

Comparative Study of Some Adaptive Fuzzy Algorithms for Manipulator Control

Sudeept Mohan, and Surekha Bhanot

Abstract—The problem of manipulator control is a highly complex problem of controlling a system which is multi-input, multi-output, non-linear and time variant. In this paper some adaptive fuzzy, and a new hybrid fuzzy control algorithm have been comparatively evaluated through simulations, for manipulator control. The adaptive fuzzy controllers consist of self-organizing, self-tuning, and coarse/fine adaptive fuzzy schemes. These controllers are tested for different trajectories and for varying manipulator parameters through simulations. Various performance indices like the RMS error, steady state error and maximum error are used for comparison. It is observed that the self-organizing fuzzy controller gives the best performance. The proposed hybrid fuzzy plus integral error controller also performs remarkably well, given its simple structure.

Keywords—Hybrid fuzzy, Self-organizing, Self-tuning, Trajectory tracking.

I. INTRODUCTION

THERE are many control strategies that can be applied for control of a Robot arm. These range from conventional [1] to adaptive [2], [3], to more recent fuzzy [4], [5], [6] and adaptive fuzzy [7], [8] control strategies. In this paper an attempt has been made to do a comparative study of self-organizing and self-tuning fuzzy control schemes.

Fuzzy control of robotic manipulators has found vast interest in the control literature. Unlike Boolean logic, fuzzy logic deals with problems of vagueness, uncertainty or imprecision. It provides an extensive freedom for control designers to exploit their understanding of the problem and to construct intelligent control strategies. Nonlinear controllers can be devised easily by using fuzzy logic principles [9]. This makes fuzzy controllers powerful tools to deal with nonlinear systems.

The fuzzy control strategy consists of situation and action pairs, similar to how a human operator uses his experience to interpret the situation and initiate the control action. A human operator usually looks at the error and the change of error so as to arrive at a particular control action. A block diagram for the fuzzy controller is shown in Fig. 1.

Manuscript received June 21, 2006.

Sudeept Mohan is with Computer Science Department, Birla Institute of Technology and Science, Pilani, 333031, India. (phone: 91-1596-245073 ext: 291; fax: 91-1596-244183; e-mail: sudeeptm@bits-pilani.ac.in).

Surekha Bhanot is with Instrumentation Department, Birla Institute of Technology and Science, Pilani, 333031, India (e-mail: surekha@bits-pilani.ac.in)

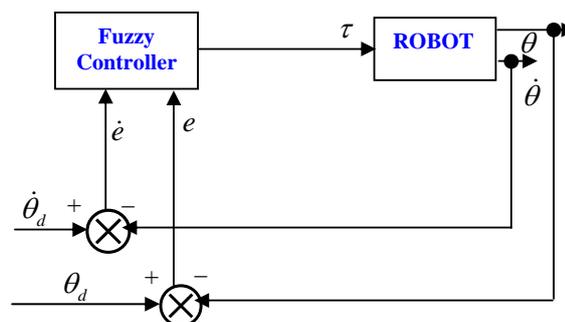


Fig. 1 Block diagram of Fuzzy Controller

The fuzzy controller here defines error (e) as

$$e = \theta_d - \theta \quad (1)$$

and rate of change of error (\dot{e}) as

$$\dot{e} = \dot{\theta}_d - \dot{\theta} \quad (2)$$

τ is the output of fuzzy controller applied as control input to the robot system. A detailed view of internal of the Fuzzy controller block shown in Fig. 1 is shown in Fig. 2.

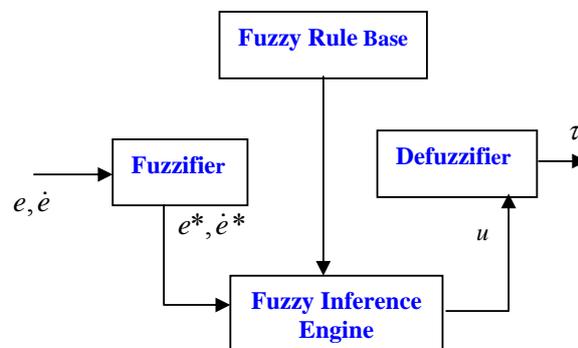


Fig. 2 Details of Fuzzy controller block

For the simulation study carried out in this paper the input variables to the fuzzy controller (e , \dot{e}) are quantized into thirteen levels represented by $-6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6$, and a set of linguistic variables such as Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZE), Positive Small (PS), Positive Medium (PM), Positive Big (PB) are assigned.

The next step in the design of the fuzzy controller is to

decide the membership functions for the linguistic variables. The decision regarding the type of the membership function is arbitrary and depends on the choice of the user. Here, we have selected the triangular membership function as shown in Fig. 3.

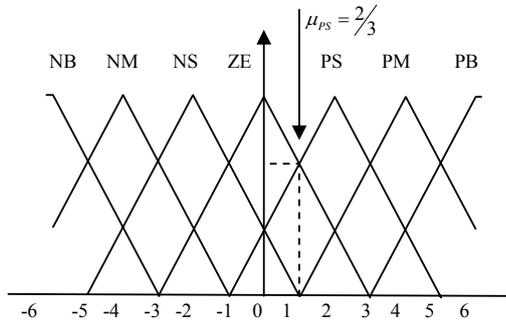


Fig. 3 Membership functions of the Linguistic variables

The control rules are formulated in a manner to represent the operator's experience regarding the system behavior.

