonlinear version of the small gain theorem [Hill, 1976]. The local stability of PD control can be interpreted as follows: Due to the quadratic boundedness of the interconnection, linear PD control can only guarantees the ultimate stability for small $r(0)$. In the case where $\|r_0\| > \delta_2$, linear control becomes impotent due to the strength of the quadratic term in the interconnections.

Regarding the effect of the PD gain, it is easy to see that if $k_i \to \infty$ then $\delta_2 \to \infty$, $\delta_1 \to 0$ and $r \to 0$; the closed loop system becomes globally asymptotically stable. However, as large k_i will stimulate the un-modeled high frequency modes of robot arms caused by the joint and link compliance, it has to be avoided in applications.

4 Hybrid Fuzzy Controller

The proposed hybrid fuzzy controller uses both PD and fuzzy logic control schemes to produce the final control action. As mentioned in the previous section, linear PD control only guarantees local stability and the size of the tracking errors size depends on the values of k_{min} and robots' dynamic parameters. In order to improve the stability and tracking performance, we add an additional nonlinear feedback in terms of fuzzy terms to enhance the control actions especially when the robots suffer from load variation and disturbance. The new control law is obtained by adding a fuzzy torque τ_i' to (10):

$$\tau_i = k_i e_i + k_i\phi_i\dot{e}_i + \tau_i' \tag{16}$$

The PD part of the proposed hybrid controller uses concepts in the previous section to provide linear control. The fuzzy control part consists of three major components: the fuzzification process, the inference engine and the defuzzification process. The fuzzification process converts a measured crisp value to its corresponding fuzzy representation using membership functions. After that, the inference engine, which also known as fuzzy processing, manipulates the fuzzified input signals by using its knowledge base to reason for an appropriate output control action described in linguistic term. This linguistic term has to be converted back to crisp value by the defuzzification process before it can be used as the control input to the plant.

We used Mamdani fuzzy rules [Mamdani and Assilian, 1975] to synthesis the fuzzy controller, which has the following general form:

$$R_k: \text{IF } (x_1 \text{ is } X_1^k) \text{ AND } \ldots \text{ AND } (x_n \text{ is } X_n^k) \text{ THEN } y_1 \text{ is } Y_1^k, \ldots, y_m \text{ is } Y_m^k \quad (17)$$

where R_k is the k^{th} rule, $\mathbf{x} = (x_1, \ldots, x_n)^T \in U$ and $\mathbf{y} = (y_1, \ldots, y_m) \in V$ are the input and output state linguistic variables of the controller respectively, $U, V \subset \Re^n$ is the universe of discourse of the input and output variables respectively, $(X_1, \ldots, X_n) \subset U$ and $(Y_1, \ldots, Y_m) \subset V$ are the labels in linguistic term of input and output fuzzy sets, and n and m are the number of input and output states respectively.

In the application of robot manipulators tracking, the fuzzy logic controller considered uses independent joint controller structure which has been adapted by most industrial robots. Based on this structure the control torque τ_i', which is the input of joint i, is only a function of position error e_i and change of error $\dot{e}_i$ (velocity error). For joint i, the universe of discourse of the linguistic variable error, change of error and control torque are modeled by three fuzzy sets A, B and C respectively. Each linguistic variable could be described by 7 linguistic terms:

$$A: e_i = \{A_1 \ldots A_7\} = \{NB, NM, NS, Z, PS, PN, PB\}$$
$$B: \dot{e}_i = \{B_1 \ldots B_7\} = \{NB, NM, NS, Z, PS, PN, PB\}$$
$$C: \tau_i' = \{C_1 \ldots C_7\} = \{NB, NM, NS, Z, PS, PN, PB\}$$

where $N = Negative, P = Positive, B = Big, M = Medium, S = Small$ and $Z = Zero$. The Mamdani fuzzy rules in (17) are implemented in our two inputs and single output fuzzy controller in the following form:

$$\text{IF } (e_i \text{ is } A_p) \text{ AND } (\dot{e}_i \text{ is } B_q) \text{ THEN } \tau_i' \text{ is } C_r \quad (18)$$

where $e_i = q_{d_i} - q_i$ and $\dot{e}_i = e_{i_t} - e_{i_{t-1}}$ and p, q and $r = 1, \cdots, 7$ are indexes for fuzzy sets A, B and C respectively.

