Fuzzy Learning Control for a Flexible-Link Robot

Vivek G. Moudgal, Waihon Andrew Kwong,
Kevin M. Passino, *Member, IEEE*, and Stephen Yurkovich, *Senior Member, IEEE*

***Abstract*—Fuzzy control has emerged as a practical alternative to several conventional control schemes since it has shown success in some application areas; however, there are several drawbacks to this approach: i) the design of fuzzy controllers is usually performed in an *ad hoc* manner where it is often difficult to choose some of the controller parameters (e.g., the membership functions), and ii) the fuzzy controller constructed for the nominal plant may later perform inadequately if significant and unpredictable plant parameter variations occur. In this paper we illustrate these two problems on a two-link flexible robot testbed by i) developing, implementing, and evaluating a fuzzy controller for the robotic mechanism, and ii) illustrating that payload variations can have negative effects on the performance of a well designed fuzzy control system. Next we show how to develop and implement a "fuzzy model reference learning controller" (FMRLC) [1]–[5] for the flexible robot and illustrate that it can: i) automatically synthesize a rule-base for a fuzzy controller that will achieve comparable performance to the case where it was manually constructed, and ii) automatically tune the fuzzy controller so that it can adapt to variations in the payload so that it can perform better than the manually constructed fuzzy controller.**

I. Introduction

FLEXIBLE robotic mechanisms are important in space structure applications, where large, lightweight robots are to be utilized in a variety of tasks, including deployment, spacecraft servicing, space station maintenance, and so on. Flexibility is not designed into the mechanism; it is usually an undesirable characteristic which results from trading off mass and length requirements in optimizing the effectiveness of the robot. In this paper we investigate the development of controllers for a two-link planar flexible robot. Distinguishing features of the robotic mechanism and its operation are the use of structure-mounted sensing only (endpoint accelerations and joint position information) for feedback control, the focus on high speed, gross motion movements in endpoint positioning, and performance requirements for carrying of significant payloads at the robot endpoint.

A. Motivation for Fuzzy Learning Control

For the two-link flexible robot considered here, our goal of achieving fast slews over the entire workspace with a minimum amount of endpoint vibration is complicated by:

1) the manner in which varying the inertial configuration of the links has an effect on structural parameters (e.g., its effects on the modes of vibration), and
2) unknown payload variations which significantly affect the plant dynamics.

Using several years of experience in developing conventional controllers for the robot mechanism, coupled with our intuitive understanding of the dynamics of the robot, we develop a fuzzy controller that achieves adequate performance for a variety of slews. Even though we were able to tune the fuzzy controller to achieve adequate performance for varying configurations; however, its performance degrades when there is a payload variation at the endpoint.

While some would argue that the solution to such a performance degradation problem is to "load more expertise into the rule-base" there are several limitations to such a philosophy including:

1) the difficulties in developing (and characterizing in a rule-base) an accurate intuition about how to best compensate for the unpredictable and significant payload variations that can occur while the robot is in any position in its workspace, and
2) the complexities of constructing a fuzzy controller that potentially has a large number of membership functions and rules.

Moreover, our experience has shown that it is possible to tune fuzzy controllers to perform very well if the payload is known. Hence, the problem does not result from a lack of basic expertise in the rule-base, but from the fact that there is no facility for automatically re-designing (i.e., re-tuning) the fuzzy controller so that it can appropriately react to unforeseen situations as they occur. In this paper, we investigate the possibility of using the "fuzzy model reference learning controller" (FMRLC) [1]–[5] for automatically synthesizing and tuning a fuzzy controller for the flexible robot.

B. Overview and Related Work

While most of the work to date for control of flexible-link robotic systems has used conventional control techniques, there has been recent interest in the literature in the use of intelligent control methodologies. In particular, the need for control theoretic approaches which can incorporate operator knowledge for the process being controlled is being recognized by more and more control engineers who apply control technologies. Since the literature abounds with work on the modeling and control of flexible robots, both from a theoretical (simulation-based) and experimental point of view, we refer

Manuscript received September 21, 1993; revised August 19, 1994. This work was supported in part by the National Science Foundation Grant IRI-9210332.

V. G. Moudgal and W. A. Kwong were with the Department of Electrical Engineering, Ohio State University.

K. Passino and S. Yurkovich are with the Department of Electrical Engineering, Ohio State University, Columbus, OH 43210-1272 USA.

IEEE Log Number 9410130.

the interested reader to [6, ch. 8] for an overview of the literature on conventional approaches. In the following, we focus primarily on recent work relevant to the focus of this paper, and on previous work for the flexible-link robot under study.

One of the most promising techniques for flexible robot control used to date is that of input command shaping, where the system inputs (e.g., motor voltages) are "shaped" in such a manner that minimal energy is injected into the flexible modes of the system. So promising is this technology that a session at the 1993 American Control Conference [7] was devoted to the subject. Indeed, very good results using input shaping with an outer loop disturbance rejection controller for the two-link robot of this study were reported in [8]. Other works employing experimental verifications of input shaping schemes are appearing, such as the session referred to above [7], the ongoing study in [9] for controlling the endpoint movement of a large two-link robot, and the innovations of [10] for an adaptive implementation on a single-link apparatus. It is well known, however, that the primary difficulty of such command-shaping schemes lies in the fact that they are open-loop strategies that require relatively precise knowledge of the system dynamics. Any attempt to improve robustness to uncertainties (such as placing the shaper in the loop, or increasing the filter order) result in delays in the system response, which may or may not be tolerable.

It should be mentioned that recent work in the area of two-time scale (singular perturbation) approaches for vibration suppression in flexible mechanical structures show promise. The control objective in those investigations is different than that of the present study, since in the former, the primary focus is on disturbance rejection effects (small deflections), after larger slew motions are complete; also, inherent in these techniques is the need for accurate models of the system dynamics. Some experimental work utilizing embedded piezoelectrics and piezoceramics has begun to appear. Other recent conventional approaches to the problem of flexible robot control include [11] for the use of linear (state feedback) techniques where a fast state estimator is employed in small angle movements, and [12] in which gross motion movements for a single flexible link are studied in the case of adaptation for payload tasks. As for previous work in developing conventional controllers for flexible robotic test beds at Ohio State (including the two-link apparatus of the current study), the control developed in [13] used a nonlinear inversion (feedback linearization) control law for rigid dynamics, with separate loops for flexure effects; the study in [14] investigated and compared time domain and frequency domain identification techniques on a single-link robot; and in [15] and [16], developed time and frequency domain identification and control schemes for payload adaptation, which were later employed on a two-link apparatus [17].

As noted above, the literature has recently seen an emergence of results using intelligent control technologies. Fuzzy logic, neural networks, and hierarchical schemes have been investigated for flexible robotic mechanisms. For example, a recent paper [18] uses fuzzy logic for a fast-moving single-link apparatus, focusing on smooth rigid body motion control. In [19]–[22] a fuzzy logic supervisory level is used for lower level controller selection and tuning for the same laboratory test bed as is used in the current study. Motivated by the success of those studies, the control scheme of this paper (which is an expanded version of the work reported in [23]) builds on the idea of supervising/tuning lower level controllers in a hierarchy by investigating the possibility of using a higher level learning mechanism to synthesize and tune a rule-based controller at the lower level.

In Section II we provide a description of the two-link flexible robot that we will use in this study and in Section III we explain how intuition and past experiments have provided us with enough information to choose the rule-base and membership functions for a fuzzy controller for the two-link robot. We evaluate the performance of the fuzzy controller for several slews and show that its performance degrades if a payload is added at the endpoint. Overall, while the performance of the fuzzy controller compares favorably to the results we have obtained for this flexible-link robot, the payload variation problem dictates the need for re-tuning the fuzzy controller during robot operation. Furthermore, we emphasize the importance of automatically synthesizing fuzzy controllers for such complex systems.

In Section IV we provide a step-by-step explanation of how to construct a FMRLC [1]–[5] that can synthesize/tune the fuzzy controller for the flexible robot. This involves establishing a structure for the fuzzy controller and choosing a "fuzzy inverse model." Simple modifications to the knowledge-base modification procedure in [1]–[5] were necessary so that the heuristic knowledge that we have about how to best control the robot could be preserved in the rule-base. In particular, we know that if the two links are at their desired locations, the voltage inputs to the motors should be zero. This information is loaded into the rule-base initially and we do not allow the knowledge-base modifier to change this basic fact when it tries to synthesize or tune the direct fuzzy controller. Next, we show that for various slewing angles the FMRLC can automatically synthesize a rule-base for a fuzzy controller that can achieve comparable performance to that obtained via the fuzzy controller of Section III that was constructed manually. Moreover, we show that if the payload is changed, the FMRLC can automatically tune the fuzzy controller so that it will perform better than the one studied in Section III.