Some of the rules that were formulated are

R1: If e is ZE and \dot{e} is ZE, then u is ZE.

R2: If e is ZE and \dot{e} is NS, then u is NS.

R3: If e is NM and \dot{e} is ZE, then u is NM.

R4: If e is NM and \dot{e} is NB, then u is NB.

The complete rule base for the fuzzy controller is shown in Table I.

TABLE I
FUZZY CONTROLLER RULE BASE

\dot{e}/e	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NM	NS	NS	ZE
NM	NB	NM	NM	NM	NS	ZE	PS
NS	NB	NM	NS	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PS	PM	PB
PM	NS	ZE	PS	PM	PM	PM	PB
PB	ZE	PS	PS	PM	PB	PB	PB

These rules constitute the knowledge base of the fuzzy controller. The rule strength of the individual rule is evaluated using the intersection operation defined as

$$\mu_{NS}(u) = \min(\mu_{NM}(e^*), \mu_{NB}(\dot{e}^*)) \quad (3)$$

where $\mu_{NS}(u)$ is the rule strength of the rule R4, $\mu_{NM}(e^*)$ is the membership of the crisp input e^* in the fuzzy set NM and $\mu_{NB}(\dot{e}^*)$ is the membership of \dot{e}^* in the fuzzy set NB.

For each possible combination of e^* and \dot{e}^* , the rules are fired individually to give the degree to which the rule antecedent has been matched by the crisp value. The clipped values for the individual rules thus obtained are aggregated forming the overall control values. The output value is then

defuzzified by using the center of gravity method, which is given by

$$u^* = \frac{\sum_{Ri} \mu_{Ri}(u_{Ri}) \cdot u_{Ri}}{\sum_{Ri} \mu_{Ri}(u_{Ri})} \quad (4)$$

The output values thus obtained for all the (e^*, \dot{e}^*) pairs are stored in the form of a lookup table (LUT) as shown in Table II.

The array implementation improves execution speed, as the run-time inference is reduced to a table look-up which is a lot faster, at least when the correct entry can be found without too much searching. The controller output values shown in the Table II were obtained after some manual adjustment through trial and error to give best possible results. This was required because the manipulator control problem is highly nonlinear and the rules formulated through user experience are not always correct under different situations.

The control strategies were tested for a two-link manipulator. Fig. 4 shows the manipulator with frames assigned to the links.

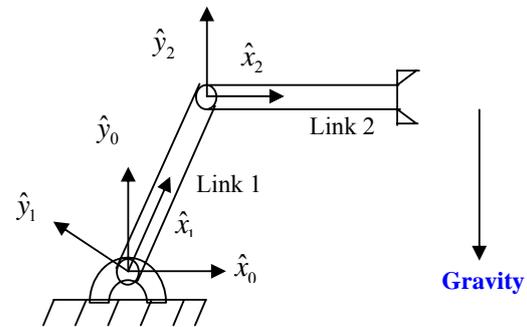


Fig. 4 Manipulator used for simulations

The inverse dynamics can be derived using the Lagrange or Newton-Euler method. The joints were assumed to have only viscous friction. This model was used for all simulations.

II. SELF-ORGANIZING FUZZY CONTROL (SOFC)

A Fuzzy controller consists of three major components that can be altered to give different controller behaviors. These three components are:

- The normalization and denormalization scaling factors
- The fuzzy set representing the meaning of linguistic values
- The if-then rule base

If the above three components remain fixed the fuzzy controller is of type *non-adaptive*. If on the other hand any of the above three components are altered when the controller is running, it is known as *adaptive-fuzzy*. Adaptive Fuzzy controller that modifies the rule base is known as 'self-organizing' controller. These controllers can start with a non-zero rule base and then modify it (**SOFC NZLUT**) or they can build the rule base entirely afresh starting with all zeros in LUT (**SOFC ZLUT**).

TABLE II
LOOKUP TABLE FOR THE FUZZY CONTROLLER

→ e ↓ ė	MEMBERSHIP FUNCTION												
	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	-5.6	-5.4	-5.0	-4.8	-4.8	-4.7	-4.7	-4.6	-4.5	-4.4	-4.3	-4.3	-4.2
-5	-4.7	-4.5	-4.4	-4.3	-4.2	-4.1	-4.0	-3.9	-3.8	-3.8	-3.7	-3.6	-3.5
-4	-3.7	-3.6	-3.5	-3.2	-3.0	-3.0	-3.0	-2.9	-2.9	-2.8	-2.8	-2.7	-2.7
-3	-2.0	-2.0	-1.9	-1.9	-1.8	-1.8	-1.7	-1.7	-1.6	-1.5	-1.4	-1.3	-1.3
-2	0.0	0.0	-0.8	-1.0	-1.2	-1.7	-2.3	-2.2	-2.2	-2.0	-2.0	-1.0	-1.0
-1	1.0	1.0	0.0	0.0	-0.5	-0.5	-0.5	-1.0	-1.2	-1.5	-1.7	-1.0	-1.0
0	1.3	1.2	1.0	0.8	0.6	0.0	-0.2	-0.4	-0.6	-0.8	-1.0	-1.0	-1.0
1	2.0	2.0	1.9	1.8	1.8	1.8	1.8	1.8	1.5	0.0	-0.3	-1.0	-0.8
2	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	1.2	0.8	0.0	0.0
3	2.0	2.1	2.3	2.5	2.5	2.5	2.6	2.7	2.8	2.8	2.9	2.9	3.0
4	2.7	2.7	2.8	3.1	3.2	3.3	3.5	3.6	3.6	3.8	3.8	3.9	3.9
5	3.6	3.3	3.7	4.0	4.1	4.3	4.3	4.4	4.4	4.5	4.5	4.6	4.7
6	4.4	4.4	4.3	4.8	5.0	5.0	5.1	5.2	5.3	5.4	5.6	5.6	5.6

Many self-organizing controllers have been proposed in literature [10], [11]. The SOFC controller tested in this paper is described in detail in [12] and consists of three main blocks as shown in Fig. 5.