Symmetrical triangular membership functions are used for both input and output fuzzy sets A, B and C and are shown in figure 1, with no more than two overlaps of membership functions used for the fuzzification process. For each pair of $(e_i, \dot{e}_i)$, four fuzzy rules will be activated at each time. For each activated fuzzy rule, a firing strength α_i is associated with it that could be defined by using the triangular-norm [Lee, 1990]. Although there are a number of possible ways to obtain the t-norm such as algebraic product and min-operation etc., the min-operator is used as t-norm in our STFLC:

R_1 : IF $(e_i$ is $A_p)$ AND $(\dot{e}_i$ is $B_q)$ THEN τ_i' is C_{r1}
R_2 : IF $(e_i$ is $A_p)$ AND $(\dot{e}_i$ is $B_{q+1})$ THEN τ_i' is C_{r2}
R_3 : IF $(e_i$ is $A_{p+1})$ AND $(\dot{e}_i$ is $B_q)$ THEN τ_i' is C_{r3}
R_4 : IF $(e_i$ is $A_{p+1})$ AND $(\dot{e}_i$ is $B_{q+1})$ THEN τ_i' is C_{r4}

with rule firing strength:

$$\alpha_1 = \mu_{A_p}(e_i) \wedge \mu_{B_q}(\dot{e}_i) = min(\mu_{A_p}(e_i), \mu_{B_q}(\dot{e}_i))$$
$$\alpha_2 = \mu_{A_p}(e_i) \wedge \mu_{B_{q+1}}(\dot{e}_i) = min(\mu_{A_p}(e_i), \mu_{B_{q+1}}(\dot{e}_i))$$
$$\alpha_3 = \mu_{A_{p+1}}(e_i) \wedge \mu_{B_q}(\dot{e}_i) = min(\mu_{A_{p+1}}(e_i), \mu_{B_q}(\dot{e}_i))$$
$$\alpha_4 = \mu_{A_{p+1}}(e_i) \wedge \mu_{B_{q+1}}(\dot{e}_i) = min(\mu_{A_{p+1}}(e_i), \mu_{B_{q+1}}(\dot{e}_i))$$

where $\wedge$ represents the t-norm, A_p and A_{p+1} are the overlapped adjacent fuzzy sets corresponding to the crisp input value and similarly for B_q and B_{q+1}. The consequent part for the four activated rules resulted in four fuzzy sets $C_{r1}, \cdots, C_{r4} \subset C$, which are determined by the knowledge base as shown in table 1. Obviously, each activated rule contributes a consequent fuzzy description of the output torque τ_i' to (16).

The Mean of Maximum (MOM) defuzzification method is used in our STFLC to convert the fuzzy output back to the usable control signal. It generates a control action which represents the mean value of all local control actions whose membership functions reach the maximum. There are other defuzzification methods available [Mendel, 1995], such as the Center of Area (COA), which will yields a better steady state performance. The MOM is chosen because it gives good transient response and is more computationally efficient. The output of the fuzzy controller is:

$$\tau_i' = \frac{\sum_{i=1}^{4} \alpha_i \nu_i}{\sum_{i=1}^{4} \alpha_i} \quad (19)$$

where $\nu_i \in \Re$ is a crisp value of the apex of output fuzzy set.