Finally, we note that the FMRLC algorithm which was first introduced in [1] and [5] grew from research performed on the linguistic self-organizing controller (SOC) presented in [24] by Procyk and Mamdani and ideas in conventional "model reference adaptive control" (MRAC) [25]. The effectiveness of the FMRLC has been demonstrated via simulations for the inverted pendulum problem, a rocket velocity control problem, a rigid two-link robot, antiskid brakes, fault tolerant aircraft control, and cargo ship steering [1]–[5]. The research results reported here contain a description of the first implementation results for the FMRLC. The linguistic SOC has been used in robotics applications [26] and [27], motor and temperature control [28], blood pressure control [29], and in satellite control [30]–[32]. In terms of comparing FMRLC and SOC, the authors in [1] and [5] have shown that the FMRLC has

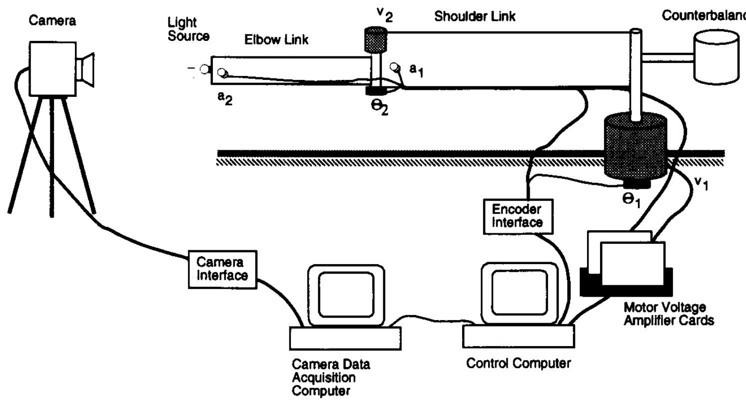


Fig. 1. Two-link flexible robot setup.

several advantages over SOC including improved performance feedback and lack of dependence on a mathematical model of the inverse of the plant. Other relevant literature that focuses on adaptation of a direct fuzzy controller includes the work in [33] where an adaptive fuzzy system is developed for a continuous casting plant, the approach in [34] where a fuzzy system adapts itself to driver characteristics for an automotive speed control device, and the approaches in [35]–[38].

II. Laboratory Test Bed

The two-link flexible robot shown in Fig. 1 consists of three principle parts: the robot with its sensors, the computer and the interface to the robot, and the camera with its computer and interface. The robot is made up of two very flexible links constrained to operate in the horizontal plane. The "shoulder link" is a counter-balanced aluminum strip 75 cm long, 12.7 cm tall and 0.23 cm thick and is driven by a DC direct drive motor with a stall torque of 4.802 N-m. The "elbow link" mounted on the shoulder link endpoint is an aluminum strip 50 cm long, 3.8 cm tall and 0.1 cm thick. The actuator for the elbow link is a 28 volt DC, geared motor $(30:1)$ with a stall torque of 2.53×10^{-3} N-m. The sensors on the robot are two optical encoders for the motor shaft positions Θ_1 and Θ_2, and two accelerometers mounted on the link endpoints to measure the accelerations a_1 and a_2. The inputs to the robot are the two voltage signals v_1 and v_2 at the motor terminals.

A Reticon LC-310 line scan camera interfaced to an IBM PC XT is used to monitor the endpoint position of the robot for plotting; this data is not used for feedback. For comparative purposes, in this paper we use the camera data for robot movements which end in a fully extended position and which begin in some position to approximate equal movements in each joint. When responses are plotted, the final endpoint position is nominally indicated (on the plot) to reflect (approximately) the total movement, in degrees, of the shoulder joint. Because movements are constrained to the horizontal plane, there are no gravity effects on the motors, and therefore it is appropriate to express performance (set points) in terms of joint angles. We note that constraining the robot to operate in the horizontal plane is done precisely for these reasons (to remove gravity effects), since the primary application for this work is large, lightweight robots in space.

The control computer for the robot is a PC with an Intel 80386SX operating at 25 mHz. The computer interface hardware used by the control computer is a Keithley MetraByte DAS1600 and a Scientific Solutions Lab Tender card. The camera computer uses a Scientific Solutions Lab Tender card. The camera interface and the encoder interface are additional circuits designed and built in house [39] for signal conditioning.

The primary objective of this research is to develop a controller that makes the robot move to its desired position as quickly as possible, with little or no endpoint oscillation. To appreciate the improvement in the plant behavior due to the application of the various control strategies we will first look at how the robot operates under the "no control" situation; that is, when no external digital control algorithm is applied for vibration compensation. To implement the no control case we simply apply $v_1 = v_2 = 0.3615$ volts at $t = 0$ seconds and return v_1 and v_2 to zero voltages as soon as the links reach their setpoints. Note that for this experiment we monitor the movement of the links but do not use this information as feedback for control.

The result of the "no control" experiment is shown in Fig. 2 where the endpoint position is shown. The response shows a significant amount of endpoint oscillation and steady state error. Here, as in all plots to follow, endpoint position refers to the position of the elbow endpoint. In the ideal case, the shaft should stop moving the instant the voltage signal to the motor amplifier is cut off. But the arm had been moving at a constant velocity before the signal was cut off, and thus has a momentum which will drag the shaft past the point it was to stop. This movement depends on the speed at which the arm was moving, which in turn depends on the voltage signal applied. Clearly there is a significant need for vibration damping in endpoint positioning.

III. Direct Fuzzy Control

We begin our experiments with an investigation into the performance of the direct fuzzy controller shown in Fig. 3.

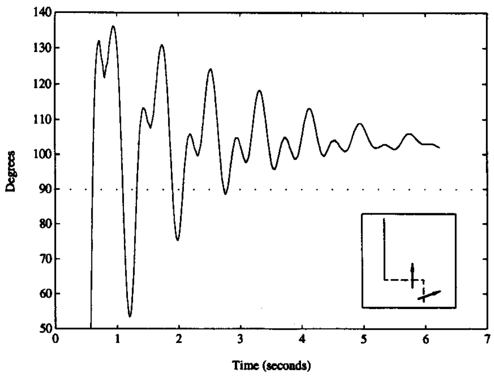


Fig. 2. "No control" response.

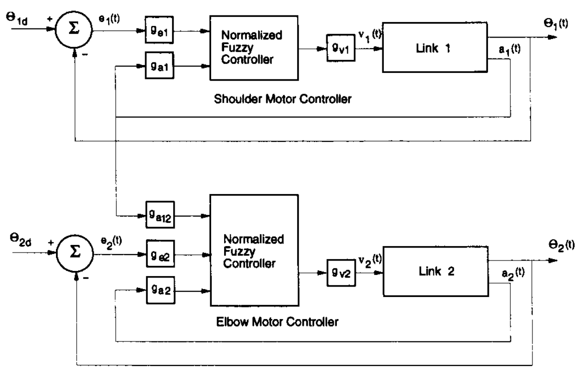


Fig. 3. Fuzzy control system design for direct fuzzy controller design.

Let θ_{1d} and θ_{2d} be the desired positions of the shoulder and the elbow links, respectively, (i.e., the commanded slew). We use one fuzzy controller for each link with position error $e_i(t) = \theta_{id}(t) - \theta_i(t)$ and acceleration inputs $a_i(t), i = 1, 2$ to each.[1] In addition, since the acceleration of the shoulder link can significantly affect the behavior of the elbow link (but not vice versa) we use the acceleration $a_1(t)$ of the shoulder link endpoint as an input to the elbow-link controller. This allows the elbow-link controller to compensate for the coupling effects from the shoulder link and hence reduce endpoint vibrations in the elbow link.

The input and the output universes of discourse of the fuzzy controllers are normalized on the range $[-1, 1]$. The gains $g_{e1}, g_{e2}, g_{a12}, g_{a1}$, and g_{a2} are used to map the actual inputs of the fuzzy system to the normalized universe of discourse and are called "normalizing gains." Similarly g_{v1} and g_{v2} are the output gains to scale the output of the controllers. We use singleton fuzzification, center of gravity defuzzification, and the min operator to implement the premise and implication throughout this paper [40].

[1]We experimented with using the change in position error of each link as an input to each of the link controllers, but found that it significantly increased the complexity of the controllers with very little if any improvement in overall performance; hence, we did not pursue the use of this controller input. Typically, we use filtered signals from the accelerometers, prior to processing, to enhance their effectiveness.

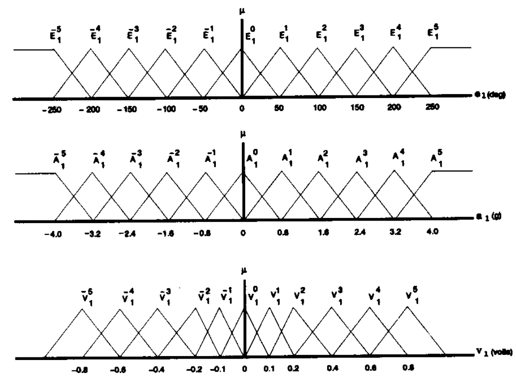


Fig. 4. Membership functions for the shoulder controller.