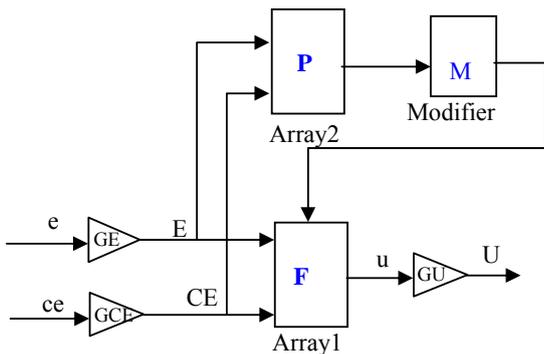


Fig. 5 Block diagram of SOFC (Jantzen, 98)

The first block marked **F** is a look up table (LUT) based fuzzy controller. It is exactly same as described in the previous section. The inputs to this block are error (e) and change in error (ce), which are normalized through input scaling factors GE and GCE respectively. The output of **F** is u, which is denormalized by the output-scaling factor GU to produce the final output U.

The second block is marked **P** and contains a performance measure, which can be used to decide if the lookup table in **F** needs to be modified. **P** also consists of a lookup table, which is of same size as that of **F**. This lookup table indicates the *desired* behavior of the controller and is used to decide if the entries of lookup table in **F** need to be modified or not. The performance table used for our simulations is shown in Table III. The performance table **P** evaluates the current state and returns a performance measure $P(i_n, j_n)$, where i_n is the index corresponding to E_n , and j_n is the index corresponding to CE_n .

The third block is M, which contains a modifier algorithm.

It modifies entries of **F** according to the present calculated performance index $P(i_n, j_n)$.

For our simulations, the rule used in M is given as

$$\mathbf{F}(i, j)_{n-1} = \mathbf{F}(i, j)_{n-1} + \mathbf{P}(i, j)_n \quad (5)$$

Where n denotes current sample. As seen from (5) the present performance index is used to modify the previous entry used in **F**.

TABLE III
PERFORMANCE TABLE (YAMAZAKI, 1982)

ė e	Membership Function												
	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	-6	-6	-6	-6	-6	-6	-6	-5	-4	-3	-2	-1	0
-5	-6	-6	-6	-6	-5	-4	-4	-4	-3	-2	-1	0	0
-4	-6	-6	-6	-5	-4	-3	-3	-3	-2	-1	0	0	1
-3	-6	-6	-5	-4	-3	-2	-2	-2	-1	0	0	1	2
-2	-6	-5	-4	-3	-2	-1	-1	-1	0	0	1	2	3
-1	-5	-4	-3	-2	-1	-1	0	0	0	1	2	3	4
0	-5	-4	-3	-2	-1	0	0	0	1	2	3	4	5
1	-3	-2	-1	0	0	0	0	1	1	2	3	4	5
2	-2	-1	0	0	0	1	1	1	2	3	4	5	6
3	-1	0	0	0	1	2	2	2	3	4	5	6	6
4	0	0	0	1	2	3	3	3	4	5	6	6	6
5	0	0	1	2	3	4	4	4	5	6	6	6	6
6	0	1	2	3	4	5	6	6	6	6	6	6	6

III. SELF-TUNING FUZZY CONTROLLER (STFC)

A Controller that changes scaling factors or modifies the fuzzy set definitions is known as ‘self-tuning’ controller. Many such controllers have been proposed in literature [13], [14]. The Adaptive Fuzzy controller that we investigated is of PD type and is proposed by Mudi, and Pal [15]. The output gain (denormalization, GU) of this controller is adjusted on-line depending on the present values of error and error derivative.