5 Self-Tuning Algorithm

Adjustable parameters in the fuzzy logic controller include the scaling factors for input/output state variables, shape of fuzzification membership functions, and the knowledge base fuzzy rules. However, multiple simultaneous adjustments on scale factors, membership

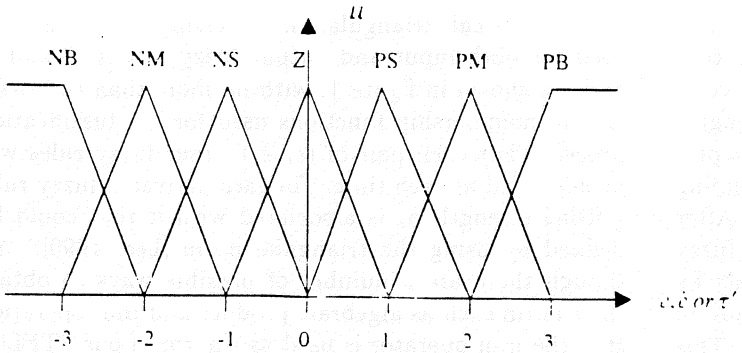

Figure 1: Membership function.

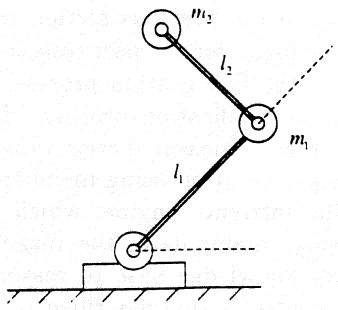

Figure 2: A two-link robot arm.

e \ $\dot{e}$	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	Z
NM	NB	NB	NB	NM	NS	Z	PS
NS	NB	NB	NM	NS	Z	PS	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PM	PB	PB
PM	NS	Z	PS	PM	PB	PB	PB
PB	Z	PS	PM	PB	PB	PB	PB

Table 1: Knowledge base rule

Joint No.	Mass(kg)	Length(m)	Inertia($kgm2$)
Shoulder	5	0.5	1.0
Elbow	3	0.4	0.8

Table 2: Robot model parameters.

functions and inference rules would make the optimum tuning more difficult and complicated. Thus, in this paper, we will only look at adjusting the scaling factors of its input variables online based on some performance measurements.

The fuzzy controller part consists of three scaling factors a, b and c for the universe of discourse of error and change of error as well as the output control torque respectively. The scaling factor of the output is fixed and has an value of $c = 0.333$ (3 fuzzy sets on each side) so that the output torque universe is normalized with τ' value lies in the range [-1,1]. In other words, the output from the controller represents the percentage of output torque to the robot. An output gain k_τ is set to the actuator saturation limit for converting the percentage value back to usable control signal to the system. The remaining two scaling factors a and b are set to unity initially and tuned according to the performance measurement. Adjusting a and b will cause the entire universe of discourse to contract when decreasing or expand when increasing. Indeed, it affects the fuzzification process and rule evaluation for generating the control output.

The tuning mechanism measures the maximum absolute value of error and change of error and then assigns the scaling factors as one third of the maximum values. The reason of using one third is that there are three fuzzy sets in the positive region of the universe of discourse of each input variable. By doing that, we can make sure that the entire universe of discourse of each input variables are covered and therefore produces a stable control to the system.

The tuning provides a stable control even with quite large initial errors to the system. The tuning mechanism uses the performance index that encompasses pass performance information on the system. Using this performance index, the scaling factors are re-adjusted such that the universe of discourse of the error e and change of error $\dot{e}$ should converge to some bounded values according to the pre-specified performance.

The auto-tuning law can be stated as follows: Define a_k and b_k as the scaling factors of error e and change of error $\dot{e}$ at any time instance t_k; ρ_{a_k} and ρ_{b_k} as the performance indexes of e and $\dot{e}$ respectively; σ as the number of sample for performance evaluation and λ as the number of membership function in the positive universe of discourse (in our example $\lambda = 3$). The following equations are applied to evaluate the performance indexes:

$$\rho_{a_k} = \frac{\sum_{k=1}^{\sigma} |e_{k+\sigma}|}{\sigma} \quad (20)$$

$$\rho_{b_k} = \frac{\sum_{k=1}^{\sigma} |\dot{e}_{k+\sigma}|}{\sigma} \quad (21)$$

. Then the following update laws are used for the scaling factors a_k and b_k :

$$a_k = a_{k-1} + \frac{\rho_{a_k} - \rho_{a_{k-1}}}{\lambda} \quad (22)$$

$$b_k = b_{k-1} + \frac{\rho_{b_k} - \rho_{b_{k-1}}}{\lambda} \quad (23)$$

with the initial value of a_0 and b_0 set to unity.