A. Rule-Base

The shoulder controller uses triangular membership functions as shown in Fig. 4. Notice that the membership functions for the input fuzzy sets are uniform, but the membership functions for the output fuzzy sets are narrower near zero. Experience has shown that this serves to decrease the "gain" of the controller near the setpoint so we can obtain good steady state control and yet avoid excessive overshoot. For the shoulder controller the universe of discourse for the position error is chosen to be [−250, +250] degrees.[2] The universe of discourse for the endpoint acceleration of the shoulder link is [−4, +4] g. This width of 8 g was picked after experimentation with different slews at different speeds, upon observing the output of the acceleration sensor. The output universe of discourse of −0.8, +0.8 volts was chosen so as to keep the shaft speed within reasonable limits.

The rule-base array that we use for the shoulder controller is shown in Fig. 5. The rule-base is an 11×11 array, as we have 11 fuzzy sets on the input universes of discourse. The top most row shows the indices for the eleven fuzzy sets for the acceleration input a_1 and the column at extreme left shows the indices for the eleven fuzzy sets for the position error input e_1. The body of the array shows the indices m for V_1^m in fuzzy implications of the form

$$\text{If } E_1^j \text{ and } A_1^k \text{ Then } V_1^m$$

where E_i^j, A_i^j, and V_i^j; $(i = 1, 2; -5 \le j \le +5)$ denote the jth fuzzy sets associated with e_i, a_i, and v_i, respectively. Notice the uniformity of the indices in Fig. 5 and that for the row $j = 0$ there are three zeros in the center. These zeros have been placed so as to reduce the sensitivity of the controller to the accelerometer signal which is somewhat noisy.

The membership functions for the elbow controller are shown in Fig. 6. The universe of discourse for the position

[2]Note that in this paper we will refer to $[X, Y]$ as being the universe of discourse while in actuality the universe of discourse is made up of all reals (e.g., in Fig. 4 we will refer to the universe of discourse of $e_1(t)$ as [−250, +250]). In addition, will refer to $Y - X$ as being the "width" of the universe of discourse (so that the width of the universe of discourse [−250, +250] is 500). Moreover, note that by specifying the width for the universes of discourse, we are also specifying the corresponding scale factor. For example, if the input universe of discourse for $e_1(t)$ is [−250, +250] then $g_{e1} = \frac{1}{250}$, and if the output universe of discourse for $v_1(t)$ is [−0.8, +0.8] the $g_{v1} = 0.8$.

V_1^m		A_1^k										
		−5	−4	−3	−2	−1	0	+1	+2	+3	+4	+5
E_1^j	−5	−5	−5	−5	−4	−4	−3	−3	−2	−2	−1	0
	−4	−5	−5	−4	−4	−3	−3	−2	−2	−1	0	+1
	−3	−5	−4	−4	−3	−3	−2	−2	−1	0	+1	+2
	−2	−4	−4	−3	−3	−2	−2	−1	0	+1	+2	+2
	−1	−4	−3	−3	−2	−2	−1	0	+1	+2	+2	+3
	0	−4	−3	−2	−1	0	0	0	+1	+2	+3	+4
	+1	−3	−2	−2	−1	0	+1	+2	+2	+3	+3	+4
	+2	−2	−2	−1	0	+1	+2	+2	+3	+3	+4	+4
	+3	−2	−1	0	+1	+2	+2	+3	+3	+4	+4	+5
	+4	−1	0	+1	+2	+2	+3	+3	+4	+4	+5	+5
	+5	0	+1	+2	+2	+3	+3	+4	+4	+5	+5	+5

Fig. 5. Rule-base for the shoulder link.

error is [−250, +250] degrees, and for the elbow link endpoint acceleration is [−8, +8] g. The universe of discourse for the shoulder link acceleration is [+2, −2] g. This small range was chosen to make the elbow-link controller sensitive to small changes in the shoulder link endpoint oscillation. The universe of discourse for the output voltage is [−4, +4] volts. Fig. 7(a)–(g) depicts a three dimensional rule-base. Fig. 7(d) represents the case when the acceleration input from the shoulder link is zero and is the center of the rule-base (the body of the table denotes the indices m for V_2^m). Fig. 7(a)–(c) are for the case when the shoulder endpoint acceleration is negative and Fig. 7(e)–(g) are for the case where the shoulder endpoint acceleration is positive. The central portion of the rule base makes use of the entire output universe of discourse. This is the portion of the rule base where the acceleration input from the shoulder link endpoint is zero or small. As we move away from the center of the rule base (to the region where the shoulder link endpoint acceleration is large), only a small portion of the output universe of discourse is used to keep the output of the controller small. Thus the speed of the elbow link is dependent on the acceleration input from the shoulder link endpoint. The speed of the elbow link is decreased if the acceleration is large and is increased as the acceleration input decreases. Also note in Fig. 7(c)–(e) that there are three zeros in the middle rows to reduce the sensitivity of the controller to the noisy accelerometer signal. This noise is not a significant problem when the endpoint is oscillating and so the rule-base does not have the zeros in the outer region. Taking the rule-base as a three dimensional array we get a central cubical core made up of zeros. Also notice that some parts of the rule-base, especially toward the extremes of the third dimension, are not fully uniform. This has been done to slow down the elbow link when the acceleration input from the shoulder link is very large.

The direct fuzzy controller seeks to vary the speed of the elbow link depending on the amplitude of oscillations in the shoulder link. If the shoulder link is oscillating too much, the speed of the elbow link is reduced so as to allow the oscillations in the shoulder link to be damped, and if there are no oscillations in the shoulder link then the second link speed is increased. We do this to eliminate the oscillation of the elbow link close to the set point where the control voltage from the elbow controller is small. The number of

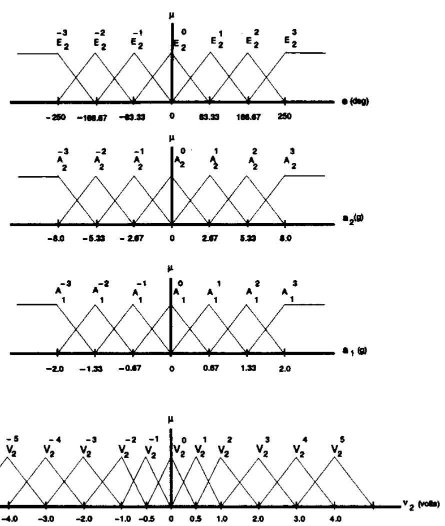


Fig. 6. Membership functions for the elbow controller.

rules used for the direct fuzzy controller is 121 for shoulder controller, plus 343 for elbow-link controller, for a total of 464 rules. Experiments showed that use of fewer rules resulted in degraded performance. We used a sampling period $T = 15$ ms.

Obviously, much effort and experience has gone into the construction of the rule base for this fuzzy controller; hence, the ability to automatically synthesize this rule base would be a definite improvement. Results below will show that good performance is achieved for no payloads, and presumably equally good performance is possible if the controller could be tuned for varying payloads.

B. Results

The experimental results obtained using direct fuzzy control are shown in Fig. 8. The slew requested here is shown by the inset (90 degrees for each link). Note that there is no overshoot in the response, with negligible residual vibrations. The dip in the curve in the initial part of the graph is due to the first link "braking" as it reaches the set point, primarily because of the deadzone nonlinearity in the gears. As the shoulder link brakes, the elbow link is accelerated due to its inertia. The elbow link, which was at one end of its deadzone while the shoulder was moving, shoots to the other end of the deadzone causing the local maxima seen in Fig. 8(a) at around 0.9 seconds. The link recoils due to its flexibility and starts moving to the lower end of the deadzone. By this time the elbow motor speed increases and prevents further oscillation of the elbow link in the deadzone.