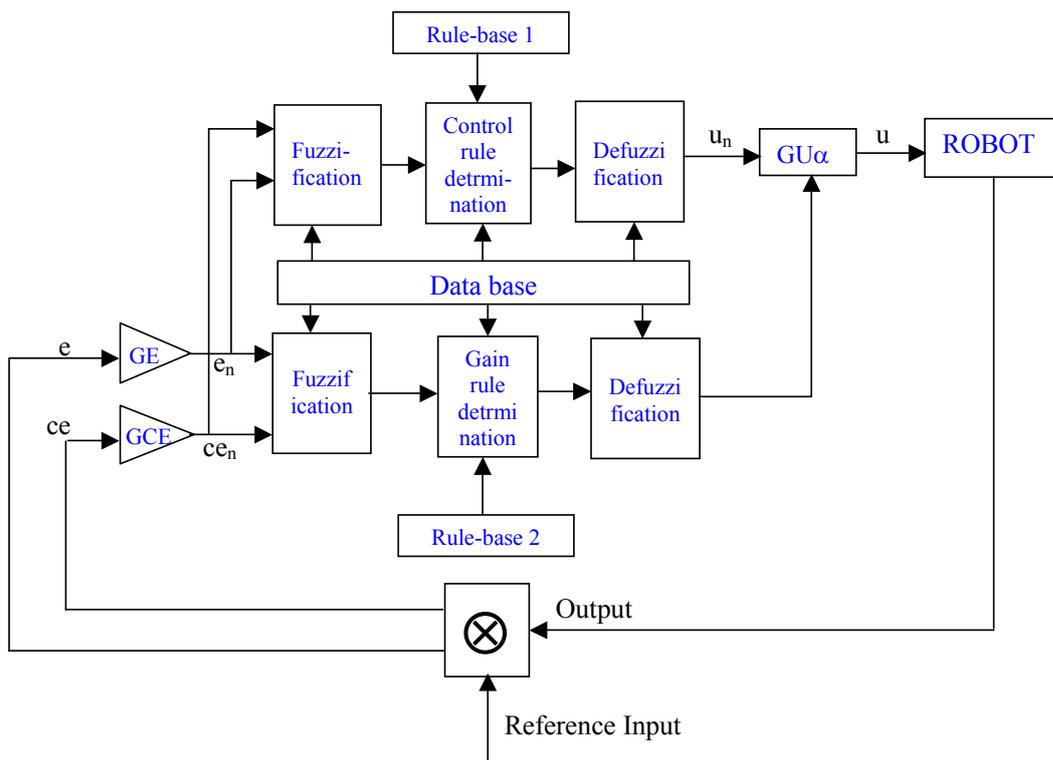


Fig. 6 Block diagram of the self-tuning controller (adapted from Mudi, 1999)

Thus the controller is of self-tuning type. The block diagram of the self-tuning fuzzy controller is shown in Fig. 6. The membership functions for controller inputs (error and error derivative) and output are defined on the common interval [-6 6] and are same as shown in Fig. 3. The membership functions for gain updating factor (α) are defined on [0 1]. These membership functions are shown in Fig. 7.

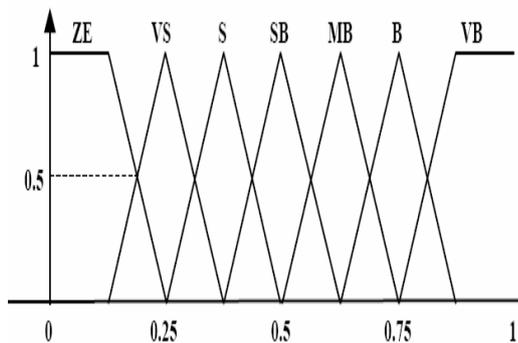


Fig. 7 Membership functions for gain updating factor (α)

For the conventional fuzzy controller the controller output is mapped to the respective actual output by the output gain GU. On the other hand in the self-tuning fuzzy controller the actual output is obtained by multiplying the controller output with $GU\alpha$. The gain-updating factor α is calculated on-line using a model independent fuzzy rule base which has e and

\dot{e} as inputs. The governing equations for this self-tuning fuzzy controller are given below.

$$e_n = GE.e \tag{6}$$

$$ce_n = GCE.ce \tag{7}$$

$$u = \alpha.GU.u_n \tag{8}$$

The Fuzzy controller used the rule base and membership functions as discussed in section 1. However unlike SOFC the controller implementation is not lookup table based. The gain updating part of the controller produces output based on rules of the form:

$$R_i: \text{If } e \text{ is } E \text{ and } ce \text{ is } CE \text{ then } \alpha \text{ is } \alpha$$

The complete rule base used for updating α is shown in Table IV.

The parameter α is independent of any manipulator parameter and depends only on current system states. Thus the self-tuning scheme is largely independent of the process being controlled.

The following steps were used for tuning the controller:

Assuming that $\alpha=1$, we first adjust the value of GE so that the normalized error covers the entire domain [-6 6] to make efficient use of rule base. We then adjust the values of GCE and GU to make the output as acceptable as possible. This process is done through trial and error for any one trajectory.

TABLE IV
FUZZY RULE BASE FOR α

\dot{e}/e	NB	NM	NS	ZE	PS	PM	PB
NB	VB	VB	VB	B	SB	S	ZE
NM	VB	VB	B	B	MB	S	VS
NS	VB	MB	B	VB	VS	S	VS
ZE	S	SB	MB	ZE	MB	SB	S
PS	VS	S	VS	VB	B	MB	VB
PM	VS	S	MB	B	B	VB	VB
PB	ZE	S	SB	B	VB	VB	VB

The output-scaling factor (GU) is now set to three times (to keep the rise time almost same) the value found in previous step. The other two scaling factors are kept same as determined in previous step. α is no longer fixed at 1 but is calculated on-line from its rule base.

IV. HYBRID FUZZY+INTEGRAL ERROR CONTROLLER (HFIE)

The PD self-tuning adaptive controller investigated in previous sections gives a reasonably good performance but as will be seen in later sections suffers from considerable steady state errors.

With a view to reduce this steady state error we investigated a simple hybrid fuzzy controller. Many hybrid Fuzzy controllers have been proposed in the past [16], [17]. The block diagram of our controller is shown in Fig. 8. This controller consists of two parts. First is a simple non-lookup table based fuzzy controller. This is same as that used for STFC.