6 Simulation Results

Without losing generality, the simulation model used is a two-link robot manipulators shown in figure 2 even though the approach can be used to multi-link robots.

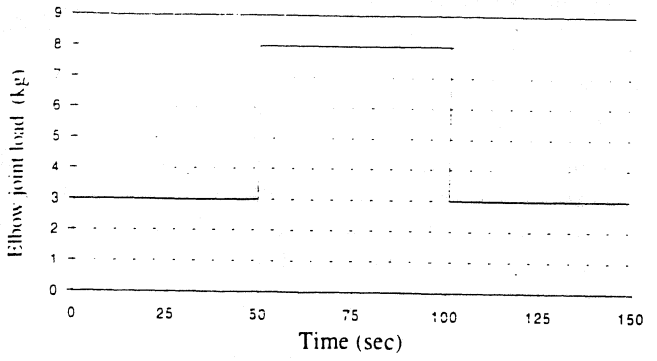

Figure 3: Load profile of elbow joint.

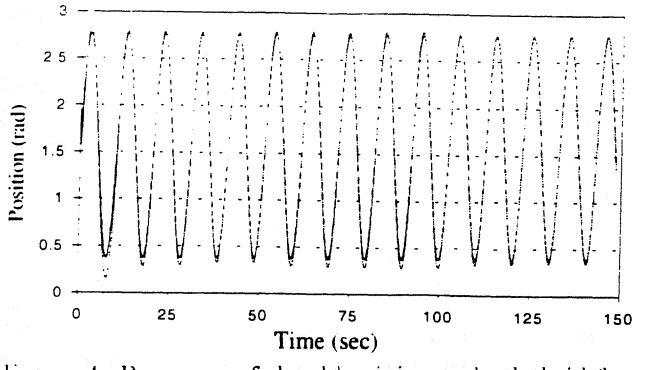

Figure 4: Response of shoulder joint under hybrid fuzzy control.

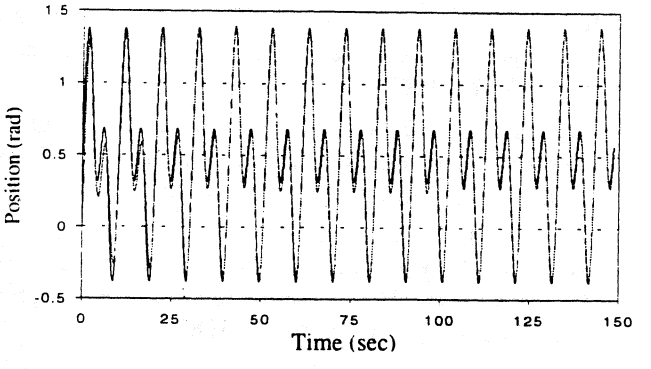

Figure 5: Response of elbow joint under hybrid fuzzy control.

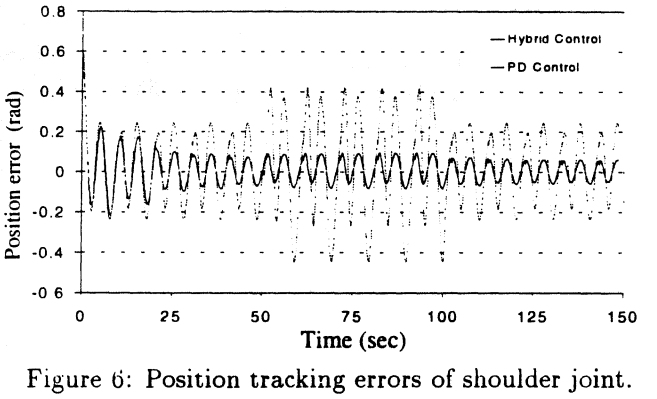

Figure 6: Position tracking errors of shoulder joint.