Fig. 9 shows the response of the robot to a counter-relative slew (i.e., links moving in opposite directions). The requested slews were 90 degrees for each link as shown in the inset. The response shows similar performance to that obtained for the previous slew even though it is known that performance for counter-relative slews can degrade. The initial hump seen in the plot at 0.5 seconds is due to the nature of the commanded slew. As seen from the inset, the shoulder link is commanded

A_1^{-3}: E_2^j \ V_2^m	A_2^k: −3	−2	−1	0	+1	+2	+3
−3	−3	−3	−2	−2	−1	−1	0
−2	−3	−2	−2	−1	−1	0	+1
−1	−2	−2	−1	−1	0	+1	+1
0	−2	−1	−1	0	+1	+1	+2
+1	−1	−1	0	+1	+1	+2	+2
+2	−1	0	+1	+1	+1	+2	+2
+3	0	+1	+1	+1	+2	+2	+2

(a)

A_1^{-2}: E_2^j \ V_2^m	A_2^k: −3	−2	−1	0	+1	+2	+3
−3	−3	−3	−3	−2	−2	−1	0
−2	−3	−3	−2	−1	−1	0	+1
−1	−3	−2	−2	−1	0	+1	+1
0	−2	−2	−1	0	+1	+2	+2
+1	−2	−1	0	+1	+1	+2	+2
+2	−1	0	+1	+1	+2	+2	+2
+3	0	+1	+1	+2	+2	+2	+3

(b)

A_1^{-1}: E_2^j \ V_2^m	A_2^k: −3	−2	−1	0	+1	+2	+3
−3	−4	−4	−3	−3	−2	−1	0
−2	−4	−3	−3	−2	−1	0	+1
−1	−3	−3	−2	−1	0	+1	+1
0	−2	−2	0	0	0	+1	+2
+1	−2	−1	0	+1	+2	+2	+3
+2	−1	0	+1	+2	+2	+3	+3
+3	0	+1	+2	+2	+3	+3	+3

(c)

A_1^{0}: E_2^j \ V_2^m	A_2^k: −3	−2	−1	0	+1	+2	+3
−3	−5	−4	−4	−3	−3	−2	0
−2	−4	−4	−3	−2	−1	0	+1
−1	−4	−3	−2	−1	0	+1	+2
0	−2	−1	0	0	0	+1	+2
+1	−2	−1	0	+1	+2	+3	+4
+2	−1	0	+2	+2	+3	+4	+4
+3	0	+1	+2	+3	+4	+4	+5

(d)

A_1^{+1}: E_2^j \ V_2^m	A_2^k: −3	−2	−1	0	+1	+2	+3
−3	−4	−3	−3	−2	−2	−1	0
−2	−3	−3	−2	−2	−1	0	+1
−1	−3	−2	−2	−1	0	+1	+2
0	−2	−1	0	0	0	+1	+2
+1	−2	−1	0	+1	+2	+3	+3
+2	−1	0	+1	+2	+3	+3	+4
+3	0	+1	+2	+3	+3	+4	+4

(e)

A_1^{+2}: E_2^j \ V_2^m	A_2^k: −3	−2	−1	0	+1	+2	+3
−3	−3	−2	−2	−1	−1	−1	0
−2	−3	−2	−2	−1	−1	0	+1
−1	−2	−2	−1	−1	0	+1	+2
0	−2	−1	−1	0	+1	+1	+2
+1	−1	−1	0	+1	+1	+2	+3
+2	−1	0	+1	+1	+2	+3	+3
+3	0	+1	+2	+2	+3	+3	+4

(f)

A_1^{+3}: E_2^j \ V_2^m	A_2^k: −3	−2	−1	0	+1	+2	+3
−3	−2	−2	−2	−1	−1	−1	0
−2	−2	−2	−1	−1	−1	0	+1
−1	−2	−2	−1	−1	0	+1	+2
0	−2	−1	−1	0	+1	+1	+2
+1	−1	−1	0	+1	+1	+2	+2
+2	−1	0	+1	+1	+2	+2	+3
+3	0	+1	+1	+2	+2	+3	+3

(g)

Fig. 7. Rule-base array for the elbow link.

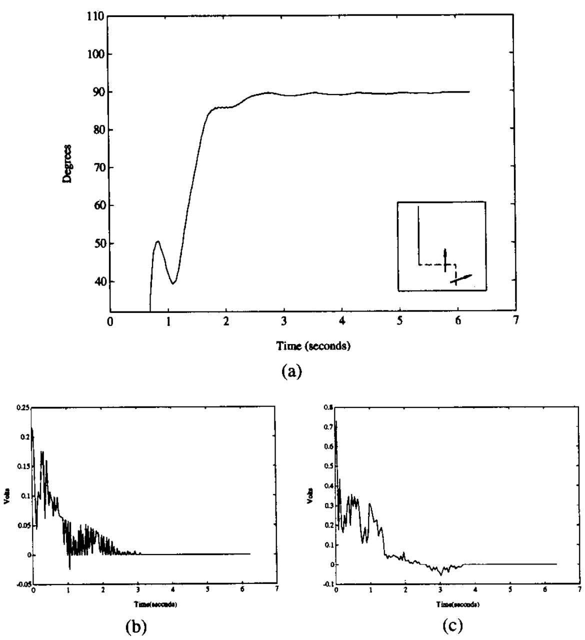


Fig. 8. (a) Plant response for direct fuzzy controller design. (b) Shoulder link input. (c) Elbow link input.

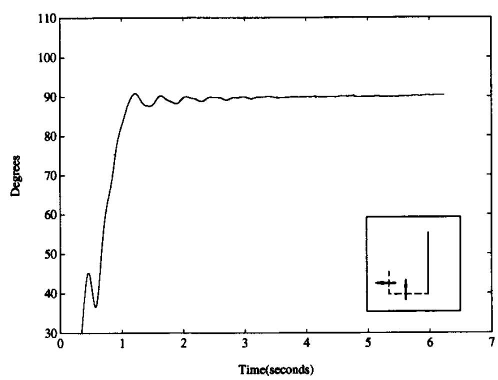


Fig. 9. Endpoint position for counter-relative slews using direct fuzzy controller.

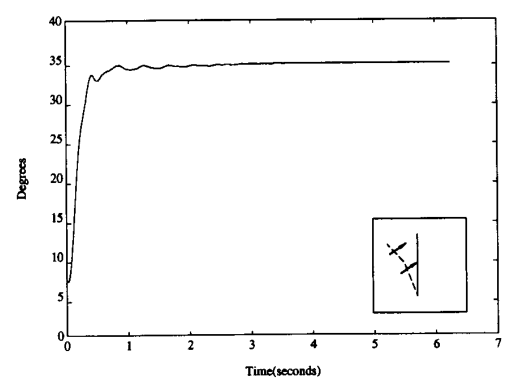


Fig. 10. Endpoint position for small slews using direct fuzzy controller.

to move clockwise and the elbow link is commanded to move counter-clockwise. The camera is placed so that when both the links have completed their slews, the tip of the elbow link endpoint is pointed directly at the camera. The shoulder link moves so as to bring the endpoint into the visual range of the camera, but at the same time the elbow link is moving in the opposite direction. If the speed of the elbow link is greater than the speed of the shoulder link at that point it appears as a hump in the data collected by the camera. Fig. 10 shows the response of the robot to a small slew. The commanded slew is 20 degrees for both the links and is shown in the inset. We see that as expected we get even higher performance for smaller angles.

Fig. 11 shows the endpoint response of the robot with a 30 gram payload (assumed unknown) attached to its endpoint. The commanded slew is 90 degrees on each link as shown in the inset. Notice that the dip in the curve (between 1.0 to 1.5 sec) is reduced as compared to the case without payload. This is due to the increased inertia of the elbow link, which reduces the frequency of oscillation of the link and the elbow link motor speeds up at this point preventing further oscillations. Note,

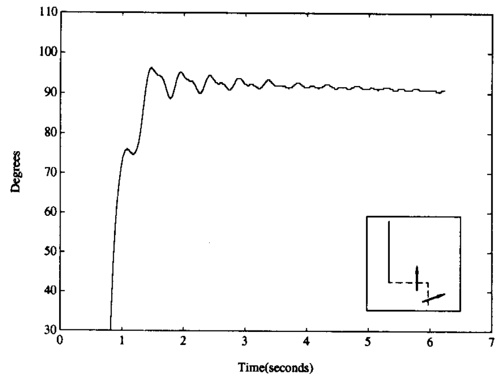


Fig. 11. Endpoint position for fuzzy controller with payload.

however, that the amplitude and duration of the vibrations is significantly degraded compared to the case where there is no payload (see Fig. 2); we see that there is a need to tune the fuzzy controller so that it can compensate for the effects of the unknown payload variation. This is investigated in the next section.

IV. Fuzzy Learning Control

While the fuzzy control approach in Section III offers adequate performance for the robot it still has several disadvantages including: i) the problems one encounters in trying to manually synthesize the rule-base, and ii) the possible performance degradations that can occur due to unpredictable and unknown plant parameter variations or disturbances (e.g., due to payload variations). In this section we will study the use of the FMRLC [1]–[5] for automatically synthesizing and tuning the rule-base of a direct fuzzy controller to alleviate these two problems.

A. Fuzzy Model Reference Learning Control

The FMRLC shown in Fig. 12 utilizes a learning mechanism that observes data from a fuzzy control system, characterizes its current performance, and automatically adjusts the knowledge base of the fuzzy controller so that the closed-loop system performs according to the specifications given by the reference model. Next, we will describe each component of the FMRLC for the two-link robot.