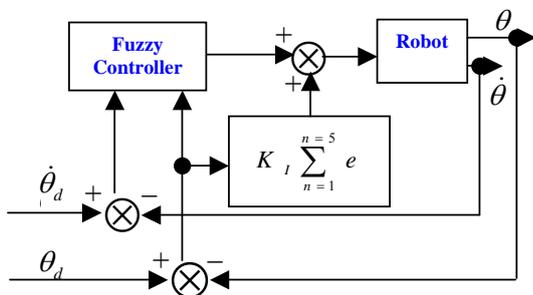


Fig. 8 Hybrid Fuzzy plus Integral error Controller (HFIE) block diagram

Second part is an integral error controller, as shown in Fig. 8. Usually the integral part of a controller produces an output, which is proportional to integral of error over the entire period of motion. But this simple addition of integral term also increases the order of system and might result in unstable closed loop system. The controller that we propose **does not** perform summation (integration) of error over entire period of motion.

For our simulations the trajectory generator provides the controller with information about the desired position, velocity and acceleration ($\theta_d, \dot{\theta}_d, \ddot{\theta}_d$) for each joint and

keeps updating this information at the path update rate which has been chosen as 3ms (333Hz). The controller takes this information and compares it with the present (actual) position and velocity of joints ($\theta, \dot{\theta}$), which are provided as feedback through the sensors. Based upon the error between the desired and actual values, the controller calculates a vector of joint torques (τ), which should be applied at respective joints by the actuators to minimize these errors. In the simulations, the control loop runs five times for every set point supplied by the trajectory generator. The integral action of our controller is limited to summing up these five errors for every set point provided by the trajectory generator. The sum of these errors is reset to zero every time the trajectory generator gives a new set point. This type of integral action cannot of course give zero values of steady state error but can nevertheless reduce them. Further the overall resulting controller does not suffer from danger of instability.

V. COARSE/FINE ADAPTIVE FUZZY CONTROLLER (CFAF)

When a controller is required to operate under conditions of both large and small excursions of its inputs from their nominal values, it is convenient to use *two* or more sets of fuzzy rules to effect improved control. For large excursions of the controller input variables, coarse control is applied with the objective of forcing the plant to return to its nominal operating point as rapidly as possible. This can be achieved by using only a few rules. When the plant variables reach some small region about the nominal operating point then fine control is applied. Here a new set of control rules necessary to effect the desired fine control actions are used and these involve a larger number of rules and fuzzy sets. Under normal operating conditions the controller uses fine control whereas under situation of disturbance or large change in set point, it uses coarse control.

An alternative way of achieving coarse-fine control is through *zooming* of the universe of discourse of each controller input variable. In this case the universe of discourse is varied in discrete regions in control space as the plant approaches the desired operating point. This approach has been used to great effect for the control of high precision mechatronic devices and is investigated in this paper for effectiveness in case of mechanical manipulator.

The basic controller is still the self-tuning adaptive fuzzy (STFC) controller discussed in section 3. In that controller the output-scaling factor alone is adapted via the variable gain factor α . The characteristics of a PI- or PD-type fuzzy logic controller depends on both input and output scaling factors, i.e., for the best performance, simultaneous adjustment of both input and output SF's is more justified. To this effect the proposed controller normalizes the position and velocity errors to limit them to domain [-6 6]. It then checks if the position and velocity errors are both within [-3 3]. If they are, then both the position and velocity error are doubled to provide the zooming effect. If the errors are not within [-3 3],

then they are used as they are, without being doubled. This simple strategy results in much improved performance of the controller as discussed in the following sections.

VI. SIMULATION DETAILS

The above controllers were tested for two different trajectories. In the first trajectory, the first joint was required to move from its initial home position (0°) to a final position of $+90^\circ$ in 5 seconds. On reaching the final position the manipulator picks up a load and returns back to its home position in another 5 seconds. On reaching the home position the manipulator was required to stay there with the load for another 5 seconds. Thus the desired position of first joint remains constant at 90° for the last 5 seconds of its motion. This kind of trajectory enables us to test the steady state performance of the controller. The desired motion for the second joint is exactly the same as for the first one except that it is required to move from 0° to -90° and then back to 0° in a total time of 15 seconds. Fig. 9 shows the desired joint position profiles for this trajectory.

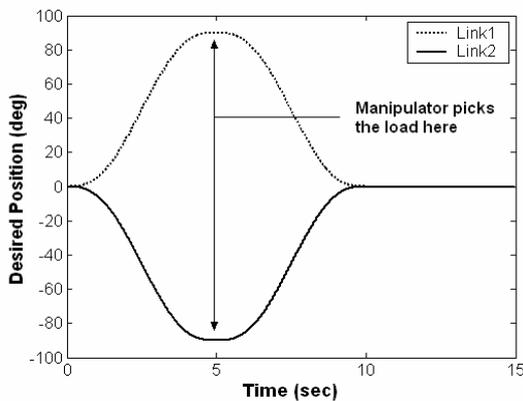


Fig. 9 Desired Trajectory 1

The second test trajectory was chosen to simulate the motion of manipulator during a typical pick and place operation. Here the manipulator's first joint was required to move from its home position of 0° to a final position of $+45^\circ$ in 2 seconds. At this point the manipulator picks up a load and returns back to its home position in the next 2 seconds. On reaching home the manipulator releases the load and this cycle is repeated all over again. The second joint of the manipulator has a motion similar to the first one except that it moves to a final position of -45° . The errors for this trajectory were traced for two cycles, i.e., 8 seconds. The RMS and the maximum values of the errors were used for quantitative performance comparisons of various controllers for this trajectory. Fig. 10 shows the joint motion profiles for this trajectory.