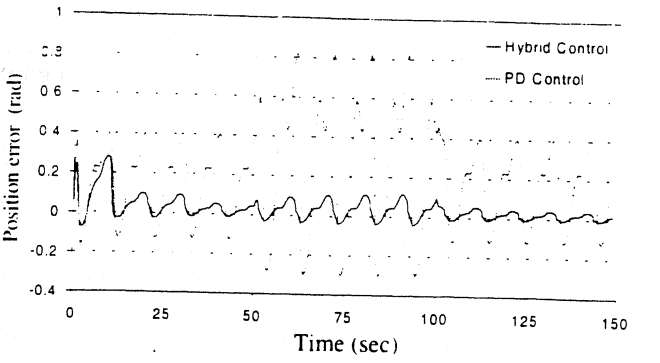

Figure 7: Position tracking errors of elbow joint.

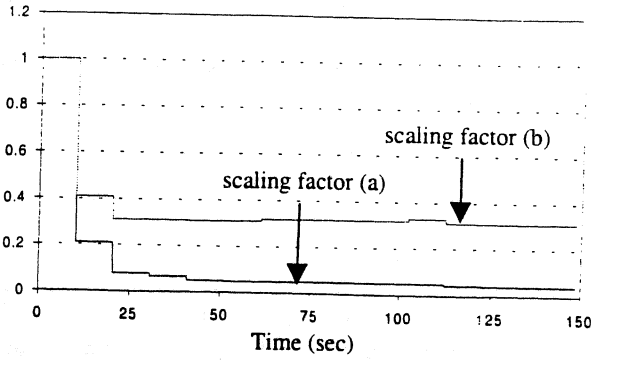

Figure 8: Shoulder joint scaling factors profile.

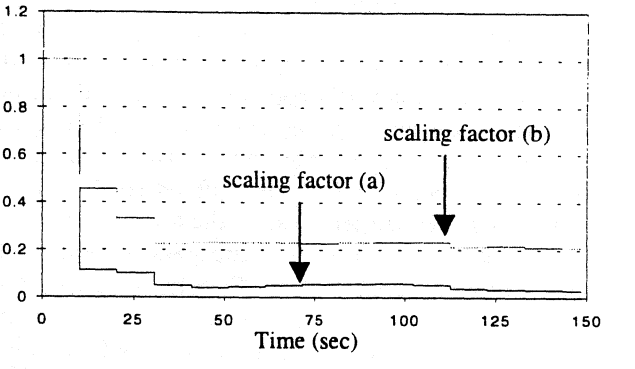

Figure 9: Elbow joint scaling factors profile.

Joint	PD	STFLC
Shoulder	$k_v = 30$	$a = 1\ b = 1$
	$\phi = 0.35$	$c = 0.333\ k_\tau = 20$
Elbow	$k_v = 20$	$a = 1\ b = 1$
	$\phi = 0.3$	$c = 0.333\ k_\tau = 20$

Table 3: Controller parameters.

The sampling frequency used for the simulation is 28 ms. which is the default sampling period used in real PUMA560 robot arm. Tables 2 and 3 show the model parameters and the control values used for PD and STFLC in the simulations. For the PD control part, k and ϕ are the gain constants as shown in (16). For the fuzzy control part, a, b and c are the scaling factors for the membership functions of error, change of error and output torque respectively, and k_τ is the gain for the final output torque representing the actuator saturation limit of the robot arm. As mentioned in section 5, a and b are set to unity initially, as they will self-tune themselves to meet the errors specification, and c was set to 0.333 throughout the experiment.