1) The Fuzzy Controller: We used the same basic structure for the fuzzy controller as was used in Section III with the same input fuzzy sets as shown in Figs. 4 and 6, but the difference here is that the output fuzzy sets for both controllers are all initially centered at zero resulting in rule-bases filled with zeros. This implies that the fuzzy controller by itself has no knowledge about how to control the plant. As the algorithm executes, the output fuzzy sets are rearranged by the learning mechanism, filling up the rule-base. For instance, once a slew is commanded the learning mechanism described below will move the centers of the activated rules away from zero and begin to synthesize the fuzzy controller.

The universe of discourse for the position error input e_1 to the shoulder link controller was chosen to be [-100, +100] degrees, and the universe of discourse for the endpoint acceleration a_1 is [-10, +10] g. For the elbow-link controller the universe of discourse for the position error e_2 is [-80, +80] degrees and the universe of discourse for the acceleration input a_2 is [-10, +10] g. The universe of discourse for the shoulder link acceleration input a_{12} to the elbow-link controller is [-8, +8] g. We choose the output universe of discourse for v_1 and v_2 by letting $g_{v1} = 0.125$ and $g_{v2} = 1.0$. We determined all these values from our experiences in experimenting with the fuzzy controller in Section III and in our experiments with the FMRLC.

2) The Reference Model: The reference model is a model of how we would like the closed-loop system to behave. The reference model may be any type of dynamical system either linear or nonlinear. It is used to characterize closed-loop specifications such as rise-time, overshoot, and settling time. The performance of the overall system is computed with respect to the reference model by generating error signals between the reference model output and the plant outputs (i.e., y_{e1} and y_{e2}; see Fig. 12). To achieve the desired performance the learning mechanism must force $y_{e1}(kT) \approx 0$ and $y_{e2}(kT) \approx 0$ for all $k \geq 0$. It is important to make a proper choice for a reference model so that the desired response does not dictate unreasonable performance requirements for the plant to be controlled. Through experimentation we determined that $(3/(s+3))$ is a good choice for the reference models for both the shoulder and the elbow links.

3) The Learning Mechanism: The learning mechanism performs the function of modifying the knowledge base of the fuzzy controller so that the closed-loop system behaves like the reference model. The learning mechanism essentially consists of two parts: a "fuzzy inverse model" and a "knowledge-base modifier" as we discuss next.

a) The Fuzzy Inverse Model: The fuzzy inverse model makes an assessment of the deviation of the current closed-loop system behavior from the behavior specified by the reference model (the desired closed-loop system behavior) and decides how to change the plant command inputs (controller outputs) so that this deviation goes to zero.[3] Successful fuzzy inverse model designs have been completed for several applications including a cart-pendulum system [1], cargo ship steering [2], and antiskid brakes [3]. In the cart-pendulum application the fuzzy inverse model indicates how to change the force being applied to the cart so that the pendulum will balance whether the pendulum is in the upright or downward position. In the cargo ship steering application, the fuzzy inverse model simply indicates how to change the way the rudder input is being generated so that the ship heading tracks the desired heading. In the antiskid brakes application the fuzzy inverse model characterizes the knowledge we have about how to change the way braking torque is being applied so that the slip can be regulated to an optimum point. Hence, knowledge similar to what is used for standard direct fuzzy control design is used to construct the fuzzy inverse model.

[3] By providing an association between plant output deviations and changes in plant inputs it models an "inverse behavior" of a dynamical system; hence, as it is explained in more detail in [1]–[3], we use the term "fuzzy inverse model."

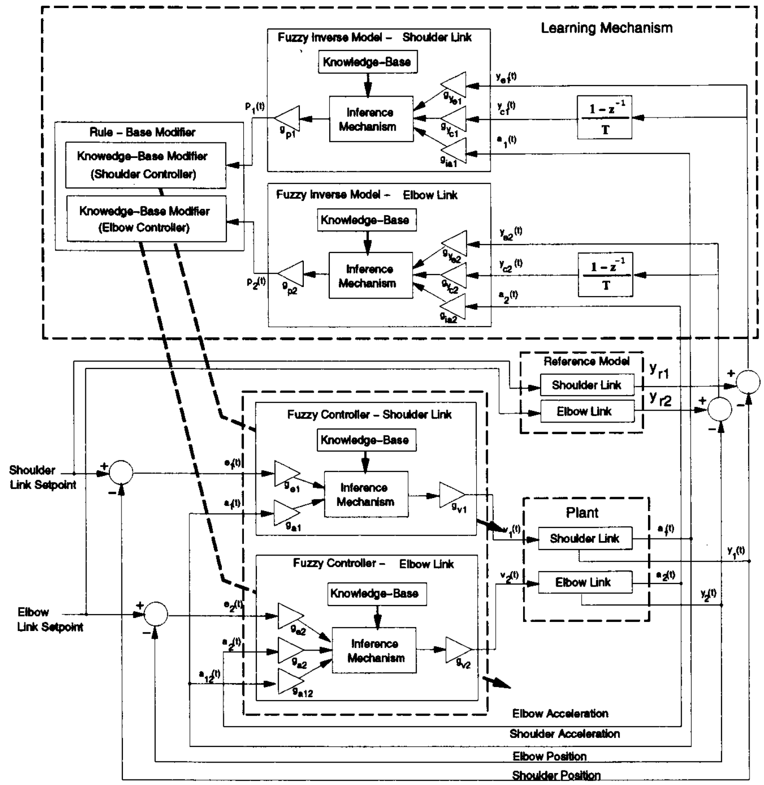


Fig. 12. Fuzzy model reference learning control.

From this perspective, the fuzzy inverse model acts as a controller in the adaptation loop to try to reduce the deviation in the closed-loop system behavior from the desired behavior by changing the underlying fuzzy controller. Essentially, the fuzzy inverse model constructs (via the knowledge-base modifier described below) the fuzzy controller using on-line data from the closed-loop system so that the behavior specified in the reference model is achieved. Having knowledge about how to specify the fuzzy inverse model does not imply that we know how to specify a fixed (nonadaptive) direct fuzzy controller that can perform at similar levels since i) the fuzzy inverse model initially synthesizes the direct fuzzy controller by also using information gathered during on-line operation, and ii) subsequent tuning of the fuzzy controller can occur by using on-line information about plant behavior changes. Moreover, whether in an initial synthesis stage or tuning phase (which blend together for real applications), the fuzzy inverse model acts to construct the fuzzy controller so that the specifications modeled with the reference model are achieved (often, off-line design of direct fuzzy controllers requires many iterations to achieve a specified behavior—the FMRLC folds such iterations into its on-line operation). Hence, the FMRLC uses similar *a priori* knowledge to that used in the off-line design of conventional fixed direct fuzzy controllers, coupled with performance information that is gathered on-line, to decide how to synthesize/adjust the direct fuzzy controller. Clearly on-line design can sometimes have advantages over off-line design if information about the on-line performance is used appropriately.

In summary, it is the goal of the design of the fuzzy inverse model to capture the best way to incorporate i) the *a priori* knowledge that we have about how to control the plant, and ii) the on-line performance information that is gathered while the closed-loop system operates. It is important to note that as with conventional fuzzy control, ultimately the design of the fuzzy inverse model relies on heuristic expertise that we have about how to best control the plant. Experience with the FMRLC [1]–[3] has shown that by using such heuristic expertise (which at times is not completely accurate), the fuzzy inverse model can achieve very efficient and high performance adaptation for the FMRLC approach. Next, we explain how we designed the fuzzy inverse model for the two-link flexible robot.

For our robot there are two fuzzy inverse models, each with three inputs $y_{ej}(t), y_{cj}(t)$, and $a_j(t)$ ($j = 1$ corresponding to the shoulder link and $j = 2$ to the elbow link as shown in Fig. 12). Several issues dictated the choice of these inputs: i) we found it easy to specify reference models for the shoulder and elbow link position trajectories (as it was discussed above) and hence the position error signal is readily available, ii) we found via experimentation that the rates of change of position errors, $y_{cj}(t), j = 1, 2$, and acceleration signals $a_j(t), j = 1, 2$ were very useful in deciding how to adjust the fuzzy controller, and (iii) we sought to minimize the number of inputs to the fuzzy inverse models to ensure that we could implement the FMRLC with a short enough sampling interval (in our case, 15 ms). The direct use of the acceleration signals $a_j(t), j = 1, 2$, for the inverse models actually corresponds to choosing reference models for the acceleration signals that say "no matter what slew is commanded, the desired accelerations of the links should be zero." While it is clear that the links cannot move without accelerating, with this choice the FMRLC with attempt to accelerate the links as little as possible to achieve the command slews, thereby minimizing the amount of energy injected into the modes of vibration (investigations into the use of other reference models for the acceleration signals is an important future direction). Next, we discuss rule-base design for the fuzzy inverse models.