The controllers were tested using the above two trajectories. The parameters of the manipulator were further assumed to have changed to new values whenever it picked up a load. The actual and new values of the parameters of manipulator with load were taken as below. The simulations were done using

the C language and MATLAB.

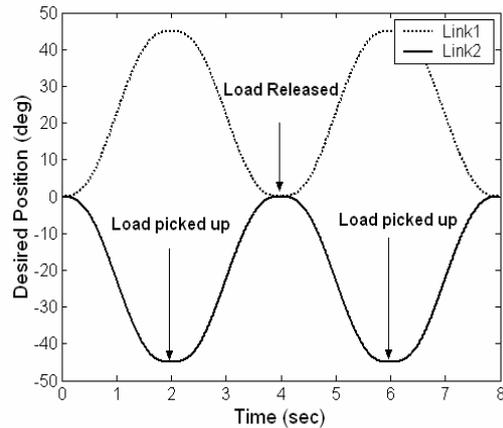


Fig. 10 Desired Trajectory 2

ACTUAL PARAMETERS

$$m_1 = 2.0 \text{ kg}$$

$$m_2 = 2.0 \text{ kg}$$

$$l_1 = 0.26 \text{ m}$$

$$x_1 = 0.13 \text{ m}$$

$$x_2 = 0.14 \text{ meters}$$

$$I_{zz1} = 0.09 \text{ kg} - \text{m}^2$$

$$I_{zz2} = 0.09 \text{ kg} - \text{m}^2$$

$$F_1 = 2.5 \text{ N} - \text{m} / \text{rad} / \text{sec}$$

$$F_2 = 2.5 \text{ N} - \text{m} / \text{rad} / \text{sec}$$

PARAMETERS WITH LOAD

$$m_1 = 3.0 \text{ kg}$$

$$m_2 = 3.0 \text{ kg}$$

$$l_1 = 0.26 \text{ m}$$

$$x_1 = 0.15 \text{ m}$$

$$x_2 = 0.16 \text{ meters}$$

$$I_{zz1} = 1.5 \text{ kg} - \text{m}^2$$

$$I_{zz2} = 0.09 \text{ kg} - \text{m}^2$$

$$F_1 = 2.5 \text{ N} - \text{m} / \text{rad} / \text{sec}$$

$$F_2 = 2.5 \text{ N} - \text{m} / \text{rad} / \text{sec}$$

where

m_1 and m_2 are masses of link 1 and link 2 respectively

l_1 is the length of link 1

F_1 and F_2 are coefficients of viscous friction of link 1 and link 2 respectively

x_1 and x_2 are locations of the center of mass of link 1 and link 2 respectively along the respective x- axis

I_{zz1} and I_{zz2} are moments of inertia of link 1 and link 2 respectively about respective Z axis.

TABLE V
ERRORS (IN DEGREES) FOR DIFFERENT CONTROLLERS

S.No	CONTROL STRATEGY	TRAJECTORY NO.1				TRAJECTORY NO.2			
		link1 0°→90°→0°→		link2 0°→-90°→0°→		link1 0°→45°→0°→45°→0°		link2 0°→-45°→0°→-45°→0°	
		RMS	SS	RMS	SS	RMS	MAX	RMS	MAX
1.	Pure Fuzzy	0.2347	0.2345	0.1206	0.0590	0.2907	0.4088	0.1579	0.2774
2.	SOFC ZLUT	0.0245	0.0166	0.0333	-0.0169	0.0477	0.1059	0.0395	-0.1032
3.	SOFC NZLUT	0.0271	0.0166	0.0305	-0.0166	0.0380	-0.1014	0.0387	0.0898
4.	STFC	0.1707	0.1819	0.1010	0.0784	0.1923	0.2384	0.1147	0.1651
5.	HFIE	0.0381	0.0813	0.0292	0.0208	0.1026	0.1474	0.0335	0.0498
6.	CFAF	0.0846	0.0910	0.0435	0.0392	0.0961	0.1266	0.0448	0.0644

VII. DISCUSSION OF RESULTS

The values of different errors for various control strategies are tabulated in Table V.

Following observations are made based on our simulation studies:

1. From the simulations carried out for self-organizing controller with zero lookup table (SOFC ZLUT), it was seen that its performance is better than Pure Fuzzy controller. This is primarily because SOFC can auto-tune the LUT as the manipulator parameters change whereas Pure Fuzzy controller cannot. The error profiles for the two link of manipulator for this controller are shown in Fig. 11 for trajectory 1.

2. The self-organizing controller was also tested with a non-zero LUT (SOFC NZLUT). This LUT was same as that used for pure fuzzy controller. As can be seen from Table V, the errors for the two links go down further when compared to SOFC ZLUT. However this reduction in error is not very significant. Moreover it was observed that if we start with a non-optimized version of lookup table, the errors could be higher. Thus a non-zero LUT should be used only when the operator experience has been properly incorporated in it. The error profiles for the two link of manipulator for this controller are shown in Fig. 12 for trajectory 1.