The robot arm is subjected to track simple sinusoidal trajectories with period of 5 seconds (0.2 Hz). The reference trajectories for the shoulder joint q_1 and the elbow joint q_2 are:

$$q_{d1} = 1.57 + 1.2sin\omega t \quad (24)$$

$$q_{d2} = 0.5 + 1.2(sin\omega t + sin2\omega t) \quad (25)$$

with $\omega = 1.26rad/sec$

The tracking performance of the proposed hybrid fuzzy controller and that of traditional PD controller are compared in the same condition that load variation at the elbow joint occurs. The load profile is shown in figure 3. As can be seen that it is increased from $3kg$ to $8kg$ at time $t = 50sec$, then dropped back to $3kg$ at $t = 100sec$. Figures 4 and 5 show the reference trajectories and joint responses of the robot arm.

In figures 6 and 7, we show the improvement of position tracking errors using our hybrid fuzzy controller over the traditional PD control method. As can be seen that the position tracking errors are reduced gradually at the beginning due to the self-tuning actions that drive the errors to an acceptable predefined level. Also, the effect of external disturbances in the form of load variation (from $t = 50$ to $t = 100$) is suppressed significantly by our hybrid fuzzy controller. Finally, in figures 8 and 9, the convergence of scaling factors of each joint is verified. The bounded tracking errors and scaling factors leads to bounded control torque for each joint in the simulation, which are omitted from here due to the space limitation.

7 Conclusion

For the trajectory tracking control of robot manipulators, we proposed a hybrid fuzzy controller which not only improves the tracking errors but also enhance system robustness. With the PD control part of the hybrid controller stabilizes the system, the fuzzy part featured with self-tuning mechanism can adjust the scaling factors of input universe of discourses to reduce the size of the tracking error and the effect of payload variations. Simulation results on a two-link robot arm verify the validity and effectiveness of the proposed hybrid fuzzy controller.

References

[Arimoto and Miyazaki, 1984] S. Arimoto and F. Miyazaki. On the stability of PID feedback with sensory information. In *Robotics Research*, eds. M. Brady and R. P. Paul, Cambridge: MIT Press, 1984.

[Hill, 1976] D. Hill and P. Moylan. The stability of nonlinear dissipative systems. *IEEE Trans. Automatic Control*, vol. AC-21, pp. 708-711, 1976.

[Kim and Bien, 1995] Y. T. Kim and Z. Bien. Robust self learning fuzzy logic controller. *IEEE inter. conf. on robotics and Automation*, pp. 1172, 1995.

[Lee, 1990] C. C. Lee. Fuzzy logic in control system: fuzzy controller - part I and II. *IEEE Trans. Syst. Man and Cybern.*, SMC-20: 404-435, Nov. 1990.

[Liu, 1997] M. Liu. Decentralized PD and robust nonlinear control for robot manipulators. *Journal of Intelligent and Robotics Systems*, vol. 20, no. 2-4, pp 319-332, 1997.

[Mamdani and Assilian, 1975] E. H. Mamdani and S. Assilian. An experimental in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, vol. 7, No.1:1-13, 1975.

[Mendel, 1995] J. M. Mendel. Fuzzy logic systems for engineering: a tutorial. *IEEE Proc.*, vol. 83, no. 3, pp. 245-377, 1995.

[Slotine and Li, 1987] J. J. Slotine and W. Li. On the adaptive control of Robot Manipulators. *Int. Journal of Robotic Research*, vol. 6, pp. 49-59, 1987.

[Tang and Mulholland, 1987] K. L. Tang and R. J. Mulholland. Comparing fuzzy logic with classical controller designs. *IEEE Trans. Syst. Man and Cybern.*, vol. SMC-17, pp. 1085, No. 6, 1987.

[Yi and Chung, 1997] S. Y. Yi and M. J. Chung. A robust fuzzy logic controller for robot manipulators with uncertainties. *IEEE Trans. Syst. Man and Cybern*, vol. 27, No. 4, pp. 706, 1997.

[Zadeh, 1965] L. A. Zadeh. Fuzzy set. *Journal of Information and Control*, vol. 8, pp. 338-353, 1965.