For the rule-bases of the fuzzy inverse models we use rules similar to those described in Fig. 7 for both the shoulder and elbow links except that the cubical block of zeros is eliminated by making the pattern of consequents uniform. These rules have premises that quantify the position error, the rate of change of the position error, and the amount of acceleration in the link. The consequents of the rules represent the amount of change that should be made to the direct fuzzy controller by the knowledge-base modifier. For example, fuzzy inverse model rules capture knowledge such as: i) if the position error is large and the acceleration is moderate, but the link is moving in the correct direction to reduce this error, then a smaller change (or no change) is made to the direct fuzzy controller than if the link is moving to increase the position error; and ii) if the position error is small but there is a large change in position error and a large acceleration, then the fuzzy controller must be adjusted to avoid overshoot. Similar interpretations can be made for the remaining portions of the rule-bases used for both the shoulder and elbow link fuzzy inverse models.

The membership functions for both the shoulder and elbow link fuzzy inverse models are similar to those used for the elbow-link controller shown in Fig. 6 except that the membership functions on the output universe of discourse are uniformly distributed and there are different widths for the universes of discourse as we explain next (these widths define the gains $g_{y_{ej}}, g_{y_{cj}}, g_{iaj}$, and g_{pj} for $j = 1, 2$). We choose the universe of discourse for y_{ei} to be [-80, +80] degrees for the shoulder link and [-50, +50] for the elbow link. We have chosen a larger universe of discourse for the shoulder link inverse model than the elbow link inverse model because we need to keep the change of speed of the shoulder link gradual so as not to induce oscillations in the elbow link (the elbow link is mounted on the shoulder link and is affected by the oscillations in the shoulder link). The universe of discourse for y_{c1} is chosen to be [-400, +400] degrees/sec for the shoulder link and [-150, +150] degrees/sec for y_{c2} of the elbow link. These universes of discourse were picked after experimental determination of the angular velocities of the links. The output universe of discourse for the fuzzy inverse model outputs (p_1 and p_2) is chosen to be relatively small to keep the size of the changes to the fuzzy controller small which helps ensure smooth movement of the robot links. In particular, we choose the output universe of discourse to be [-0.125, +0.125] for the shoulder link inverse model, and [-0.05, +0.05] for the elbow link inverse model. Choosing the output universe of discourse for the inverse models to be [-1, +1] causes the learning mechanism to continually make the changes in the rule-base of the controller so that the actual output is exactly equal to the reference model output, making the actual plant follow the reference model closely. This will cause significant amounts of speed variations in the motors as they try to track the reference models exactly, resulting in chattering along a reference model path. The choice of a smaller width for the universe of discourse keeps the actual output below the output of the reference model until it reaches the setpoint. This increases the settling time slightly but the response is much less oscillatory. This completes the definition of two fuzzy inverse models in Fig. 12.

b) The Knowledge-Base Modifier: The knowledge-base modifier performs the function of modifying the fuzzy controller so that better performance is achieved. Given the information (from the inverse models) about the necessary changes in the input needed to make $y_{e1} \approx 0$ and $y_{e2} \approx 0$, the knowledge-base modifier changes the knowledge-base of the fuzzy controller so that the previously applied control action will be modified by the amount specified by the inverse model outputs $p_i, i = 1, 2$. To modify the knowledge-base, the knowledge-base modifier shifts the centers of the rules (initialized at zero) that were "on" during the previous control action by the amount $p_1(t)$ for the shoulder controller and $p_2(t)$ for the elbow controller. Suppose we have some nonzero error $y_{ei}(t), i = 1, 2$ between the reference model and the actual plant output. This will normally produce some finite nonzero fuzzy inverse model output $p_i(t), i = 1, 2$. The knowledge-base modification procedure consists of two steps: i) determine the rules that are "on," i.e., the rules that produced the previous control action that produced the error $y_{ei}(t), i = 1, 2$ and, ii) modify the entries in the knowledge-base array for those rules by the amount $p_i(t), i = 1, 2$. Via such knowledge-base modification the learning mechanism is able to force the fuzzy controller to produce the desired output given the similar controller inputs. For more details, including a mathematical description of the knowledge-base modifier, see [1]–[3].

Note that via experimentation we found that certain enhancements to the FMRLC knowledge-base modification procedure were needed. In particular, based on the physics of the flexible robot, we know that if the errors e_1 and e_2 are near zero, the fuzzy controller should choose $v_1 = v_2 = 0.0$. Hence, using this knowledge about how to control the plant, we use the same FMRLC knowledge-base modification procedure as in [1]–[3] except that we never modify the rules at the center of the rule-base so that the fuzzy controller will always output zero when there is zero error. Essentially, we make this adjustment to the knowledge-base modification procedure to overcome a high gain effect near zero that we observed in the previous experiments.

Finally, we note that the total number of rules used by the FMRLC is 121 for the shoulder controller, plus 343 for the elbow controller, plus 343 for the shoulder fuzzy inverse model, plus 343 for the elbow fuzzy inverse model, for a total

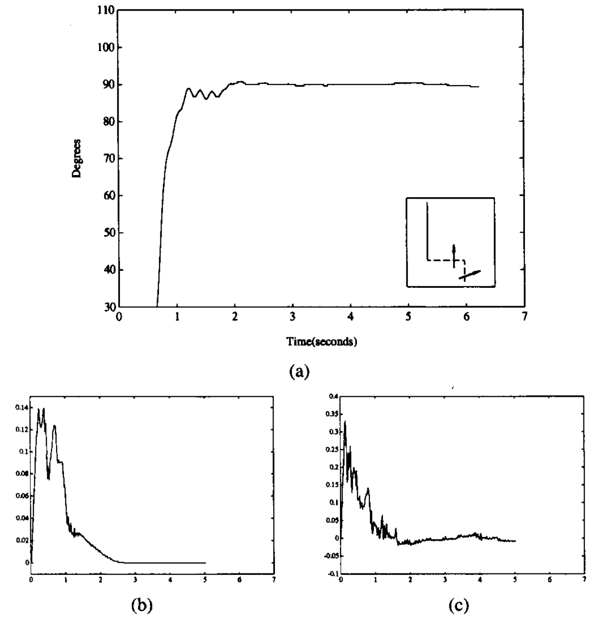


Fig. 13. (a) Endpoint position for FMRLC design. (b) Shoulder link input voltage v_1. (c) Elbow link input voltage v_2.

of 1150 rules. Even with this number of rules we were able to keep the same sampling time of $T = 15$ ms that was used for the direct fuzzy controller in Section III.

B. Results

Experimental results obtained from the use of FMRLC are shown in Fig. 13 for a slew of 90 degrees for each link. The rise time for the response is about 1.0 second and the settling time is approximately 1.8 seconds. Comparing this response to the direct fuzzy control response (Fig. 8) we see an improvement in the endpoint oscillation and the settling time. Note that the settling time for the robot is slightly larger than that of the reference model (1.5 seconds). This is because of the way the learning mechanism modified the rule-base of the controller to keep the response below that of the reference model. Fig. 14 shows the response of the controller for a counter-relative slew. The commanded slew is 90 degrees for each link, and has a geometry as shown in the inset. The local maxima appearing in the plot at 0.7 seconds is due to the geometry of the slew as was explained in Section III. The results are comparable to the direct fuzzy control (see Fig. 9). Fig. 15 shows the response for small angle slew of 20 degrees for each link. The response is comparable but has slightly more oscillations compared to the responses obtained from the direct fuzzy control algorithms in Fig. 10. This is expected due to the fact that the controller starts off with no knowledge in its knowledge base and learns as it executes the algorithm. In the case of small angles, it does not get enough samples to learn the dynamics of the plant completely, resulting in slightly more oscillations.

Fig. 16 shows the robot response for the loaded endpoint case. The elbow link endpoint is loaded with a 30 gram

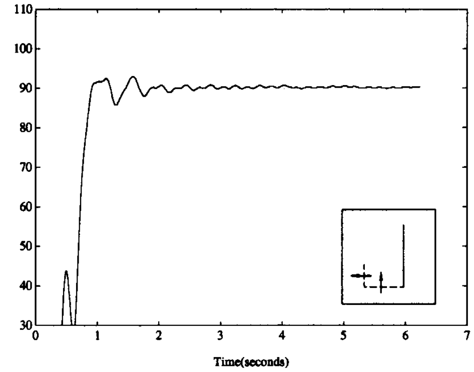


Fig. 14. Endpoint position for counter-relative slew for FMRLC.