3. The STFC gives significant error reduction when compared to pure fuzzy counterpart. However it is not as good as SOFC. Moreover the number of calculations required for STFC is much more compared to SOFC. However the error profiles for STFC are smoother compared to SOFC. This will translate into smoother manipulator motion. This is mainly because the STFC is not based on lookup tables. Fig. 13 shows a comparison of Errors for STFC Vs Pure Fuzzy control for trajectory 1.

4. The proposed HFIE controller performs remarkably well when compared to STFC. The HFIE controller gives better performance with r.m.s., maximum and steady state errors all getting reduced, when compared to STFC. Fig. 14 shows this comparison for trajectory 1. HFIE performance however, is not as good as either versions of SOFC. The simulations demonstrate the effectiveness of the modified integral action in reducing the overall errors for manipulator trajectory tracking. This improved performance is further achieved with having much less number of calculations to perform compared to adaptive fuzzy controllers. Although for our simulations we have kept the sampling rate for both adaptive fuzzy and HFIE controllers same, the much higher possible sampling rates for HFIE controller will improve its performance further.

5. Fig. 15 shows a comparison of errors between STFC and the proposed CFAF controller for trajectory 1. It is observed that the CFAF controller gives much improved performance compared to STFC. Errors for both links for CFAF controller are reduced. The CFAF controller also does not involve any additional computational burden when compared to STFC. The additional complexity is only in terms of few additional if-then-else statements. The CFAF controller however is still not as good as SOFC or HFIE controller.

6. The self-organizing fuzzy controller gives the best performance and is followed by HFIE. The self-tuning controller (CFAF included) has substantial additional computational burden and still do not perform better than SOFC or HFIE.

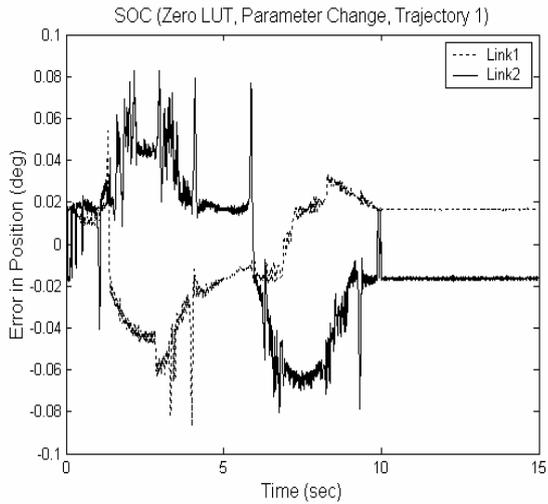


Fig. 11 Errors for SOFC control (Zero LUT, Trajectory 1)

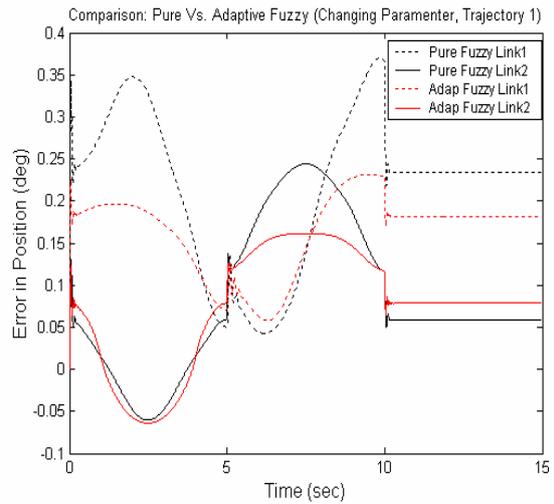


Fig. 13 Comparison of Errors for STFC Vs Pure Fuzzy control (Trajectory 1)

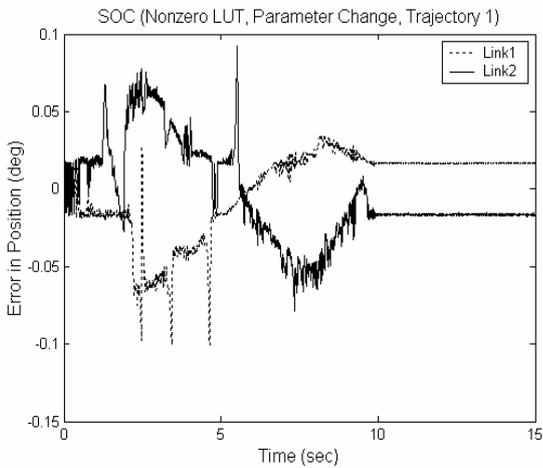


Fig. 12 Errors for SOFC control (Nonzero LUT, Trajectory 1)

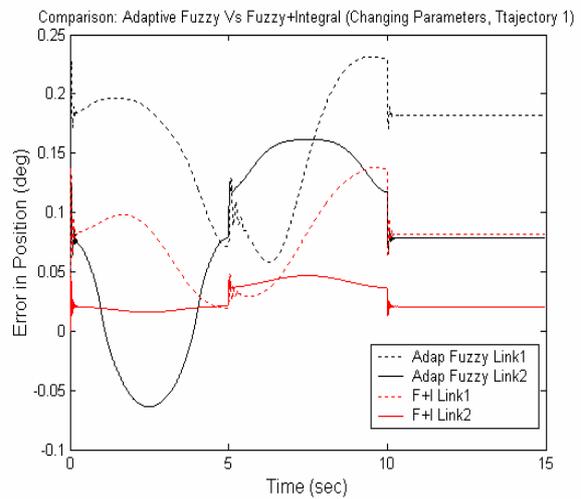


Fig. 14 Comparison of Errors for HFIE Vs STFC (Trajectory 1)

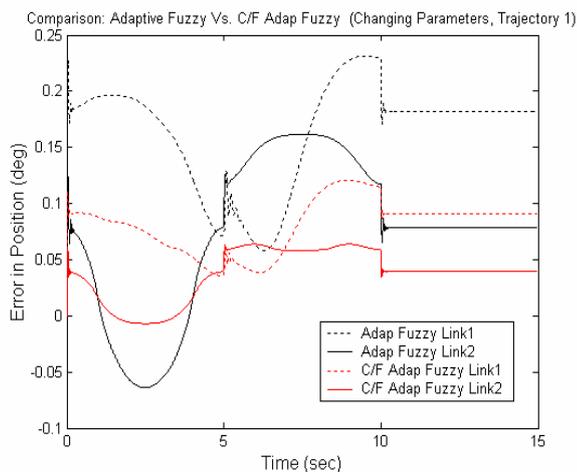


Fig. 15 Comparison of Errors for CFAF controller Vs STFC (Trajectory 1)

REFERENCES

- [1] Luh, J. Y. S., 'Conventional Controller Design for Industrial Robots - A Tutorial', IEEE Transactions on Systems, Man, and Cybernetics, 1983, Vol. SM- 13, no. 3, Pages: 298-316.
- [2] Hsia, T., 'Adaptive control of robot manipulators-A review', Proceedings IEEE International Conference on Robotics and Automation, April 1986, Vol. 3, Pages: 183 – 189.
- [3] Tosunoglu, S., Tesar, D., 'State of the Art in Adaptive control of robotic systems', IEEE Transactions on Aerospace and Electronic Systems, Sept. 1988, Vol. 24, Issue: 5, Pages: 552 – 561.
- [4] Er, M.J., 'Recent developments and futuristic trends in robot manipulator control', Proceedings IEEE Asia-Pacific Workshop on Advances in Motion Control, July 1993 Pages: 106 – 111.
- [5] Banerjee, S., Peng Yung Woo, 'Fuzzy logic control of robot manipulator', Proceedings Second IEEE Conference on Control Applications, Sept. 1993, Vol.1, Pages:87 – 88.
- [6] Zhou, J., Coiffet, P., 'Fuzzy Control of Robots', Proceedings IEEE International Conference on Fuzzy Systems, March 1992, Pages: 1357 – 1364.
- [7] Byung Kook Yoo, Woon Chul Ham, 'Adaptive Control of Robot Manipulator Using Fuzzy Compensator', IEEE Transactions on Fuzzy Systems, April 2000, Vol. 8, Issue: 2, Pages:186 – 199.

- [8] Purwar, S., Kar, I.N., Jha, A.N., 'Adaptive control of robot manipulators using fuzzy logic systems under actuator constraints', Proceedings of IEEE International Conference on Fuzzy Systems, 25-29 July 2004, Vol 3, Pages: 1449 - 1454.
- [9] Nagrath, I.J., Pahade Paras Shripal, Chand, A., 'Development and implementation of intelligent control strategy for robotic manipulator' IEEE/IAS International Conference on Industrial Automation and Control, 1995, Pages: 215 - 220.
- [10] Kazemian, H.B., 'The Self Organizing Fuzzy PID Controller', IEEE World Congress on Computational Intelligence., 4-9 May 1998, Vol. 1, Pages:319 - 324.
- [11] Koh, K.C., Cho, H.S., Kim, S.K., Jeong, I.S., 'Application of a self-organizing fuzzy control to the joint control of a Puma-760 robot', Proceeding of IEEE International Workshop on Intelligent Robots and Systems, 3-6 July 1990, vol.1, Pages: 537 - 542.
- [12] Jantzen J., 'The Self-Organizing Fuzzy Controller', Tech. report no 98-H 869 (soc), Technical University of Denmark, 19 Aug 1998.
- [13] Llama, M.A., Kelly, R., Santibanez, V., 'Control of Robot Manipulators via fuzzy self tuning', IEEE Transactions on Systems, Man and Cybernetics, Feb. 2000, Part B, , Vol. 30 , Issue: 1 ,Pages:143 - 150.
- [14] Chatterjee, A., Watanabe, K., 'An adaptive fuzzy strategy for motion control of robot manipulators' ACM Journal of Soft Computing, March 2005, Vol. 9, Issue: 3, Pages: 185 - 193.
- [15] Mudi, R.K., Pal, N.R., 'A robust self-tuning scheme for PI- and PD-type fuzzy controllers', IEEE Transactions on Fuzzy Systems, Feb. 1999, Vol. 7, Issue: 1, Pages: 2 - 16.
- [16] Lin, C., Mon, Y., 'Hybrid adaptive fuzzy controllers with application to robotic systems', Fuzzy sets and systems, 2003, Vol. 139, Pages: 151-165.
- [17] Sun, Y., Joo, M., 'Hybrid Fuzzy Control of Robotics Systems', IEEE Transactions on Fuzzy Systems, Dec. 2004, Vol. 12, Issue 6, Pages: 755 - 765.