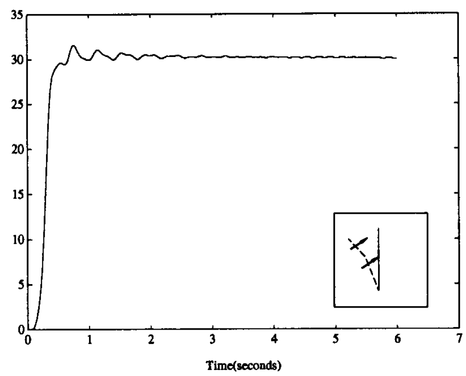


Fig. 15. Endpoint position for small angle slew for FMRLC.

mass of aluminum and is commanded to slew 90 degrees at each link. The response with the payload is superior to the direct fuzzy controller (see Fig. 11). To achieve the improved performance shown in Fig. 16 the FMRLC exploits i) the information that we have about how to control the flexible robot that is represented in the fuzzy inverse model (see discussion in Subsection A above) and ii) data gathered during the slewing operation as we discuss next. During the slew, the FMRLC observes how well the fuzzy controller is performing (using data from the reference model and robot) and seeks to adjust it so that the performance specified in the reference model is achieved and vibrations are reduced. For instance, in the initial part of the slew the position errors are large, the change in errors are zero, the accelerations are zero, and the fuzzy controller has all its consequent membership functions centered at zero. For this case, the fuzzy inverse model will indicate that the fuzzy controller should generate voltage inputs to the robot links that will get them moving in the right direction. As the position errors begin to change and the change in errors and accelerations vary from zero, the fuzzy inverse model will cause the knowledge-base modifier to fill in appropriate changes to the fuzzy controller consequent membership functions until the position trajectories match the ones specified by the reference models (note that the fuzzy inverse model was designed so that it will continually adjust the fuzzy controller until the

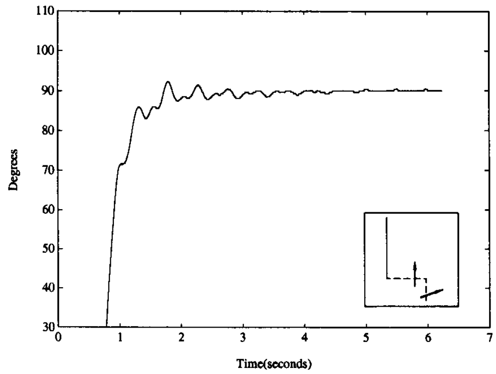


Fig. 16. Endpoint position for loaded elbow link for FMRLC.

reference model behavior is achieved). Near the end of the slew (i.e., when the links are near their commanded positions) the FMRLC is particularly good at vibration damping since in this case the plant behavior will repeatedly return the system to the portion of the fuzzy controller rule-base that was learned the last time a similar oscillation occurred (i.e., the learning capabilities of the FMRLC enable it to develop, remember, and re-apply a learned response to plant behaviors). Different payloads change the modal frequencies in the link/payload combination (e.g., heavier loads tend to reduce the frequencies of the modes of oscillation) and the shapes of the error and acceleration signals $e_1(t), e_2(t)$, and $a_1(t)$ (e.g., heavier loads tend to slow the plant responses). Hence, changing the payload simply results in the FMRLC developing, remembering, and applying different responses depending on the type of the payload variation that occurred. Essentially, the FMRLC uses data from the closed-loop system that is generated during on-line operation of the robot to specially tailor the manner in which it designs/tunes the fuzzy controller. This enables it to achieve better performance than the direct fuzzy controller in Section III where no on-line information is used.

V. Conclusions

We have explained how we can use intuition and experience from previous experiments to manually construct a fuzzy controller for a two-link manipulator. Moreover, we have shown how to develop and implement a FMRLC that can: i) automatically synthesize a fuzzy controller to achieve comparable performance to that obtained with a manually constructed fuzzy controller, and ii) automatically tune a fuzzy controller so that it can maintain high performance operation even when there are variations in the payload.

A comprehensive study for this problem, which is beyond the scope of this paper, would attempt to compare conventional adaptive techniques, as well as other nonconventional techniques (such as those outlined in the introduction). We note, however, that several studies using controller autotuning and conventional identification for flexible-link robot systems (e.g., [10], [14]–[17]) have met with varying degrees of success. While the FMRLC approach presented here shows considerable promise as compared to these approaches, several issues remain for further study including: i) a detailed comparative analysis between conventional and fuzzy control approaches for the two-link flexible robot, ii) a mathematical analysis of the fuzzy controller and FMRLC to prove that the system possesses certain stability and convergence properties, iii) a careful theoretical and experimental investigation into persistency of excitation issues and how they can influence the performance of the FMRLC, iv) an investigation into alternative choices for the fuzzy controller used to initialize the FMRLC, v) an investigation into the possibility of using reference models for the acceleration signals, and vi) investigations where gravity effects (motion in the vertical plane) come into play, in which case steady-state positioning errors would dictate the need for clever estimation schemes to account for effects of gravity loading (i.e., "drooping" effects) on the links.

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful comments.

References

[1] J. R. Layne and K. M. Passino, "Fuzzy model reference learning control," in *Proc. 1st IEEE Conf. Contr. Applicat.*, Dayton, OH, Sep. 1992, pp. 686–691.

[2] ——, "Fuzzy learning control for cargo ship steering," *IEEE Contr. Syst.*, vol. 13, pp. 23–24, Dec. 1993.

[3] J. R. Layne, K. M. Passino, and S. Yurkovich, "Fuzzy learning control for anti-skid braking systems," *IEEE Trans. Contr. Syst. Technol.*, vol. 1, no. 2, pp. 122–129, June 1993.

[4] W. A. Kwong, K. M. Passino, and S. Yurkovich, "Fuzzy learning control for aircraft control law reconfiguration," in *Proc. IEEE Int. Symp. on Intell. Contr.*, Columbus, OH, Aug. 1994, pp. 333–338.

[5] J. R. Layne, "Fuzzy model reference learning control," Master's thesis, Dept. Elec. Eng., The Ohio State Univ., Mar. 1992.

[6] S. Yurkovich, "Flexibility effects on performance and control," in *Robot Control*, M. W. Spong, F. L. Lewis, and C. T. Abdallah, Eds. New York: IEEE Press, chap. 8, 1992.

[7] "Motion and vibration control using command shaping methods," in *Am. Contr. Conf.*, San Francisco, CA, June 2–4 1993, pp. 2690–2720.

[8] K. L. Hillsley and S. Yurkovich, "Vibration control of a two link flexible robot arm," *Dynamics and Control*, vol. 3, no. 3, pp. 261–280, July 1993.

[9] D. P. Magee and W. J. Book, "Eliminating multiple modes of vibration in a flexible manipulator," in *IEEE Int. Conf. Robot. Automat.*, Atlanta, GA, May 2–6, 1993, vol. 2, pp. 474–479.

[10] A. Tzes and S. Yurkovich, "An adaptive input shaping control scheme for vibration suppression in slewing flexible structures," *IEEE Trans. Contr. Syst. Tech.*, vol. 1, no. 2, June 1993.

[11] S. Panzieri and G. Ulivi, "Design and implementation of a state observer for a flexible robot," in *IEEE Int. Conf. Robot. Automat.*, Atlanta, GA, May 2–6 1993, vol. 3, pp. 204–209.

[12] L. J. Alder and S. M. Rock, "Adaptive control of a flexible-link robotic manipulator with unknown payload dynamics," in *Am. Contr. Conf.*, San Francisco, CA, June 2–4 1993, pp. 2088–2092.

[13] E. García-Benitez, J. Watkins, and S. Yurkovich, "Nonlinear control with acceleration feedback for a two-link flexible robot," *IFAC Contr. Eng. Practice*, vol. 1, no. 6, pp. 987–989, 1993.

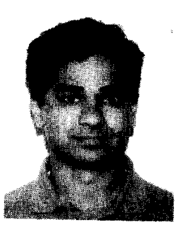
[14] A. P. Tzes and S. Yurkovich, "Application and comparison of on-line identification methods for flexible manipulator control," *Int. J. Robot. Res.*, vol. 10, no. 5, pp. 515–527, Oct. 1991.

[15] S. Yurkovich, F. E. Pacheco, and A. P. Tzes, "On-line frequency domain information for control of a flexible-link robot with varying payload," *IEEE Trans. Automat. Contr.*, vol. 34, pp. 1300–1304, Dec. 1989.

[16] S. Yurkovich and F. E. Pacheco, "On controller tuning for a flexible-link manipulator with varying payload," *J. Robot. Syst.*, vol. 6, no. 3, pp. 233–254, June 1989.

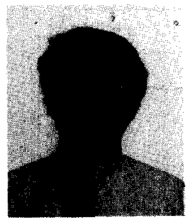
[17] S. Yurkovich, A. P. Tzes, and K. L. Hillsley, "Identification and control for a manipulator with two flexible links," in *IEEE Conf. Dec. Contr.*, Honolulu, HI, vol. 6, pp. 1995–2000, Dec. 1990.

[18] E. Kubica and D. Wang, "A fuzzy control strategy for a flexible link robot," in *IEEE Int. Conf. Robot. Automat.,* Atlanta, GA, May 2–6, 1993, vol. 2, pp. 236–241.
[19] E. Garcia-Benitez, S. Yurkovich, and K. Passino, "A fuzzy supervisor for flexible manipulator control," in *Proc., 1991 IEEE Int. Symp. Intell. Contr.,* Arlington, VA, Aug. 1991, pp. 37–42.
[20] ——, "Rule-based supervisory control of a two-link flexible manipulator," *J. Intell. Robot. Syst.,* vol. 7, pp. 195–213, 1993.
[21] V. G. Moudgal, K. M. Passino, and S. Yurkovich, "Expert supervisory control for a two-link flexible robot," in *IEEE Int. Conf. on Robot. Automat.,* San Diego, CA, May 8–13, 1994, pp. 3296–3301.
[22] ——, "Rule-based control for a flexible-link robot," *IEEE Trans. Contr. Syst. Tech.,* vol. 2, no. 4, pp. 392–405, Dec. 1994.
[23] V. G. Moudgal, W. A. Kwong, K. M. Passino, and S. Yurkovich, "Fuzzy learning control for a flexible-link robot," in *Proc. Am. Contr. Conf.,* Baltimore, MD, June, 1994, pp. 563–567.
[24] T. Procyk and E. Mamdani, "A linguistic self-organizing process controller," *Automatica,* vol. 15, no. 1, pp. 15–30, 1979.
[25] K. J. Åström and B. Wittenmark, *Adaptive Control.* Reading, MA: Addison-Wesley, 1989.
[26] E. Scharf and N. Mandic, "The application of a fuzzy controller to the control of a multi-degree-of-freedom robot arm," in *Ind. Applicat. Fuzzy Contr.,* M. Sugeno, Ed. Amsterdam, The Netherlands: North-Holland, 1985.
[27] R. Tanscheit and E. Scharf, "Experiments with the use of a rule-based self-organizing controller for robotics applications," *Fuzzy Sets Syst.,* vol. 26, pp. 195–214, 1988.
[28] S. Shao, "Fuzzy self-organizing controller and its application for dynamic processes," *Fuzzy Sets Syst.,* vol. 26, pp. 151–164, 1988.
[29] S. Isaka, A. Sebald, A. Karimi, N. Smith, and M. Quinn, "On the design and performance evaluation of adaptive fuzzy controllers," in *Proc., 1988 IEEE Conf. Dec. Contr.,* Austin, TX, Dec. 1988, pp. 1068–1069.
[30] S. Daley and K. F. Gill, "Comparison of a fuzzy logic controller with a P+D control law," *J. Dynamical Syst., Measurement, Contr.,* vol. 111, pp. 128–137, June 1989.
[31] ——, "Altitude control of a spacecraft using an extended self-organizing fuzzy logic controller," *Proc. I. Mech. E.,* vol. 201, no. 2, pp. 97–106, 1987.
[32] ——, "A design study of a self-organizing fuzzy logic controller," *Proc. I. Mech. E.,* vol. 200, pp. 59–69, 1986.
[33] G. Bartolini, G. Casalino, F. Davoli, M. Mastretta, R. Minciardi, and E. Morten, "Development of performance adaptive fuzzy controllers with applications to continuous casting plants," *Ind. Applicat. Fuzzy Contr.,* pp. 73–86, 1985.
[34] H. Takahashi, "Automatic speed control device using self-tuning fuzzy logic," in *1988 IEEE Workshop on Automotive Applicat. Electron.,* Dearborn, MI, Oct. 1988, pp. 65–71.
[35] E. Czogała and W. Pedrycz, "On identification in fuzzy systems and its applications in control problems," *Fuzzy Sets Syst.,* vol. 6, pp. 73–83, 1981.
[36] F. Van Der Rhee, H. Van Nauta Lemke, and J. Dijkman, "Knowledge based fuzzy control of systems," *IEEE Trans. Automat. Contr.,* vol. 35, no. 2, pp. 148–155, Feb. 1990.
[37] P. Graham and R. Newell, "Fuzzy adaptive control of a first-order process," *Fuzzy Sets Syst.,* vol. 31, pp. 47–65, 1989.
[38] ——, "Fuzzy identification and control of a liquid level rig," *Fuzzy Sets Syst.,* vol. 8, pp. 255–273, 1988.
[39] V. G. Moudgal, "Direct, supervisory, and adaptive fuzzy control of a two-link flexible manipulator," Master's thesis, Dept. Elec. Eng., The Ohio State Univ., Sep. 1993.
[40] C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller-Part I," *IEEE Trans. Syst., Man., Cybern.,* no. 2, vol. 20, pp. 404–418, Mar./Apr., 1990.



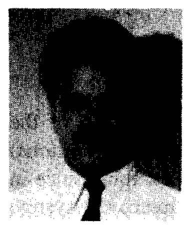
Vivek G. Moudgal received the M.S. degree in electrical engineering from The Ohio State University in 1993 and the B.E. degree in instrumentation engineering from the College of Engineering, Poona, India in 1990.

He is presently employed by dSPACE, Inc., Southfield, MI as an Application Engineer. He also worked as a Project Engineer at Alfa Laval, Ltd. His current research interests are in designing, setting up, and commissioning control systems in the food and chemical industry as well as DSP's for real time simulation and control applications.



Waihon Andrew Kwong received the M.S. and B.S. degrees in electrical engineering from The Ohio State University in 1994 and 1992, respectively.

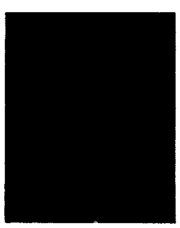
He has worked as a Project Engineer at Sun's Engineering, Inc. He is presently employed by Epsilon Lambda Electronics Corp. Chicago, IL as a Design Engineer. His current research interests are designing quality control systems in the paper and pulp industry, and using DSP's for a radar detection and identification system for automotive applications.



Kevin M. Passino (S'79–M'89) received the M.S. and Ph.D. degrees in electrical engineering from the University of Notre Dame in 1989 and the B.S.E.E. degree from Tri-State University in 1983.

He has worked in the Control Systems Group at Magnavox Electronic Systems Co., Ft. Wayne, IN, on research in missile control, and at McDonnell Aircraft Co., St. Louis, MO, on research in flight control. He spent a year at Notre Dame as a Visiting Assistant Professor and is currently an Assistant Professor in the Department of Electrical Engineering at The Ohio State University. His current research interests include intelligent and autonomous control techniques, nonlinear analysis of intelligent control systems, failure detection and identification systems, and scheduling and stability analysis of flexible manufacturing systems.

Dr. Passino is an Associate Editor for the IEEE Transactions on Automatic Control; served as the Guest Editor for the 1993 IEEE Control Systems Magazine Special Issue on Intelligent Control; is currently a Guest Editor for a special track of papers on Intelligent Control for *IEEE Expert Magazine*; and is on the Editorial Board of the *International Journal for Engineering Applications of Artificial Intelligence*. He is the Publicity Co-Chair for the *IEEE Conference on Decision and Control* in Japan in 1996; was a Program Chairman for the *8th IEEE International Symposium on Intelligent Control*, 1993; and he is serving as the Finance Chair for the *Ninth IEEE International Symposium on Intelligent Control.* He is co-editor (with P. J. Antsaklis) of the book "*An Introduction to Intelligent and Autonomous Control*," (Kluwer Academic Press, 1993). He is a member of the IEEE Control Systems Society Board of Governors. He also serves as the IEEE Control Systems Society Liaison to the IEEE Press and has served in the past as the Chairman of Student Activities for the IEEE Control Systems Society.



Stephen Yurkovich (S'83–M'84–SM'93) received the B.S. degree in engineering science from Rockhurst College in Kansas City, MO, in 1978, and the M.S. and Ph.D. degrees in electrical engineering from the University of Notre Dame in 1981 and 1984, respectively.

He held teaching and postdoctoral research positions at Notre Dame in 1984, and a Visiting Associate Professor position in 1992. In 1984 he moved to the Department of Electrical Engineering at The Ohio State University, where he is currently Associate Professor. His current research, published in more than 100 technical journal articles and conference papers, includes work in the areas of system identification and parameter set estimation for control, and fuzzy logic for control, in application areas including flexible mechanical structures, intelligent vehicle highway systems, and automotive systems. He teaches a variety of undergraduate and graduate level courses in control theory, and has authored the text *Control Systems Laboratory*.

Dr. Yurkovich is General Chair for the 1996 Conference on Control Applications, and Program Chair for the 1997 American Control Conference. He has held numerous positions within the IEEE Control Systems Society: he is past chairman of the Standing Committees on Student Activities and on Publications, is an elected member of the Board of Governors, was an Executive Officer in 1993, and has been elected as Vice-President for Publications for a term beginning Jan. 1, 1995. He is currently Editor-in-Chief of IEEE Control Systems.